UNIVERSIDAD POLITÉCNICA DE VALENCIA
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

TESIS DOCTORAL

Extracción y contextualización de reglas comprensibles a partir de modelos de "caja negra"

Ricardo Blanco Vega

Memoria para optar al grado de Doctor en Informática
bajo la dirección de

José Hernández Orallo
María José Ramírez Quintana

Valencia, España, julio de 2006

A toda mi familia por su apoyo y comprensión,
especialmente a mi ma´linda por su amor infinito

Resumen

En las últimas décadas, han aparecido muchos métodos de aprendizaje automático que permiten obtener modelos altamente precisos, como, por ejemplo, las redes neuronales, las máquinas de vectores soporte y la combinación de clasificadores. Sin embargo, estos métodos en general no son idóneos para obtener conocimiento. El problema que presentan es la pobre comprensibilidad del modelo aprendido, es decir, no permiten mostrar estructuras explicativas de cómo se obtienen las predicciones, siendo estas estructuras o patrones entendibles para los humanos. En otras palabras, el modelo subyacente es preciso, pero su representación no es adecuada. Adicionalmente, estos modelos precisos con el paso del tiempo presentan fallos de predicción ante cambios en la evidencia o han de adaptarse a otro contexto. Generalmente, en estos casos, no es conveniente desechar el modelo "antiguo" completamente, por lo que es necesaria su revisión o adaptación.

En esta tesis se abordan estos problemas a través del método mimético, el cual puede verse como una forma general de extraer una explicación (en forma de árboles de decisión o sistemas de reglas) de cualquier modelo existente sin considerar ningún mecanismo interno del mismo (esto es, considerándolo una caja negra). La técnica mimética puede verse como un "reaprendizaje", donde se generan datos, que al etiquetarse por el modelo existente, sirven para aprender un "nuevo" modelo que imita o "mimetiza" el original.

En primer lugar, estudiamos teóricamente el comportamiento del método mimético usando como herramienta las curvas de aprendizaje, analizamos el efecto de usar o no el conjunto de entrenamiento del modelo original, caracterizamos los escenarios en los que la técnica es beneficiosa, atendiendo a la precisión del modelo mimético, etc.. En segundo lugar, realizamos una evaluación experimental de la técnica usando conjuntos de datos del repositorio UCI y considerando dos casos generales de estudio: incluyendo o no el conjunto de datos originales. Dentro de esta evaluación experimental, estudiamos algunos de los factores que influyen en el método, como por ejemplo, el nivel de poda, el nivel de confianza, el tamaño de los datos inventados, etc., así como su influencia combinada. El objetivo del estudio de estos factores ha sido encontrar un ajuste de los

mismos para obtener un compromiso entre comprensibilidad y precisión, que ha permitido establecer "procedimientos" para aplicar el método mimético de una manera óptima.

En tercer lugar, y como aportación más novedosa de la tesis, ampliamos las fronteras de aplicación del método mimético a la contextualización de modelos. La contextualización o adaptación es necesaria en la vida de todo modelo y pretende alargar la vida útil del mismo. Los cambios de contexto se pueden presentar de diferentes maneras, como por ejemplo, la aparición de nuevos datos que anteriormente no existían o no eran importantes, el cambio de formato de los datos cuando aparecen o desaparecen atributos o valores de clase, la modificación de los costes de los errores, etc.. Si tomamos como base que la técnica mimética es un proceso de aprendizaje, en esta tesis mostramos que cualquier técnica de aprendizaje computacional se puede utilizar como método general de revisión y de adaptación de modelos, independientemente de cómo el modelo fue generado y sin usar los datos originales o la distribución de los mismos.

De manera transversal y con la vista en las aplicaciones, se ha extendido, respecto a trabajos anteriores, el uso del método mimético a diferentes tipos de modelos de "caja negra": multiclasificadores, redes neuronales, naive bayes, expertos humanos, etc.. Hemos perseguido mejorar la técnica para aplicaciones en áreas donde es necesaria al mismo tiempo la comprensibilidad y una buena precisión.

Los diferentes resultados y contribuciones de la tesis pueden ser aplicados a diferentes áreas: minería de datos, ingeniería inversa (por ej. dado un modelo existente obtener un modelo que imite al antiguo, el cual tenga las características de nuestro interés: mayor precisión, mejor comprensibilidad o un equilibrio entre ambas), revisión y actualización de modelos, adquisición y gestión de conocimiento, ayudando a la obtención de conocimiento para sistemas expertos.

Palabras claves:
Aprendizaje automático, minería de datos, extracción de reglas, comprensibilidad, revisión de modelos, adaptación de modelos, revisión de teorías, ingeniería de conocimiento, árboles de decisión.

Resum

En les últimes dècades, han aparegut molts mètodes d'aprenentatge automàtic que permeten obtenir models altament precisos, com ara les xarxes neuronals, les màquines de vectors suport i la combinació de classificadors. No obstant això, aquests mètodes en general no són idonis per a obtenir coneixement. El problema que presenten és la pobra comprensibilitat del model après, és a dir, no permeten mostrar estructures explicatives de com s'obtenen les prediccions, sent aquestes estructures o patrons intel•ligibles per als humans. En altres paraules, el model subjacent és precís, però la seua representació no és adequada. Addicionalment, aquests models precisos amb el pas del temps presenten fallades de predicció davant canvis en l'evidència o han d'adaptar-se a un altre context. Generalment, en aquests casos, no és convenient rebutjar el model "antic" completament, pel que és necessària la seua revisió o adaptació.

En aquesta tesi s'aborden aquests problemes a través del mètode mimètic, el qual pot veure's com una forma general d'extraure una explicació (en forma d'arbres de decisió o sistemes de regles) de qualsevol model existent sense considerar cap mecanisme intern del mateix (açò és, considerant-lo una caixa negra). La tècnica mimètica pot veure's com "un reaprenentatge", on es generen dades, que en etiquetar-se pel model existent, serveixen per a aprendre un "nou" model que imita o "mimetitza" l'original.

En primer lloc, estudiem teòricament el comportament del mètode mimètic usant com a eina les corbes d'aprenentatge, analitzem l'efecte d'usar o no el conjunt d'entrenament del model original, caracteritzem els escenaris en els quals la tècnica és beneficiosa, atenent a la precisió del model mimètic, etc. En segon lloc, realitzem una avaluació experimental de la tècnica usant conjunts de dades del repositori UCI i considerant dos casos generals d'estudi: incloent o no el conjunt de dades originals. Dins d'aquesta avaluació experimental, estudiem alguns dels factors que influeixen en el mètode, com per exemple, el nivell de poda, el nivell de confiança, la grandària de les dades inventades, etc., així com la seua influència combinada. L'ob-

jectiu de l'estudi d'aquests factors ha estat trobar un ajustament dels mateixos per a obtenir un compromís entre comprensibilitat i precisió, que ha permès establir “procediments” per a aplicar el mètode mimètic d’una manera òptima.

En tercer lloc, i com aportació més novella de la tesi, ampliem les fronteres d'aplicació del mètode mimètic a la contextualització de models. La contextualització o adaptació és necessària en la vida de tot model i pretén allargar-ne la vida útil. Els canvis de context es poden presentar de diferents maneres, com per exemple, l'aparició de noves dades que anteriorment no existien o no eren importants, el canvi de format de les dades quan apareixen o desapareixen atributs o valors de classe, la modificació dels costos dels errors, etc.. Si prenem com a base que la tècnica mimètica és un procés d'aprenentatge, en aquesta tesi mostrem que qualsevol tècnica d'aprenentatge computacional es pot utilitzar com a mètode general de revisió i d'adaptació de models, independentment de com el model va ser generat i sense usar les dades originals o la seua distribució.

De manera transversal i amb la vista en les aplicacions, s'ha estès, respecte a treballs anteriors, l'ús del mètode mimètic a diferents tipus de models de “caixa negra”: multiclassificadors, xarxes neuronals, Naïve Bayes, experts humans, etc. Hem perseguit millorar la tècnica per a aplicacions en àrees on és necessària al mateix temps la comprensibilitat i una bona precisió.

Els diferents resultats i contribucions de la tesi poden ser aplicats a diferents àrees: mineria de dades, enginyeria inversa (per ex. donat un model existent obtenir un model que imite a l'antic, el qual tinga les característiques del nostre interès: major precisió, millor comprensibilitat o un equilibri entre ambdues), revisió i actualització de models, adquisició i gestió de coneixement, ajudant a l'obtenció de coneixement per a sistemes experts.

Abstract

In recent years, many machine learning methods have been introduced which are able to generate highly accurate models, such as artificial neural networks, support vector machines or ensemble methods. However, these methods are not especially suitable to extract knowledge from data. The problem is the poor comprehensibility of the models which are generated, i.e., these techniques do not easily show any explanatory structure about how the model works, in the way these structures can be understood by humans. In other words, the underlying model is accurate, but its representation is not adequate. Additionally, the performance of these accurate models can diminish with time, since the evidence might change or the model has to be adapted to new contexts. Generally, in these cases, it is not convenient to reject the "old" model completely; a revision or adaptation of the model can be more sensible.

In this thesis we tackle these problems through the mimetic method, a method which can be seen as a general way of extracting an explanation (in the form of decision trees or rule systems) from any existing model, without considering any inner mechanism inside the model (i.e., seeing it as a black box). The mimetic technique can be seen as a "re-learning", where some new data is generated, this data is labelled with the existing model, and the labelled data is used to learn a "new" model which mimics the original one.

First, we study the mimetic method theoretical by using learning curves as a tool, we analyse the effect of using (or not) the training set which led to the original model, we characterise the scenarios where the technique is beneficial, according to the accuracy of the mimetic model, etc. Secondly, we perform an experimental evaluation by using several datasets from the UCI repository and considering two general cases of study: including or not the original dataset. Within this experimental evaluation, we study some of the factors which have influence in the method, such as, the pruning level, the confidence level, the size of the invented data, etc., as well as their combined influence. The goal of this study has been to find an appropriate tuning to find a trade-off between comprehensibility and accuracy, which has allowed us to establish several "procedures" to apply the mimetic method in an optimal way.

Thirdly, and as the most novel contribution of this work, we extend the frontiers of the application of the mimetic method to model contextualisation. Model contextualisation or adaptation is necessary in the lifecycle of every model and pursues the extension of its lifespan. Context changes can appear in many different ways, as for instance, the appearance of new data which didn't exist previously or wasn't important, format changes where new attributes or class values appear, changes in the cost matrix, etc. If we reconsider that the mimetic techniques is a learning process, in this thesis we show that any machine learning technique can be used as a general method to revise and adapt existing models, independently from how the model was generated and without the original or knowledge about their distribution.

In a transversal way and looking over to applications, we have extended the use of the mimetic method to different kinds of black-box models: ensemble methods, neural networks, Naïve Bayes, human experts, etc. We have striven to improve the technique for applications in areas where we need, at the same time, comprehensibility and a good accuracy.

The several results and contributions of this work can be applied to many areas: data mining, inverse engineering (e.g. given an existing model, we can obtain a model which mimics the original, but with some features of interest: more accuracy, more comprehensibility or a trade-off in between), model revision and update, knowledge acquisition and management to assist knowledge extraction for expert systems, ...

Agradecimientos

Esta tesis es el resultado del trabajo realizado en el grupo *Multi-paradigm Inductive Programming* (*MIP*) del Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia. Culminarla no sería posible sin el apoyo, la colaboración de equipo y el cariño de muchas personas.

Antes que nada agradezco a Dios por hacer posible esta ilusión, que sin sus múltiples bendiciones no hubiera sido posible llevar a buen fin esta fabulosa empresa.

A Raquel mi amantísima esposa que con su apoyo y amor me da fuerzas para realizar cualquier cosa.

A Raquel Aracely, Iván Ricardo y Edgar Andrés por haber tenido el coraje y la valentía de aceptar la desarticulación de su vida académica a cambio de la de su padre.

A mis directores de tesis Mª José Rámirez Quintana y José Hernández Orallo que sin su colaboración de equipo e ideas no sería posible la realización de esta tesis.

El desarrollo de esta tesis es debido al apoyo económico de todas las siguientes instituciones y personas: ANUIES (SUPERA becario no. 5177), licencia beca comisión de la Dirección General de Institutos Tecnológicos de México, beca del convenio SEIT-ANUIES, especialmente el Instituto Tecnológico de Chihuahua II, a mi alma mater de licenciatura la Universidad Autónoma de Chihuahua (UACH), al Sindicato del Personal de la UACH, a mi alma mater del doctorado la Universidad Politécnica de Valencia y a mis amados familiares.

Tabla de Contenido

1. Introducción 1

1.1. Motivación 2

1.2. Hipótesis de partida: El método mimético 6

1.3. Objetivos 7

1.4. Organización de la tesis 9

2. Preliminares 11

2.1. Árboles de decisión 14

2.2. Técnicas de evaluación de hipótesis 20

2.3. Otro criterio de evaluación: la comprensibilidad 29

2.4. Contextualización 31

3. Extracción de modelos comprensibles 41

3.1. Extracción de reglas a partir de modelos de caja negra . 44

3.2. El método mimético 45

3.3. Generación de los datos inventados 48

3.4. Ventajas y desventajas del método mimético 51

4. Estudio teórico 53

4.1. Notación y configuración 54

4.2. Resultados generales 57

4.3. Resultados para árboles de decisión 63

4.4. Curvas de aprendizaje e influencia del conjunto de datos de entrenamiento 65

4.5. Discusión y conclusiones 93

5. Análisis con datos de entrenamiento 97

5.1. Configuración experimental 101

5.2. Análisis de la poda 102

5.3. Análisis del tamaño de los datos inventados 105

5.4. Uso de la confianza en el método mimético 111
5.5. Búsqueda del caso óptimo por combinación de factores 115
5.6. Conclusiones 117

6. Análisis sin datos de entrenamiento 119

6.1. Análisis del modelo mimético desde la perspectiva del principio MML 122
6.2. Optimización del tamaño de R usando el principio MML modificado 124
6.3. Evaluación experimental 128
6.4. Procedimiento de aplicación 133
6.5. Conclusiones 134

7. Contextualización con revisión 137

7.1. Revisión de modelos a través de aprendizaje 139
7.2. Escenario con novedad 141
7.3. Evaluación experimental 146
7.4. Discusión 155
7.5. Conclusiones 160

8. Contextualización por cambios de formato.....161

8.1. Los árboles de decisión y los datos perdidos 162
8.2. Escenario con la aparición de un nuevo atributo 164
8.3. Experimentos cuando aparece un atributo importante 169
8.4. Experimentos cuando aparece una nueva clase 175
8.5. Conclusiones 181

9. Contextualización al coste 183

9.1. Trabajos relacionados 185
9.2. Contextualización mimética 188
9.3. Determinación del umbral 191
9.4. Aprendizaje de los modelos miméticos 209
9.5. Experimentos 214

9.6. Discusión 229
9.7. Conclusiones 232

10. Conclusiones y trabajo futuro 235
10.1. Conclusiones 236
10.2. Contribuciones de la tesis 240
10.3. Trabajo futuro 243

Referencias 245

Lista de Tablas

Tabla 2.1 Funciones de impureza para algunos criterios de partición. 19
Tabla 2.2 Las métricas basadas en la matriz de confusión para problemas binarios. 23
Tabla 5.1. Información de los conjuntos de datos empleados en los experimentos..... 102
Tabla 5.2. Información sobre los conjuntos de datos de validación. 109
Tabla 5.3. Tamaños y calidades estimadas y máximas para los conjuntos de datos de validación. 111
Tabla 5.4. Resultados experimentales obtenidos por la combinación de varios factores: poda, tamaño del conjunto de datos inventados y confianza. Prop. Significa proporción de reglas y Prec es la precisión. 116
Tabla 6.1. Conjuntos de datos usados en los experimentos .. 129
Tabla 6.2. Parámetros y coeficientes de determinación para las curvas de aprendizaje miméticas creada con cien puntos (n=100). 130
Tabla 6.3 Parámetros y coeficientes de determinación para las curvas de aprendizaje miméticas creadas con tres puntos (n = 3) 131
Tabla 6.4 Costes mínimos estimados para 3 modelos y el observado de entre 100 modelos. 132
Tabla 7.1. Información de los conjuntos de datos utilizados en los experimentos..... 146
Tabla 7.2. Precisión del modelo antiguo empleando los datos originales y nuevos, así como el modelo nuevo con los datos nuevos. 148
Tabla 7.3. Configuración del entrenamiento y validación de los modelos miméticos. 149
Tabla 7.4. Factores α reales y estimados para los modelos miméticos. 152
Tabla 7.5. Precisión promedio de los modelos miméticos. Los dos mejores resultados en negrita. 152
Tabla 7.6. Precisión promedio de los diferentes modelos. 154
Tabla 7.7. Resultados de las pruebas t de *Student* entre M_{Mim_α} y los modelos combinados (M_{Comb} y M_{Comb_α}). 'V' denota que M_{Mim_α} gana, 'X' denota que M_{Mim_α} pierde y '–' denota un empate. El resumen se interpreta como (V/–/X). 154

Tabla 7.8. Resultados de las pruebas t de *Student* entre M_{Mim_α} vs. M_{Old} y M_{New}. 'V' denota que M_{Mim_α} gana, 'X' denota que M_{Mim_α} pierde y '–' denota un empate. El resumen se interpreta como (V/–/X). 155
Tabla 7.9. Precisiones promedio obtenidas con los experimentos con diferentes patrones de la situación original y final. 159
Tabla 8.1. Conjunto de datos *weather*. 162
Tabla 8.2. Conjunto de datos *weather*, con un nulo en el atributo *outlook* del primer ejemplar. 163
Tabla 8.3. Lista de acciones estudiadas sobre un árbol de decisión creado a partir del conjunto de datos *weather*. 165
Tabla 8.4. Resultados al realizar acciones sobre los atributos de un árbol de decisión creado con el conjunto de datos *weather*. 168
Tabla 8.5. Información de los conjuntos de datos utilizados en los experimentos cuando aparece un nuevo atributo importante. 169
Tabla 8.6. Resultados promedio usando un 10% de los datos para T_{New}. 171
Tabla 8.7. Resultados promedio usando un 15% de los datos para T_{New}. 172
Tabla 8.8. Resultados promedio usando un 20% de los datos para T_{New}. 172
Tabla 8.9. Resultados promedio usando un 25% de los datos para T_{New}. 173
Tabla 8.10 Pruebas t para modelos obtenidos usando un 25% de los datos para T_{New}. 175
Tabla 8.11. Información de los conjuntos de datos empleados en los experimentos cuando aparece una nueva clase. 176
Tabla 8.12. Resultados promedio usando un 10% de los datos para T_{New}. 177
Tabla 8.13. Resultados promedio usando un 15% de los datos para T_{New}. 178
Tabla 8.14. Resultados promedio usando un 20% de los datos para T_{New}. 179
Tabla 8.15 Pruebas t para modelos obtenidos usando un 25% de los datos para T_{New}. 181
Tabla 9.1 Enumeración de todos los posibles modelos miméticos estudiados. 213
Tabla 9.2. Información de los conjuntos de datos utilizados en los experimentos. Los conjuntos de datos con la observación de balanceados fueron balanceados

aleatoriamente. Los conjuntos de datos con observación casi balanceado y desbalanceado son copia fiel del repositorio de la UCI. 215
Tabla 9.3 Costes promedio de todos los modelos miméticos agrupados por *skew*. En negrita los de menor coste global..... 217
Tabla 9.4 Precisiones y costes promedio de todos los modelos miméticos de acuerdo al tipo de batería de experimento usando Naive Bayes como oráculo. ... 219
Tabla 9.5 Precisiones y costes promedio de todos los modelos miméticos de acuerdo al tipo de batería de experimento usando la red neuronal como oráculo. 220
Tabla 9.6. Ranking de los modelo en base al coste..... 221
Tabla 9.7. Modelos seleccionados para realizar las pruebas t de *Student*. Bala se refiere a los conjuntos de datos balanceados, Mayo significa que son los conjuntos de datos donde la clase 1 es la mayoritaria y Mino representa los conjuntos de datos donde la clase 1 es la minoritaria..... 221
Tabla 9.8 Pruebas t para modelos obtenidos con conjuntos de datos balanceados y NB como oráculo. 222
Tabla 9.9 Pruebas t para modelos obtenidos con conjuntos de datos balanceados y como oráculo RN. 222
Tabla 9.10 Pruebas t para modelos obtenidos con conjuntos de datos no balanceados y clase mayoritaria usando el oráculo NB..... 223
Tabla 9.11 Pruebas t para modelos obtenidos con conjuntos de datos no balanceados y clase mayoritaria usando el oráculo RN..... 223
Tabla 9.12 Pruebas t para modelos obtenidos con conjuntos de datos no balanceados y clase minoritaria usando el oráculo NB..... 224
Tabla 9.13 Pruebas t para modelos obtenidos con conjuntos de datos no balanceados y clase minoritaria usando el oráculo RN..... 224
Tabla 9.14 Índice de pérdida en los promedios de precisión y coste para los modelos miméticos. En negrita el menor índice. Los valores negativos representan malos modelos, debido a que no lograron ganarle en coste al oráculo..... 226
Tabla 9.15 Resumen de las reglas de los árboles de decisión de los modelos miméticos Mim2Ordb generados con el conjunto de datos balanceado *vote*. Entre paréntesis se muestran la cantidad de instancias que cubre la

regla y los errores encontrados (total instancias cubiertas/errores)..... 227
Tabla 9.16 Resumen de los parámetros de los árboles de decisión de los modelos miméticos Mim2Ordb generados con el conjunto de datos vote. Donde *R* = número de reglas, *C* = número de condiciones, *Max_R* = máximo número de reglas por clase, *Max_C* = Máximo número de condiciones por clase, *Compl* = complejidad y *Compre* = comprensibilidad. 228
Tabla 9.17 Resumen de las métricas del error cuadrático medio (MSE), Logloss y área debajo de la curva ROC (AUC) usando NB como oráculo. 230
Tabla 9.18 Resumen de las métricas del error cuadrático medio (MSE), Logloss y área debajo de la curva ROC (AUC) usando RN como oráculo. 231

Lista de Figuras

Figura 2.1 Proceso del KDD 11
Figura 2.2. Árbol de decisión para determinar el tipo de plantas iris 15
Figura 2.3 Algoritmo de aprendizaje de árboles de decisión por "Partición" (Divide y Vencerás) 16
Figura 2.4. Partición cuadricular del espacio obtenido por un árbol de decisión para un problema con dos atributos (X e Y) numéricos..... 18
Figura 2.5. Matriz de costes..... 34
Figura 2.6. Clasificador en el espacio ROC. 35
Figura 2.7 Ejemplo de diagrama ROC..... 36
Figura 3.1 El algoritmo de meta-aprendizaje CMM..... 46
Figura 3.2 La Técnica Mimética 47
Figura 4.1 Proceso mimético sin conjunto de datos de entrenamiento en la segunda etapa. 65
Figura 4.2 Una curva de aprendizaje aproximándose a su punto de saturación..... 72
Figura 4.3 Gráfica de respuesta mimética para determinar cuándo el método es útil..... 76
Figura 4.4 Gráfica de respuesta mimética cuándo el método es inútil. 77
Figura 4.5 Gráfica de respuesta mimética generalizada cuando usamos el conjunto de datos de entrenamiento..... 89
Figura 5.1 La Técnica Mimética con datos de entrenamiento. . 97
Figura 5.2. Precisión contra Proporción de Reglas de Mim para diferentes grados de poda 103
Figura 5.3. Métrica de calidad vs proporción de reglas de Mim RN para diferentes grados de poda..... 104
Figura 5.4. Fidelidad contra Proporción de Reglas de Mim para varios grados de poda cuando el oráculo mejora en precisión al J48 105
Figura 5.5. Precisión contra Tamaño del conjunto de datos inventados..... 106
Figura 5.6. Precisión contra Proporción de Reglas de Mim para diferentes tamaños 107
Figura 5.7. Métrica de Calidad contra Proporción de Reglas de Mim para diferentes tamaños 108
Figura 5.8. Métrica de Calidad contra Número de Reglas en los conjuntos de datos de validación 111
Figura 5.9. Trasformación del conjunto de entrenamiento para la técnica mimética..... 112

Figura 5.10. Precisión contra Proporción de Reglas del modelo mimético dependiendo de un umbral de confianza y un factor de repetición. 114
Figura 5.11. Métrica de Calidad contra Factor de Repetición para diferentes umbrales de confianza 115
Figura 6.1. Proceso mimético con un experto como oráculo .. 121
Figura 6.2. Error vs. Tamaño para el conjunto de datos *balance-scale* 125
Figura 6.3. Número de reglas vs. Tamaño para el conjunto de datos *balance-scale* 126
Figura 6.4. Procedimiento para estimar el tamaño óptimo 134
Figura 7.1 Modelos alternativos en un escenario de cambio.. 140
Figura 7.2. Revisión mimética en un escenario accionado por novedad 143
Figura 7.3. Comportamiento dependiendo de $|T_{Inv}|$ para el conjunto de datos *zoo*. 150
Figura 7.4. Modelo de comportamiento dependiendo de $|T_{Inv}|$ para el conjunto de datos *balance-scale*. 150
Figura 7.5. Modelo de comportamiento dependiendo de $|T_{Inv}|$ para el conjunto de datos *diabetes*. 151
Figura 7.6. Cambio del modelo de forma global y profunda... 157
Figura 7.7. Cambio del modelo de forma global y profunda donde la técnica mimética funciona bien. 158
Figura 8.1. Árbol de decisión para el conjunto de datos *weather*. 163
Figura 8.2. Árbol de decisión para el conjunto de datos *weather*, con un nulo en el atributo *outlook* del primer registro 164
Figura 8.3. Árbol de decisión para el conjunto de datos *weather*, con la mitad de nulos en el atributo *outlook* y la adición del un nuevo atributo existe. 166
Figura 8.4. Árbol de decisión para el conjunto de datos *weather* habiendo eliminado el atributo *humidity*. 167
Figura 8.5. Árbol de decisión para el conjunto de datos *weather* habiendo asignado la mitad de valores nulos al atributo *humidity*. 167
Figura 8.6. Árbol de decisión para el conjunto de datos *weather* habiendo asignado la mitad de valores nulos al atributo *humidity* y la adición del atributo existe. 168
Figura 8.7. Esquema de los conjuntos de datos y modelos empleados en los experimentos cuando aparece un nuevo atributo. 170
Figura 8.8. Resultados promedio usando un 10% de los datos para T_{New}. 171

Figura 8.9. Resultados promedio usando un 15% de los datos para T_{New}. 172
Figura 8.10. Resultados promedio usando un 20% de los datos para T_{New}. 173
Figura 8.11. Resultados promedio usando un 25% de los datos para T_{New}. 174
Figura 8.12. Esquema de los conjuntos de datos y modelos empleados en los experimentos cuando aparece una nueva clase..... 176
Figura 8.13. Resultados promedio usando un 10% de los datos para T_{New}. 178
Figura 8.14 Resultados promedio usando un 15% de los datos para T_{New}. 179
Figura 8.15 Resultados promedio usando un 20% de los datos para T_{New}. 180
Figura 9.1. La contextualización mimética..... 189
Figura 9.2 Esquema de un vector de probabilidades ordenado para explicar el cálculo del $Threshold_{Ord}$..... 193
Figura 9.3. Función para predecir la clase de un ejemplo dados el umbral por ordenamiento ($Threshold_{Ord}$), el índice de retención (Ro) y su probabilidad estimada p_i.... 196
Figura 9.4. Ejemplo de curva NROC. 201
Figura 9.5. Normalización del espacio ROC. 201
Figura 9.6. Proceso para la obtención y aplicación del modelo mimético simple (esquema 0). 211
Figura 9.7. Proceso para la obtención y aplicación del modelo mimético a posteriori (esquema 1)..... 212
Figura 9.8. Proceso para obtención y aplicación del modelo mimético a priori (esquema 2). 212
Figura 9.9. Proceso para obtención y aplicación del modelo mimético usando sobremuestreo (esquema 3). 212
Figura 9.10 Resultados promedio de costes agrupados por *skew*. 218

Capítulo 1
Introducción

Uno de los principales objetivos de la extracción de conocimiento a través del aprendizaje automático y otras técnicas inductivas es el obtener modelos que sean precisos y comprensibles a la vez. No obstante, no todas las técnicas desarrolladas cumplen con este doble objetivo. Así, técnicas como las redes neuronales son capaces de generar modelos muy buenos desde el punto de vista de la precisión pero, en principio, incomprensibles para el usuario. Por otra parte, técnicas como el aprendizaje de árboles de decisión, que pueden generar directamente modelos comprensibles expresados como un conjunto de reglas, a menudo no obtienen precisiones altas. El dilema se ha acrecentado recientemente con la aparición de nuevas técnicas que proporcionan muy alta precisión a costa de una muy baja comprensibilidad, como son las máquinas de vectores soporte y los sistemas de combinación de modelos.

En esta tesis nos centramos en estudiar y extender un método que permite obtener modelos en forma de reglas a partir de cualquier otro modelo existente, preciso pero incomprensible, que denominamos modelo de "caja negra". Estudiaremos cuáles son los factores de que depende la precisión y la comprensibilidad del modelo resultante. Asímismo, abordaremos dos interesantes extensiones: su aplicación a la adquisición de conocimiento en sistemas expertos y a la contextualización de modelos, entendiendo como tal la revisión de modelos, la adaptación ante cambios de formato y la clasificación sensible al coste.

En este capítulo se plantean las motivaciones que han dado lugar a la elaboración de la tesis. A la vista de las necesidades e inconvenientes actuales de la adquisición y adaptación automática de conocimiento, nuestra propuesta, que desarrollamos en esta tesis, se basa en un método ya existente, el cual nos sirve como hipótesis de partida. A continuación se presentan los objetivos generales perseguidos con este trabajo. Finalmente, se describe la organización del resto de los capítulos de este documento.

1.1. Motivación

Hoy en día, el número y la diversidad de técnicas de modelado usadas en el descubrimiento del conocimiento es ciertamente asombroso. Los programas comerciales de minería de datos, las aplicaciones computacionales estadísticas y otros paquetes de técnicas de aprendizaje automático incorporan generalmente una muestra representativa de estos métodos, incluyendo, entre otros, los árboles de decisión y otros métodos de aprendizaje de reglas, las redes neuronales artificiales, las máquinas de vectores soporte, las técnicas estadísticas de modelación, los métodos basados en distancias, las técnicas borrosas y evolutivas, etc. Se ha demostrado (teórica y prácticamente) que ninguna técnica puede ser óptima en cualquier situación. Esta variedad es entonces necesaria para abordar la diversidad de los problemas encontrados por los expertos en el descubrimiento del conocimiento. A pesar de las muchas ventajas que se derivan del hecho de disponer de tal variedad de métodos, lo que posibilita, por ejemplo, poder seleccionar el método con el que se obtienen mejores prestaciones de acuerdo a los objetivos perseguidos (mejor precisión, menor coste, etc.), esta diversidad también tiene ciertos inconvenientes. Así, los usuarios en el descubrimiento del conocimiento tienen que trabajar con muchos tipos de paradigmas, lo que conlleva una complejidad en el uso de herramientas de minería de datos, pudiendo al mismo tiempo requerir una mayor formación por parte de los usuarios. Finalmente, no todas las técnicas permiten extraer modelos inteligibles, ni siquiera parcialmente o con posprocesamientos específicos.

Otra cuestión importante relacionada en este caso no con la generación de los modelos sino con el uso de los mismos, es la necesidad de alargar la vida útil de los modelos obtenidos, es-

pecialmente aquéllos que se han validado adecuadamente. Para esto es necesario el mantenimiento de los mismos, es decir, contextualizar y transformar su conocimiento para que se adapte a los diferentes cambios que ocurran, y así poder seguir empleándolos. Además, es importante que la adaptación a cambios en el fondo del problema vengan acompañados de adaptaciones de los modelos existentes también en forma de reglas, para poder entender cómo se realiza la adaptación.

1.1.1. Adquisición del conocimiento

Como hemos dicho, dentro del aprendizaje automático existen diversas técnicas, como las redes neuronales, los métodos de combinación, las máquinas de vectores soporte, etc. que generan modelos que son considerados como cajas negras, dado que transforman las entradas en las salidas sin que sea posible expresar de una manera sencilla para el usuario cómo esta transformación se lleva a cabo. Desde el punto de vista de la extracción de conocimiento, aunque estos modelos tengan buenas prestaciones tienen el inconveniente que se puede extraer de ellos poco conocimiento útil para poder ser usado en la toma de decisiones, ya que no son comprensibles. Sin embargo, para muchas aplicaciones la comprensibilidad del modelo es crucial. La comprensibilidad de los modelos no sólo es una propiedad deseable, pues permite al usuario conocer cuáles son las razones por las que el modelo presenta cierto comportamiento, sino que para algunos ámbitos de aplicación, como la bioinformática, el diagnóstico médico, la banca, etc., es de crucial importancia.

Es por ello necesario que se busquen y estudien métodos que logren la comprensibilidad de los modelos de caja negra. En base a esta necesidad se han propuesto diversos métodos para convertir un modelo incomprensible en una representación simple y comprensible: un conjunto de reglas. La extracción de reglas de la forma si-entonces es usualmente aceptada como la mejor forma de expresar el conocimiento representado en una caja negra. No porque sea un trabajo fácil, sino porque al final las reglas creadas son más entendibles para los humanos que otra representación. Las diferentes técnicas de aprendizaje automático y otras técnicas de modelización estadísticas son, por tanto, útiles para la adquisición de conocimiento sólo si son capaces de extraer modelos inteligibles. Sólo así podrán susti-

tuir a los procesos de adquisición de conocimiento que hasta ahora se realizan manualmente.

Los sistemas expertos es una de las áreas donde uno de los problemas principales es el cuello de botella en la adquisición del conocimiento (Greenwell, 1988; Debenham, 1989; Brulé y Blount, 1989; Meyer y Booker, 1991; Feigenbaum, 2003). Cuando ésta se realiza manualmente, muchos expertos no pueden expresar su conocimiento en forma de reglas claras e inequívocas. Los expertos emplean generalmente reglas explícitas, reglas "a ojo" y reglas inconscientes. Incluso en el caso de que el experto pueda escribir todo su conocimiento, esto representa un alto esfuerzo, puede ser un proceso donde se desperdicia mucho tiempo, es difícil de mantener y el resultado es, a veces, un modelo trascrito que no se puede aplicar de una forma completamente automatizada puesto que todavía hay una cierta ambigüedad.

También existen diferentes métodos de adquisición de conocimiento basados en técnicas de aprendizaje automático (por ejemplo: Grover, 1983; BGM de García Martínez, 1994; IDEAL de Pazos, 1996). En general, podemos resumir los pasos que siguen estos métodos en: (1) la definición del dominio del problema, en donde se recopilan casos de uso y luego uno o varios expertos "responden" a ejemplos, actuando como si fueran un "oráculo"; (2) la formulación del conocimiento, aquí es donde se aplican las técnicas de aprendizaje automático para obtener los prototipos; (3) la verificación del conocimiento obtenido para conseguir el mantenimiento perfectivo.

Por tanto, existen varias vías para la adaptación de conocimiento: manualmente, con el consiguiente coste del experto y los problemas mencionados de ambigüedad y la semi-automática, con problemas de precisión o de comprensibilidad (o de ambas).

Independientemente de cómo se ha adquirido el conocimiento (bien manualmente o bien (semi-)automáticamente), y tal y como ya hemos mencionado anteriormente, para alargar su vida útil es necesario mantenerlo a través de su contextualización y transformación.

1.1.2. Contextualización y transformación del conocimiento

En general, cualquier sistema (teoría o modelo) tiene que ser eventualmente revisado o extendido para cubrir nuevas situaciones. En muchos casos, esto se debe a una variación en el problema o en la forma en que éste se presenta, por lo que los cambios pueden obedecer a diferentes causas. Una de estas causas es el cambio de formato del problema, es decir, pueden aparecer o desaparecer variables o dejar de ser importantes algunas de ellas, también pueden aparecer o desaparecer valores de salida, ya sean clases o el rango de una salida numérica. Asímismo, puede suceder que los casos o instancias que trata el modelo cambien sus frecuencias, generando de esta forma cambios en su distribución. Otras veces, podemos tener un modelo desarrollado de una manera general a partir de distintas situaciones, pero que puede aplicarse en contextos diferentes. Por ejemplo, un modelo general de diagnóstico de una enfermedad que debe adaptarse a distintos países, por tener sus poblaciones características diferentes.

El aprendizaje automático y la minería de datos proporcionan métodos muy poderosos para obtener modelos a partir de datos, pero su uso para "retocar" o adaptar un método existente depende de la técnica que se tenga disponible. Así, esto se puede realizar de forma directa si la técnica es incremental. Sin embargo, muchas técnicas de aprendizaje computacional o de minería de datos no son incrementales. Adicionalmente, en muchas situaciones reales no tenemos los datos originales de entrenamiento usados para construir el modelo inicial. Esto puede deberse a que el modelo fue construido mucho tiempo atrás o porque el modelo fue construido, como ya hemos comentado, por un experto humano sin utilizar ninguna técnica de aprendizaje automático o minería de datos. Consecuentemente, en este caso reentrenar el modelo no es una opción. Esta es una situación típica en muchas áreas como la ingeniería, la diagnosis, la manufactura, la medicina, los negocios, etc.

Otro posible cambio en los modelos es la modificación en los costes de los diferentes errores de clasificación. La clasificación en presencia de costes es importante para aplicaciones prácticas, incluyendo muchas aplicaciones de minería de datos, tales como la administración de la relación comercial con los clientes de una empresa (en inglés *Customer Relationship Management* o sus siglas CRM), la detección de fraudes y de instrusos, el

filtrado de correo *spam* y la diagnosis médica, entre otras. La línea de trabajo sobre este tema es conocida como aprendizaje sensible al coste (*cost-sensitive learning*), en las áreas del aprendizaje automático y de la minería de datos.

Resumiendo, el monitorizar y mantener actualizado un modelo proveniente de la minería de datos es una de las dificultades principales en el mantenimiento del conocimiento. Como hemos visto, también existe dificultad para fusionar conocimiento de fuentes diferentes, o para adaptar el conocimiento a situaciones cambiantes.

Sería, por tanto, deseable contar con herramientas sencillas que permitan integrar y transformar conocimiento de distintas fuentes y formatos, adaptarlo a nuevos contextos y formulaciones del problema, revisarlo ante situaciones cambiantes, etc. Contar con estos métodos nos permitirá tener una mayor reutilización de modelos y dotarles de una duración mayor.

1.2. Hipótesis de partida: El método mimético

Para dar solución a los problemas planteados en la sección anterior partimos de la hipótesis de que es posible encontrar un método basado en el aprendizaje automático para extraer reglas comprensibles a partir de un oráculo o experto (tratado como caja negra) con las características de ser general, simple, fácil de implementar y con posibilidades de poder extenderse a diversas áreas de aplicación.

Esta hipótesis se basa en el hecho de que existe una técnica para convertir cualquier modelo experto u oráculo en un conjunto de reglas que lo capturan fidedignamente.

Esta técnica fue presentada por Domingos (Domingos, 1997a, 1997b, 1998). La idea básica consiste en considerar el modelo aprendido como un oráculo y generar un nuevo conjunto de datos etiquetado con él (conjunto de datos inventados). Después, el conjunto de datos etiquetado se utiliza para aprender un árbol de decisión, tal como C4.5 (Quinlan, 1993), que genera en última instancia un modelo en forma de reglas. Domingos llamó a su método Combinación de Múltiples Modelos (CMM) porque lo aplicó exclusivamente a extraer modelos comprensibles de multiclasificadores. Independientemente, nuestro grupo

de investigación llegó a una idea similar (Estruch et al 2003, Estruch y Hernández, 2003) y, dado que básicamente consiste en mimetizar un modelo existente, se le denominó método mimético.

No se ha planteado hasta ahora que el método mimético es mucho más versátil y que podría aplicarse, por ejemplo, a la creación de sistemas expertos, a la revisión de modelos, o para adaptar modelos a costes cambiantes. La idea de usar el aprendizaje mimético en diferentes aplicaciones de adquisición, transformación, adaptación y revisión de modelos nace de la facilidad de implementación y adaptación del método sin virtualmente imponer ninguna condición adicional: no hay necesidad de los datos antiguos, no tenemos que saber su distribución original, cualquier técnica de aprendizaje computacional puede ser utilizada y cualquier modelo origen puede utilizarse. Todo esto lo hace ideal tanto para la adquisición de conocimiento como para su adaptación.

Queda, en definitiva, mucho por estudiar sobre esta técnica: cómo aplicarla a otros tipos de "oráculos", examinar la influencia del tamaño de los conjuntos de datos inventados, el efecto de la exclusión del conjunto de datos originales, la aplicación a otros problemas (contextualización) y un largo etcétera, que permitirán abordar ciertos problemas existentes en muchas áreas desde otra perspectiva.

1.3. Objetivos

El objetivo general de la tesis es estudiar y mejorar el método mimético para que permita la extracción de conocimiento, la contextualización, la revisión de conocimiento y la fusión de modelos, sin necesidad de analizar internamente la fuente de datos o el modelo a adaptar o integrar, minimizando, en su caso, las preguntas al experto y proporcionando modelos finales en forma de reglas.

A continuación, se listan los objetivos específicos perseguidos en el trabajo de tesis:

1. Objetivos específicos en el estudio y análisis del método mimético:

- Analizar los límites teóricos y caracterizar el comportamiento de la técnica mimética en base a curvas de aprendizaje para diferentes configuraciones del método, por ejemplo, con o sin los datos de entrenamiento. Esta última configuración será especialmente oportuno para la revisión de modelos y otras aplicaciones del método mimético no estudiadas con anterioridad.

- Búsqueda y prueba de diferentes métricas de calidad que permitan establecer la mejor relación entre precisión y comprensibilidad, es decir, maximizar la precisión y minimizar el número y la complejidad de las reglas resultantes.

- Análisis experimental de los diferentes factores que influyen en la técnica mimética, en comparación con los modelos teóricos y las nuevas configuraciones.

2. Objetivos específicos en las aplicaciones del método mimético:

- Propuesta de un procedimiento práctico para obtener el número óptimo de instancias que debe etiquetar el experto o conjunto de expertos para extraer el conocimiento empleando el método mimético según unos requerimientos de fidelidad al experto o precisión.

- Extender el método mimético en casos donde el contexto varíe. En primer lugar, se abordarán casos donde el formato del problema cambie, donde aparezcan o desaparezcan clases o atributos, intentando mantener la semántica del modelo original pero adaptándolo al nuevo formato.

- Desarrollar una nueva aproximación a la revisión de modelos basada en una reutilización balanceado del modelo antiguo, empleando el método de aprendizaje mimético, y la nueva evidencia.

- Extender y obtener variantes del método mimético para la contextualización de modelos en presencia de cambios de costes o ciertas distribuciones de clases.

> El objetivo es, a partir de un cambio de costes y distribución para ajustar el modelo a un nuevo contexto, obtener una reformulación del modelo que incorpore el nuevo coste y distribución, manteniendo u obteniendo comprensibilidad.

En definitiva, los objetivos específicos se centran en estudiar y mejorar las características de la técnica mimética y ampliar su abanico de aplicaciones.

La consecución de los objetivos anteriores permitiría agilizar y mejorar los procesos de adquisición, adaptación y mantenimiento del conocimiento en muchas áreas en las cuales este proceso no puede realizarse de manera automática o se trata de procesos muy costosos. Además, esto permitiría facilitar el uso de las herramientas de minería de datos y su integración con otros sistemas, la reutilización del conocimiento de sistemas expertos, y especialmente debido al hecho de poder trabajar siempre en forma de reglas, expresables en numerosos estándares existentes. El abanico de aplicaciones para las que se plantean las extensiones conllevaría una técnica genérica y potente para numerosos procesos de gestión y transformación de la información en la sociedad del conocimiento.

1.4. Organización de la tesis

Esta memoria de tesis está organizada de la siguiente forma:

El **capítulo 2** incluye los fundamentos y nociones básicas para la realización de la tesis, pasando por el descubrimiento de conocimiento a partir de bases de datos, la minería de datos, sus tareas, clasificación, árboles de decisión, sistemas de reglas, principios para ponderar la complejidad de la hipótesis con la precisión de la misma, medidas de comprensibilidad, métodos de extracción de reglas, clasificación sensible al coste y revisión de modelos.

El **capítulo 3** describe los antecedentes de esta tesis. Aquí se tratan los conocimientos relacionados con la hipótesis de trabajo: el método mimético y otros trabajos previos relacionados.

El **capítulo 4** muestra el estudio teórico del método mimético considerando como base las curvas de aprendizaje. Un tema

importante en el entrenamiento de los modelos miméticos es el tamaño del conjunto de datos que se emplea para crear el clasificador.

El **capítulo 5** presenta el análisis del método mimético considerando los datos de entrenamiento. Se analizan los factores que afectan al método mimético: poda, tamaño de los datos inventados y nivel de confianza.

El **capítulo 6** presenta la aplicación del método prescindiendo de los datos de entrenamiento originales, situación habitual cuando se trabaja con sistemas expertos. En este capítulo abordamos este problema desde una perspectiva de optimización.

Los tres capítulos siguientes abordan la contextualización de los modelos. En el **capítulo 7**, se estudia la revisión de modelos, es decir, la modificación/adaptación de los modelos para acomodarse a los cambios que se producen a lo largo del tiempo. La adaptación de los modelos ante cambios en los datos, ya sean por incremento o disminución de atributos o de valores de la clase, se aborda en el **capítulo 8**. Otros cambios de contexto relacionados con el coste y las distribuciones de clases se presentan en el **capítulo 9**.

Finalmente, el **capítulo 10** recopila diferentes conclusiones y recomendaciones, así como las contribuciones de la tesis y se proponen posibles líneas de investigación futuras.

Capítulo 2
Preliminares

El descubrimiento de conocimiento a partir de bases de datos, del inglés *Knowledge Discovery from Databases* (KDD)[1], término que surge en la década de los noventa, se definió como el "proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y en última instancia comprensibles a partir de los datos" (Fayyad et al., 1996). El KDD es un proceso interactivo (requiere en algunas etapas la intervención del usuario o de un experto) e iterativo, ya que cuenta de varias fases, algunas de las cuales pueden volver a ejecutarse a modo de bucle, tal y como se muestra en la Figura 2.1.

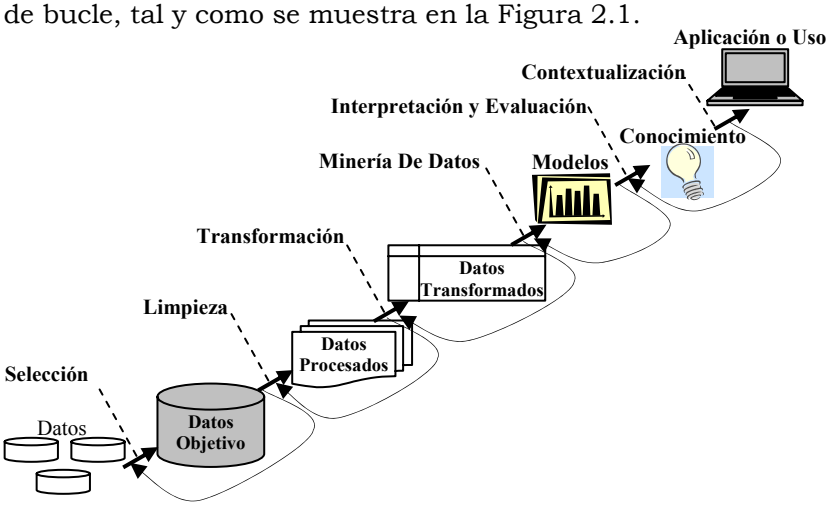


Figura 2.1 Proceso del KDD

[1] *Mucha de la información de este capítulo fue obtenida del libro Introducción a la Minería de Datos (Hernández, Ramírez y Ferri, 2004).*

Una etapa importante dentro del KDD es la minería de datos (MD). En (Witten y Frank, 2005) se define la minería de datos como el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos. Es decir, la actividad fundamental de la minería de datos es encontrar modelos inteligibles a partir de los datos. Así, los sistemas de KDD permiten la selección, limpieza, transformación y proyección de los datos; analizar los datos para extraer patrones y modelos adecuados; evaluar e interpretar los patrones para convertirlos en conocimiento; consolidar el conocimiento resolviendo posibles conflictos con conocimiento previamente extraído; y hacer el conocimiento disponible para su uso. Esta última definición extendida del proceso clarifica la relación entre el KDD y la minería de datos: el KDD es el proceso global de descubrir conocimiento útil desde las bases de datos mientras que la minería de datos se refiere a la aplicación de los métodos de aprendizaje y estadísticos para la obtención de patrones y modelos. No obstante, en muchos ámbitos, KDD y MD se vienen utilizando como sinónimos.

Dentro de la minería de datos se distinguen dos tipos de tareas: predictivas y descriptivas. Las tareas predictivas son aquellas cuyo objetivo es generar un modelo para estimar el valor de variables de interés; mientras que las tareas descriptivas son las que producen modelos que explican o resumen los datos. Entre las tareas predictivas encontramos la clasificación y la regresión, mientras que el agrupamiento (*clustering*), las reglas de asociación, y las correlaciones son tareas descriptivas.

Dado que la minería de datos es un campo muy interdisciplinar existen diferentes paradigmas detrás de las técnicas utilizadas para esta fase: técnicas de inferencia estadística, árboles de decisión, redes neuronales, inducción de reglas, aprendizaje basado en instancias, algoritmos genéticos, aprendizaje bayesiano, programación lógica inductiva y varios tipos de métodos basados en núcleos, entre otros.

De igual forma, para una misma técnica o paradigma pueden existir diferentes algoritmos; veamos algunas muestras de ello. Las técnicas basadas en árboles de decisión y sistemas de aprendizaje de reglas, se basan en dos tipos de algoritmos: los algoritmos denominados "divide y vencerás", como el ID3

(Quinlan, 1983, 1986), el C4.5 (Quinlan, 1993), o el CART (Breiman et al., 1984), y los algoritmos denominados "separa y vencerás", como el CN2 (Clark y Niblett, 1989; Clark y Boswell, 1991).

Otro ejemplo lo encontramos en las técnicas basadas en redes neuronales artificiales, que consisten en aprender un modelo mediante el entrenamiento de los pesos que conectan un conjunto de nodos o neuronas. En este caso la topología de la red y los pesos de las conexiones determinan el patrón aprendido, existiendo innumerables variantes de organización: perceptrón simple (Rosenblatt, 1962), redes multicapa (McClelland et al., 1986), redes de base radial (Shepard, 1968; Powell, 1987), redes de Kohonen (Kohonen 1982 y 1984), etc., con no menos algoritmos diferentes para cada organización, siendo el más conocido el de retropropagación (*backpropagation*).

Las técnicas bayesianas se basan en estimar la probabilidad de pertenencia (a una clase o grupo), mediante la estimación de las probabilidades condicionales inversas o a priori, utilizando para ello el teorema de Bayes. Algunos algoritmos muy populares son el clasificador bayesiano naive (Duda y Hart, 1973; Langley et al., 1992; Zheng y Webb, 2005), los métodos basados en máxima verisimilitud y el algoritmo *Expectation Maximization* (EM) (Dempster et al., 1977; McLachlan y Kirshnan, 1996).

Finalmente, las técnicas de combinación de modelos consisten en generar una cierta cantidad de modelos simples bien de forma independiente o bien encadenadamente. Existen, por lo tanto, muchos modos de generar y combinar conjuntos de modelos. Ejemplos de métodos multiclasificadores que crean modelos independientes y combinan las predicciones son *Bagging* (Breiman, 1996; Quinlan, 1996), *Boosting* (Freund y Schapire, 1996; Quinlan, 1996) y *randomisation* (Dietterich, 2000) y para modelos encadenados tenemos *Cross-validated Committees* (Parmanto et al., 1996), *Stacking* (Wolpert, 1992) y *Cascading* (Gama y Bradzil, 2000).

Dada esta gran variedad de técnicas y algoritmos es posible generar diferentes modelos para un mismo problema. Por ello, algunas de las cuestiones más trascendentes dentro del KDD son la evaluación de los modelos y su adaptación a nuevas situaciones.

En este capítulo describimos los conceptos fundamentales en que se basa esta tesis, los cuales están relacionados con las últimas etapas del KDD: la minería de datos, la evaluación y contextualización (actualización y revisión) de modelos. Como nuestra tarea fundamental es la clasificación en forma de conjuntos de reglas, en la primera sección presentamos la técnica más utilizada para este fin: los árboles de decisión. A continuación, se presentan las técnicas de evaluación que emplearemos en los experimentos de los capítulos siguientes. En la sección 3 describimos otras medidas de la calidad de los modelos que están relacionadas con la comprensibilidad. Finalmente, ya dentro de la última etapa del KDD, presentamos la contextualización de modelos, en concreto la contextualización debida al cambio de formato, la clasificación sensible al coste y, por último, la revisión de modelos ante la presencia de nuevos datos.

2.1. Árboles de decisión

El aprendizaje de árboles de decisión es uno de los métodos proposicionales de obtención de modelos comprensibles más utilizados en clasificación. El término "modelo" indica que esta técnica construye una "hipótesis", "explicación" o "representación" de las regularidades existentes en los datos. El término "proposicional" hace referencia a que aprende los modelos a partir de una única tabla de datos y que no establece relaciones entre más de una fila de la tabla a la vez ni sobre más de un atributo a la vez. A menudo, a las técnicas proposicionales se les denomina "atributo-valor" dado que las condiciones se expresan sobre el valor de un único atributo. El término "comprensibles" hace referencia al hecho de que estos modelos se pueden expresar de una manera simbólica en forma de conjunto de condiciones y, por lo tanto, pueden tener como resultado modelos inteligibles para los seres humanos (y también para sistemas semi-automáticos que procesen reglas).

Un árbol de decisión es un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que la decisión final a tomar se puede determinar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas. Los árboles de decisión se utilizan desde hace siglos, y son especialmente apropiados para expresar procedimientos médicos, legales, comerciales, estratégicos, matemáticos, lógicos, etc.

Una de las grandes ventajas de los árboles de decisión es que, en su forma más general, las opciones posibles a partir de una determinada condición son excluyentes. Esto permite analizar una situación y, siguiendo el árbol de decisión apropiadamente, llegar a una sola acción o decisión a tomar.

Consideremos, por ejemplo, la clasificación de plantas iris. Este problema consiste en determinar de qué tipo de planta se trata (iris setosa, iris versicolor o iris virginica) en base a las características del ancho y largo de su pétalo y/o sépalo. En la Figura 2.2, se muestra una representación en forma de árbol de decisión para este problema.

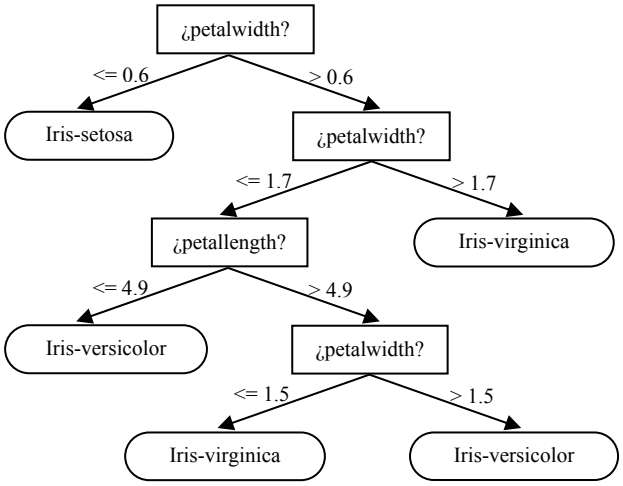


Figura 2.2. Árbol de decisión para determinar el tipo de plantas iris

Este árbol de decisión en concreto funciona como un "clasificador", es decir, dada una nueva planta nos la clasifica en una de las clases posibles, para lo cual basta con recorrer el árbol desde la raíz hasta alguna de sus hojas, en función del valor de los atributos del dato a clasificar.

Los algoritmos de aprendizaje de árboles de decisión más habituales se llaman algoritmos de partición o algoritmos de "divide y vencerás". Básicamente, el algoritmo va construyendo el árbol (desde el árbol que sólo contiene la raíz) añadiendo particiones y los hijos resultantes de cada partición. Lógicamente, en cada partición, los ejemplos se van dividiendo entre los hijos. Finalmente, se llega a la situación en la que todos los ejemplos que

caen en los nodos inferiores son de la misma clase y esa rama ya no sigue creciendo.

En la Figura 2.3 se puede observar un algoritmo básico para generar un árbol de decisión a partir de un conjunto de ejemplos, utilizando la técnica de "partición".

```
ALGORITMO Partición (N:nodo, E:conjunto de ejemplos)
 SI todos los ejemplos E son de la misma clase c ENTONCES
     Asignar la clase c al nodo N.
     SALIR // Esta rama es pura, ya no hay que seguir partiendo. N es hoja.
 SI NO
     Particiones := generar posibles particiones.
     MejorPartición := seleccionar la mejor partición según el criterio de partición.
  PARA CADA condición i de la MejorPartición HACER
     Añadir un nodo hijo i a N y asignar los ejemplos consistentes a cada hijo (Ei).
     Partición(i, Ei).  // Realizar el mismo procedimiento global con cada hijo.
  FIN-PARA
 FIN-SI
FIN-ALGORITMO

Para generar un modelo con un conjunto de ejemplos E, se invoca con la llamada
Partición(R,E), donde R es un nodo raíz de un árbol por empezar.
```

Figura 2.3 Algoritmo de aprendizaje de árboles de decisión por "Partición" (Divide y Vencerás)

Como puede verse, una característica importante de estos algoritmos es que una vez elegida la partición ya no se puede cambiar, aunque más tarde se pudiera comprobar que ha sido una mala elección. Por tanto, uno de los aspectos más importantes a considerar en estos sistemas es el denominado criterio de partición, ya que una mala elección de la partición (especialmente en las partes superiores del árbol) generará un árbol peor.

Por lo tanto, las particiones a considerar y el criterio de selección de particiones son lo que diferencian fundamentalmente a los distintos algoritmos de "partición" existentes hasta la fecha, como CART (Breiman et al., 1984), ID3 (Quinlan, 1983; Quinlan, 1986), C4.5 (Quinlan, 1993), ASSISTANT (Cestnik et al., 1987), etc.

Tal y como hemos mencionado, otra característica interesante es que los árboles de decisión se pueden expresar como conjuntos de reglas de la forma **SI** conjunción de condiciones **ENTONCES** acción. Estas reglas se obtienen al recorrer cada uno

de los caminos que llevan desde el nodo raíz hasta un nodo hoja, donde las particiones son las condiciones de las reglas. Por ejemplo, el árbol de la Figura 2.2 se puede expresar como el conjunto de reglas siguientes:

```
SI petalwidth <= 0.6 ENTONCES Iris-setosa
SI NO // petalwidth > 0.6
|  SI petalwidth <= 1.7
|  |  SI petallength <= 4.9 ENTONCES Iris-versicolor
|  |  SI NO // petallength > 4.9
|  |  |  SI petalwidth <= 1.5 ENTONCES Iris-virginica
|  |  |  SI NO Iris-versicolor    // petalwidth > 1.5
|  SI NO Iris-virginica           // petalwidth > 1.7
```

Como hemos mencionado anteriormente, las particiones son un conjunto de condiciones exhaustivas y excluyentes. Cuantas más particiones permitamos más expresivos podrán ser los árboles de decisión generados y, probablemente, más precisos. Además, más posibilidades tendremos de encontrar los patrones que hay detrás de los datos. No obstante, cuantas más particiones elijamos, la complejidad del algoritmo será mayor. Por tanto, la calidad de un algoritmo está determinada por el compromiso entre expresividad y eficiencia.

Debido a esto, la mayoría de los algoritmos de aprendizaje de árboles de decisión sólo permiten un juego muy limitado de particiones. Así, el C4.5 (Quinlan, 1993) contiene un solo tipo de partición para los atributos nominales y un solo tipo de partición para los atributos numéricos:

- Particiones nominales: una condición de la forma $x_i = v_j$, $1 \leq j \leq k$, siendo x_i el atributo nominal seleccionado y v_j cada uno de sus k posibles valores.
- Particiones numéricas: las particiones numéricas admitidas son de la forma $\{x_i \leq a,\ \ x_i > a\}$ donde x_i es un atributo numérico y continuo, y a es una constante numérica que se obtiene observando valores consecutivos del atributo y calculando sus puntos medios.

La expresividad resultante de las particiones anteriores se conoce como expresividad proposicional cuadricular ya que parte el espacio del problema mediante segmentos paralelos a los ejes, véase, por ejemplo, la Figura 2.4, para un problema bidimensional.

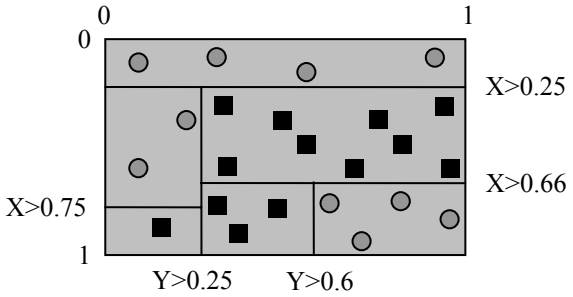


Figura 2.4. Partición cuadricular del espacio obtenido por un árbol de decisión para un problema con dos atributos (X e Y) numéricos.

Así, en problemas de clasificación, un modelo es (implícitamente o explícitamente) una división del espacio de instancias en regiones, cada una de las cuales se les asigna una determinada clase de las posibles. Dado un conjunto de ejemplos de entrenamiento del cual aprendemos un modelo, éste es preciso si las regiones que produce coincide con las regiones verdaderas de esos ejemplos; el algoritmo produce modelos estables si produce las mismas regiones al proporcionarle dos diferentes conjuntos de entrenamiento del mismo dominio; y los modelos son comprensibles en cuanto indica claramente (a un usuario humano) cuales son las regiones de la clase.

Basándose en la idea de buscar particiones que discriminen o que consigan nodos más puros (donde un nodo puro es aquel en el que los ejemplos que caen en él son todos de la misma clase), se han presentado en las últimas dos décadas numerosos criterios de partición, tales como el criterio del error esperado, el criterio Gini (Breiman et al., 1984), los criterios Gain, Gain Ratio y el criterio usado en el algoritmo C4.5 (Quinlan, 1993) y el DKM (Kearns y Mansour 1996). Estos criterios de partición buscan la partición s con la menor impureza $I(s)$, definida de la siguiente forma:

$$I(s) = \sum_{j=1..n} p_j f(p_j^1, p_j^2, ..., p_j^c)$$

donde n es el número de nodos hijos de la partición (número de condiciones de la partición), p_j es la probabilidad de que un ejemplo "caiga" en el nodo j, y p^i_j es la proporción de elementos de la clase i en el nodo j, siendo c el número de clases. La función $f(\ )$ se denomina función de impureza y, por tanto, la función $I(s)$ calcula la media ponderada (dependiendo de la cardi-

nalidad de cada hijo) de la impureza de los hijos en una partición. Bajo esta fórmula general, cada criterio de partición implementa una función *f* distinta, como se muestra en la Tabla 2.1:

Tabla 2.1 Funciones de impureza para algunos criterios de partición.

Criterio	$f(p_j^1, p_j^2, ..., p_j^c)$
Error Esperado	$\min(p^1, p^2, ..., p^c)$
GINI (CART)	$1-\sum(p_j)^2$
Entropía (gain)	$\sum p_j \log(p_j)$
DKM	$2(\prod p_j)^{1/2}$

Existen muchas variantes de estos criterios (por ejemplo el GainRatio y el criterio usado en el C4.5 son variantes de la entropía), la ortogonalidad de GID3 (Fayyad, 1994), y muchos otros que no se ajustan a la fórmula anterior (por ejemplo criterios basados en el principio MDL (*Minimum Description Length*) (Ferri et al. 2001) o criterios basados en el AUC (*Area Under the ROC Curve*) (Ferri et al. 2002). Según los experimentos realizados durante los últimos años, parece ser que tanto el criterio usado por el C4.5 (GainRatio modificado) como el criterio DKM se comportan ligeramente mejor que el GINI y bastante mejor que el error esperado.

2.1.1. Poda

Cuando el modelo se ajusta completamente a los datos de entrenamiento, surge un problema de sobreajuste (*overfitting*). En este caso el modelo será demasiado específico, poco general, y por tanto, no se comportará bien con otros datos no vistos. Este problema es más visible cuando los datos contienen ruido (errores en los atributos o incluso en las clases), ya que el modelo intentará ajustarse a los errores y esto perjudicará el comportamiento global del modelo aprendido.

La forma más frecuente de limitar este problema del sobreajuste en los métodos de aprendizaje de árboles de decisión es utilizar la poda, es decir, los algoritmos de aprendizaje incorporan algún mecanismo para eliminar algunas condiciones de las ramas del árbol o de algunas reglas. Atendiendo al momento en que esa eliminación es hecha, distinguimos los siguientes métodos de poda:

- Prepoda: el proceso se realiza durante la construcción del modelo. En realidad se trata de un criterio de parada a la hora de seguir especializando una rama o una regla. En general, los criterios de prepoda pueden estar basados en el número de ejemplos por nodo, en el número de excepciones respecto a la clase mayoritaria (error esperado) o en técnicas más sofisticadas, como el criterio *Minimum Description Length* o el principio *Minimun Message Length* que hablaremos más adelante.

- Pospoda: el proceso se realiza después de la construcción del modelo. Los criterios son los mismos que en la prepoda, la diferencia es que la pospoda tiene una visión completa del modelo. Por ello, la pospoda obtiene mejores resultados, aunque es menos eficiente que la prepoda.

Asímismo, se pueden combinar técnicas de prepoda y de pospoda. Por ejemplo, el C4.5, utiliza una prepoda por cardinalidad (los nodos con una cardinalidad inferior a una cierta constante no se explotan) y una pospoda basada en un criterio más sofisticado. Debido a esta situación es posible que algunos nodos hoja ya no sean puros, es decir, es posible que tengan ejemplos de varias clases. Normalmente se elegirá la clase con más ejemplos para etiquetar el nodo hoja.

2.2. Técnicas de evaluación de hipótesis

La medida tradicional para evaluar clasificadores es el error, el cual se define como el porcentaje de instancias mal clasificadas. El concepto complementario al error es la precisión o el porcentaje de acierto (*accuracy*), que se obtiene dividiendo el número de clasificaciones correctas por el número total de instancias.

Para garantizar que el acierto del modelo es una medida independiente de los datos que generaron ese modelo, se debe entrenar y probar el modelo con dos conjuntos de datos diferentes: el conjunto de entrenamiento (*training set*) y el conjunto de prueba o de *test* (*test set*). Si no se usan conjuntos diferentes de entrenamiento y prueba la precisión del modelo será sobreestimada, es decir, tendremos estimaciones muy optimistas.

En los modelos predictivos, el uso de esta separación entre entrenamiento y prueba es fácil de interpretar. Por ejemplo, para una tarea de clasificación, después de generar el modelo con el conjunto de entrenamiento, se puede usar para predecir la clase de los datos de *test*. La precisión obtenida es una buena estimación de cómo se comportará el modelo para datos futuros similares a los de *test*. Esta forma de proceder no garantiza que el modelo sea correcto, sino que simplemente indica que si usamos el mismo modelo y unos datos similares a los de prueba, la precisión media será bastante parecida a la obtenida con éstos.

El método de evaluación más básico, la validación simple, reserva un porcentaje de los datos como conjunto de *test*, y no lo usa para construir el modelo. Este porcentaje suele variar entre el 5% y el 50%. La división de los datos en estos dos grupos debe ser aleatoria para que la estimación no esté sesgada.

Si tenemos una cantidad no muy elevada de datos para construir el modelo, puede que no podamos permitirnos el lujo de reservar parte de los mismos para la etapa de evaluación. En estos casos se usa un método conocido como validación cruzada (*cross validation*) (Stone, 1978). En la versión más básica los datos se dividen aleatoriamente en dos conjuntos de igual tamaño, con los que se estima la precisión predictiva del modelo. Para ello, primero se construye un modelo con el primer conjunto y se usa para predecir los resultados en el segundo conjunto y calcular así un error (o de precisión). A continuación, se construye un modelo con el segundo conjunto y se usa para predecir los resultados del primer conjunto, obteniéndose un segundo porcentaje de error. Finalmente, se construye un modelo con todos los datos, se calcula un promedio de los porcentajes de error y se usa para estimar mejor su precisión.

El método que se usa normalmente es la validación cruzada con n pliegues (*n-fold cross validation*). En este método los datos se dividen aleatoriamente en n grupos. Un grupo se reserva

para el *test* y con los otros $n-1$ restantes (juntando todos sus datos) se construye un modelo y se usa para predecir el resultado de los datos del grupo reservado. Este proceso se repite n veces, dejando cada vez un grupo diferente para la prueba. Esto significa que se calculan n porcentajes de error independientes. Finalmente, se construye un modelo con todos los datos y se obtiene (para determinar la significancia desde un punto de vista estadístico de los resultados obtenidos) sus porcentajes de error y precisión promediando los n porcentajes de error disponibles.

Además de las mencionadas, existen otras técnicas de validación (como por ej. *Bootstrapping* (Efron, 1982; Ripley, 1996)). De hecho, no hay un acuerdo unánime en lo relativo al diseño experimental en problemas de aprendizaje automático y existe un debate reciente al respecto (Bouckaert y Frank, 2004; Bouckaert 2003, 2004 y 2005). La opción más difundida sigue consistiendo en combinar la validación cruzada con un *test* estadístico del tipo t de *Student* (para determinar la significancia desde un punto de vista estadístico de los resultados obtenidos), sin embargo, el número de particiones que se debe elegir no está bien definido. En (Herrera, 2004; Bouckaert 2005) se muestran que los resultados experimentales basados en 10 particiones son bastante buenos, motivo por el cual usaremos validación cruzada (generalmente con 10 ejecuciones) en los experimentos.

Existen otras métricas de evaluación que no se basan en el error (acierto) cometido. Estas métricas de evaluación son muy populares en el área de la recuperación de información (Salton y McGill, 1983; Baeza-Yates y Ribeiro-Neto, 1999) y son la precisión (*precision)* y el alcance (*recall*).

La *precision* mide la probabilidad de que si un sistema clasifica un documento en una cierta categoría, el documento realmente pertenezca a la categoría.

Por otro lado, el alcance mide la probabilidad de que si un documento pertenece a cierta categoría, el sistema lo asigne a dicha categoría.

Para problemas binarios, la precisión y el alcance se obtienen a partir de la matriz de confusión (o tabla de contingencia). La matriz de confusión muestra el recuento de casos de las clases predichas (estimadas) y su valor real. En el caso de dos clases

(que se suelen denominar positiva (*P*) y negativa (*N*)), la matriz de confusión se puede expresar como

	Real	
Estimado	*TP*	*FP*
	FN	*TN*

donde *TP* denota el número de casos positivos que han sido predichos como tal (*True Positive*), *FP* denota el número de casos positivos que han sido predichos como negativos (*False Positive*), *FN* los casos positivos que han sido estimados negativos (*False Negative*) y finalmente, *FP* son los casos negativos predichos como tales (*True Positive*).

Por lo tanto, *TP* + *TN* representa los aciertos del sistema, *FN* + *FP* son los errores, y la suma de las cuatro celdas equivale al número total de ejemplos. Los valores de la matriz de confusión permiten estimar varias métricas mostradas en la Tabla 2.2.

Tabla 2.2 Las métricas basadas en la matriz de confusión para problemas binarios.

Probabilidad	**Fórmula**	**Métrica (sinónimos)**
$p(\mathrm{P}\mid\mathrm{p}) \approx$ *True Positive Rate*	$TPR = \frac{TP}{TP+FN}$	*recall* *sensitivity* *positive accuracy*
$p(\mathrm{N}\mid\mathrm{p}) \approx$ *False Negative Rate*	$FNR = \frac{FN}{TP+FN}$	*positive error*
$p(\mathrm{N}\mid\mathrm{n}) \approx$ *True Negative Rate*	$TNR = \frac{TN}{TN+FP}$	*specificity* *negative accuracy*
$p(\mathrm{P}\mid\mathrm{n}) \approx$ *False Positive Rate*	$FPR = \frac{FP}{TN+FP}$	*negative error*
$p(\mathrm{p}\mid\mathrm{P}) \approx$ *Positive Predictive Value*	$PPV = \frac{TP}{TP+FP}$	*precision*
$p(\mathrm{n}\mid\mathrm{N}) \approx$ *Negative Predictive Value*	$NPV = \frac{TN}{TN+FN}$	

Como puede observarse, los conceptos de "*precision*" (en ingles) y "*accuracy*" no son exactamente iguales. Dado que en esta

tesis no vamos a trabajar con el concepto de "*precision*", usaremos el vocablo castellano precisión para referirnos al acierto ("acurracy"), algo por otro lado, bastante habitual.

Todas las medidas incluidas en la Tabla 2.2 se basan en que las prediciones son discretas. Sin embargo, es posible usar medidas probabilísticas siempre que las predicciones vengan acompañadas con un valor de probabilidad o de confianza.

Un clasificador discreto (*crisp*) predice una clase entre las posibles. En contra parte, un clasificador probabilístico (suave o en inlgés *soft*) predice una clase, pero acompaña a cada predicción con un valor de fiabilidad. La mayoría de los métodos de aprendizaje en minería de datos pueden acompañar las predicciones con estos valores de fiabilidad. Un tipo especial de clasificador *soft* son los estimadores de probabilidad, en este caso, en vez de predecir la clase "a", "b" o "c", dan estimaciones de probabilidad para cada clase, es decir, p_a, p_b y p_c. Por ejemplo, al aplicar un clasificador 1 sobre un ejemplar proporciona las siguientes probabilidades: $p_a = 0.2$, $p_b = 0.5$ y $p_c = 0.3$, de forma similar se utiliza un clasificador 2 y se obtienen las probabilidades: $p_a = 0.3$, $p_b = 0.4$ y $p_c = 0.3$. Como se observa los dos clasificadores predice la clase "b", sin embargo, el clasificador 1 esta más "seguro" de su predicción.

En el caso específico de problemas de dos clases, si $p_a = x$, entonces $p_b = 1 - x$, de esta forma, sólo es necesario especificar la probabilidad de una clase. Llamemos a una clase 0 (negtiva) y a la otra clase 1 (positiva). Un ranker es un clasificador suave que proporciona un valor entre 0 y 1 de la probabilidad de una de las clases. Este valor se denomina también "*score*" y determina si está más cerca de la clase 0 o de la clase 1. Por ejemplo, la probabilidad de que un cliente compre un producto o la probabilidad de que un correo sea *spam*.

Así pues, los valores de probabilidad de los clasificadores suaves se pueden utilizar para definir otras métricas. Por ejemplo, el error cuadrático medio (*Mean Squared Error*, MSE) (también conocido para dos clases como *Brier score* (Brier, 1950)) es una métrica que nos dice cuánto se desvían las predicciones de la probabilidad verdadera. Esta medida se calcula como:

$$MSE = \frac{\sum_{i=1}^{m}\sum_{j=1}^{c}(f(i,j)-p(i,j))^2}{m \cdot c}$$

donde m es el número de ejemplos, c el número de clases y $f(i,j)$ (respectivamente $p(i,j)$) representa la probabilidad real (respectivamente, predicha) de que el ejemplo i sea de la clase j. Asumimos que las probabilidades están un rango de {0,1}.

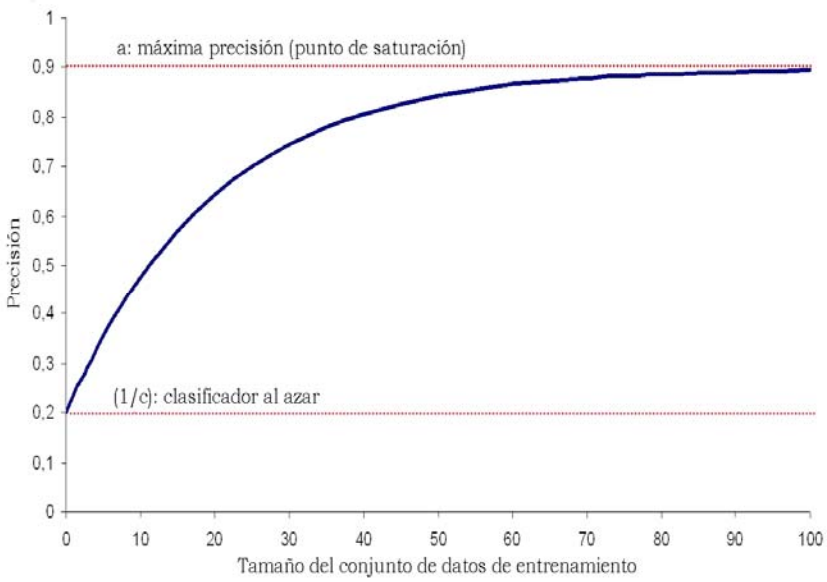
Otro ejemplo de métrica con perspectiva probabilística es el LogLoss (también conocida como *cross entropy*), la cual es una medida de la bondad de las probabilidades estimadas (Good, 1952, 1968; Dowe et al., 1996), y que se define como

$$LogLoss = \frac{-\sum_{i=1}^{m}\sum_{j=1}^{c}(f(i,j)\log_2 p(i,j))}{m}$$

El análisis ROC permite también evaluar los modelos aprendidos de manera más independiente respecto al contexto de coste mediante el valor AUC (área bajo la curva ROC), este tema lo veremos más adelante.

2.2.1. Análisis en función del tamaño del conjunto de aprendizaje: curvas de aprendizaje

El resultado de una técnica de aprendizaje, según una deteminada métrica de calidad (precisión, MSE, LogLoss, ...) varía, generalmente de forma positiva, cuantos más ejemplos de aprendizaje tengamos. Las curvas de aprendizaje permiten representar el rendimiento de una técnica de aprendizaje para distintos tamaños del conjunto de entrenamiento. En la siguiente figura vemos el comportamiento (en precisión) de una técnica de aprendizaje para distintos tamaños del conjunto de entrenamiento.



No es difícil considerar un algoritmo particular de aprendizaje y aprender uno o más modelos con incrementos del tamaño del conjunto de datos de entrenamiento. En muchos casos, la función que define la curva de precisión puede adecuarse bien al comportamiento del algoritmo y del conjunto de datos, permitiendo obtener los parámetros que la describen y así poder extrapolar para conjuntos de datos más grandes, además de poder realizar comparaciones entre diferentes algoritmos y modelos.

Esto perimite ver si un algoritmo es capaz de aprender modelos buenos con pocos datos o, por el contrario, cuantos datos son necesarios para obtener una buena precisión.

2.2.2. Evaluación basada en la complejidad de la hipótesis

Los criterios de validación y las métricas de evaluación vistas en el apartado anterior permiten, en cierto modo, la selección entre varios modelos ya aprendidos. No obstante, no proporcionan ninguna ayuda a la hora de obtener estos modelos, ni establecen qué criterios generales deberían cumplir los mismos.

El problema de seleccionar una teoría o modelo de entre un conjunto de teorías plausibles ha sido centro de discusión filosófica a lo largo de la historia. Una de las opciones más defendida ha sido el preferir las teorías más simples. En concreto, el principio que elije, en igualdad de condiciones, la hipótesis más simple, se le conoce como criterio de la "navaja de Occam" (*Occam's Razor*), debido a que fue promulgada por el filósofo medieval William de Occam. Además de las razones epistemológicas aducidas sobre la bondad de este criterio, hemos de aceptar la razón metodológica: entre dos hipótesis igualmente válidas, la más simple será más fácil de entender, verificar y usar. El problema de la navaja de Occam es que el concepto de "igualdad de condiciones" es subjetivo y, en muchos casos, podemos llegar a hipótesis demasiado simples, del estilo: "todo vale", "todo de la clase a", etc. Es decir, que el principio ha de entenderse como que hay que buscar la teoría más simple siempre que los datos no justifiquen hacerla más compleja.

En este sentido, para determinar cuál es la hipótesis más simple que explica o captura unos datos se puede usar el principio de la longitud de descripción mínima (*Minimum Description Length principle*, MDL) (Rissanen 1978; Li y Vitányi 1997) o el principio de longitud mínima de mensaje (*Minimum Message Length principle*, MML) (Wallace y Boulton, 1968)[2].

Cualquiera de las dos teorías consiste en calcular el tamaño en bits (unidad de información) de la descripción de una hipótesis más la descripción de los ejemplos que no son cubiertos por ella (excepciones).

Formalmente, el principio MDL/MML recomienda seleccionar la hipótesis h que minimice la expresión:

$$K(h) + K(D \mid h)$$

donde $K(h)$ es la complejidad en bits de describir la hipótesis h y $K(D \mid h)$ es la complejidad en bits de describir la evidencia D a partir de la hipótesis h (lo que incluye las excepciones).

La moraleja es que una hipótesis simple con muchas excepciones no es adecuada, al igual que tampoco es adecuada una

2 Ambos criterios difieren ligeramente (Wallace y Dowe, 1999), pero la determinación del costo de representación de un modelo de clasificación y los datos que no pueden ser clasificados por él, son similares y podemos usar ambos criterios de manera indistinta.

hipótesis compleja sin excepciones. La idea del principio MDL/MML es buscar el compromiso entre ambas situaciones.

La justificación del principio anterior puede razonarse de muchas maneras, pero la más usual es a través de un razonamiento bayesiano e informacional. Veamos; en muchos casos, nos interesa obtener la hipótesis más probable h a partir de un conjunto de datos D, es decir la hipótesis con mayor $p(h|D)$, también llamada hipótesis máxima a posteriori (MAP). Una manera de seleccionar la hipótesis MAP es utilizar el teorema de Bayes. Formalmente, una hipótesis h_{MAP} es una hipótesis MAP de un conjunto de hipótesis H si:

$$\begin{aligned} h_{MAP} &\equiv \arg\max_{h \in H} p(h \mid D) \\ &= \arg\max_{h \in H} \frac{p(D \mid h)p(h)}{p(D)} \\ &= \arg\max_{h \in H} p(D \mid h)p(h) \end{aligned}$$

donde en la primera igualdad aplicamos el Teorema de Bayes y en la segunda ignoramos $p(D)$ ya que es una constante independiente de h. Tomando logaritmos en base 2:

$$\begin{aligned} h_{MAP} &\equiv \arg\max_{h \in H}(\log_2 p(D \mid h) + \log_2 p(h)) \\ &= \arg\min_{h \in H}(-\log_2 p(D \mid h) - \log_2 p(h)) \end{aligned}$$

Usando la teoría de la información de Shannon que nos dice que la cantidad de información K se puede obtener como $-\log_2 p$ siendo p la probabilidad, tenemos

$$h_{MAP} = \arg\min_{h \in H}(K(D \mid h) + K(h)) \qquad (2.1)$$

Esta última expresión corresponde al principio MDL/MML.

El principio MDL es, por lo tanto, un método que nos permite seleccionar una hipótesis de acuerdo con su complejidad y la información que se requiere para describir la evidencia a partir de esa hipótesis (excepciones).

Adicionalmente, tanto el principio MDL como el MML proporcionan un marco teórico para resolver el problema del sobre-

ajuste al permitir evaluar los modelos basándose en la complejidad o tamaño y no exclusivamente en su precisión.

2.3. Otro criterio de evaluación: la comprensibilidad

Como ya hemos mencionado, hay buenas razones para preferir modelos más simples: son más fáciles de entender, recordar y usar. En un principio podríamos suponer que cuanto más simple es un modelo, entonces es más comprensible. Sin embargo, la simplicidad y la comprensibilidad no son siempre equivalentes. Por ejemplo, una tabla de decisión puede ser más grande que un árbol de decisión con precisión semejante, pero puede ser más fácilmente entendida porque todas las líneas en la tabla utilizan los mismos atributos (Langley, 1996; Kohavi y Sommerfield, 1998). Los modelos inducidos son también más comprensibles si son consistentes con el conocimiento anterior, incluso si éste se hace más complejo (Pazzani, 1991; Pazzani et al., 1997). Qué hace exactamente a un modelo comprensible es en gran parte dependiente del dominio del problema, pero también es una cuestión de investigación cognitiva y empírica (Kononenko, 1990; Pazzani, 1991; Kohavi y Sommerfield, 1998).

Así pues, el grado de comprensibilidad de un modelo es un factor subjetivo, ya que depende en gran medida de la experiencia y conocimiento de los usuarios. Aún así, existe cierto consenso en algunos aspectos relativos a la comprensibilidad. Los sistemas de reglas son considerados como una de las representaciones que permiten comprender más fácilmente el comportamiento de un modelo (Fayyad et al., 1996). Además, tienen la ventaja de que las reglas se expresan utilizando directamente los propios atributos del problema.

Sin embargo, es necesario considerar que un modelo formado por un número elevado de reglas, y cada una de ellas con multitud de condiciones sobre los valores de los atributos, puede ser más difícil de comprender que una pequeña red neuronal que en un principio podría parecernos un modelo “poco comprensible”. Por lo tanto, también es necesario tener en cuenta el tamaño de los modelos, el cual viene determinado por las características del conjunto de reglas, por ejemplo, el número de reglas y la cantidad de condiciones por regla (Tickle et al., 1998).

A continuación presentamos diferentes medidas de la comprensibilidad de un conjunto de reglas que tienen en cuenta estas características.

Lu et al. (1995) proponen una función que involucra el número de reglas extraídas (*R*) y el número total de condiciones (*C*) en las mismas. Esta métrica recibe el nombre de complejidad del conjunto de reglas y se calcula como

$$Complejidad = 2 * R + C \qquad (2.1)$$

La aptitud (*fitness*) (Santos et al., 2000) es un valor que combina tanto precisión como comprensibilidad de un conjunto de reglas. Para realizar esta combinación se debe normalizar la métrica de complejidad, de tal forma que se obtengan valores entre 0 y 1. La "comprensibilidad" se calcula mediante

$$Comp = 1 - \frac{2 + \frac{R}{Max_R} + \frac{C}{Max_C}}{3} \qquad (2.2)$$

donde *Max_R* es el número máximo de reglas para alguna clase, entre todas la generadas hasta ahora, y *Max_C* es el número máximo de condiciones de una clase individual. Finalmente la función de aptitud se calcula con

$$Aptitud = 1 - W_{acc} Acc + W_{comp} Comp \qquad (2.3)$$

donde W_{acc} y W_{comp} son pesos definidos por el usuario para la precisión (*Acc*) y la comprensibilidad (*Comp*), respectivamente. Los mejores modelos son los que tienen los valores menores de esta función de aptitud.

Como modificación de la propuesta anterior, para expresar la complejidad de un sistema de reglas, y por lo tanto la comprensibilidad, (Engelbrecht y Schoeman, 2002) proponen que aparte del número de reglas, se emplee el número promedio de condiciones por reglas.

Los principios vistos en la subsección 2.2.1 también son válidos para encontrar un compromiso entre la comprensibilidad y

la precisión. Esto es debido a que la precisión esta definida por el error en el principio MML el cual determina el tamaño del mensaje de los datos dado el modelo y el número de reglas (comprensibilidad) del modelo es la longitud del mensaje del modelo.

El concepto de comprensibilidad es pues esquivo, difícil de definir y subjetivo, dependiendo del área y contexto en que se utilice. No existe un marco general de referencia para medirla y tampoco una medida precisa. Debido a que no se ha demostrado que las fórmulas presentadas anteriormente estén correlacionadas con juicios del usuario acerca de la comprensibilidad, utilizaremos aproximaciones basads en el criterio MDL/MML para determinar el tamaño óptimo del conjunto de datos inventados que maximiza la precisión y la comprensibilidad del modelo mimético.

2.4. Contextualización

Definimos contexto como un conjunto de circunstancias que rodean o condicionan la aplicación del conocimiento. Entendemos por contextualización de un modelo la adaptación de éste a un contexto diferente al usado para generar el modelo.

En esta tesis centraremos el estudio en tres clases de contextos:

- Los contextos debidos a cambios en el formato de los datos, inducidos por un incremento o disminución del número de atributos o bien de los valores de la clase.
- Los contextos cuando aparecen nuevos datos de un área no considerada originalmente o simplemente aparecen nuevos datos con los que el modelo antiguo empieza a fallar. Esto se conoce como revisión de teorías.
- Los contextos debidos a los cambios en la matriz de costes. Es decir, la clasificación sensible al coste.

2.4.1. Contextualización por cambios de formato

Los cambios de formato pueden ser debidos a la aparición o eliminación de un atributo. Cuando un modelo se tiene que contextualizar por la aparición de un nuevo atributo (presente

en los nuevos datos), es necesario determinar el valor de dicho atributo para los datos originales, dado que deseamos conservarlos. De esta forma, una vez homogenizados los datos, podemos reentrenar el modelo. Para determinar los valores del nuevo atributo en el conjunto de datos inicial podemos proceder de las siguientes formas: intentar determinar el valor manualmente (en el caso de que no haya muchos) o automáticamente asignando un valor de tendencia central. En concreto para el caso de valores numéricos asignar la media y en el caso de valores nominales se puede utilizar la moda.

Una manera más sofisticada de estimar el valor es predecirlo a partir de los nuevos ejemplos (esto se llama a veces "imputación de datos perdidos"), utilizando cualquier técnica predictiva de aprendizaje automático (clasificación o regresión).

Mucho más simple es cuando desaparece un atributo de los datos. En este caso, se elimina este atributo de los datos originales y luego se aprende el modelo adaptado usando los datos nuevos con los originales modificados.

Para estos escenarios es fácil encontrar una solución, pero la cuestión se suscita cuando los datos originales no existen y aparece un nuevo atributo. En esta tesis propondremos una solución a este problema.

2.4.2. Contextualización al coste

Como ya se mencionó en el capítulo 1, la clasificación en presencia de costes que varían de acuerdo a diversos tipos de errores de clasificación es importante para aplicaciones prácticas, incluyendo muchas aplicaciones de minería de datos, tales como la comercialización, el fraude, la intrusión en redes y sistemas informáticos, el correo *spam* y el diagnóstico médico entre otras. La línea de trabajo sobre este tema es conocida como aprendizaje sensible al coste (*cost-sensitive learning*), en las áreas del aprendizaje computacional y la minería de datos.

Así, en clasificación es a veces conveniente evaluar no sólo la precisión del modelo, sino también el contexto donde éste se va a utilizar, por ejemplo, teniendo en cuenta que puede haber clases minoritarias, o que incluso el coste de los errores en la clasificación puede ser diferente según la clase.

Por lo tanto, el aprendizaje sensible al coste puede considerarse como una generalización más realista del aprendizaje predictivo. En este contexto, la calidad de un determinado modelo se mide en términos de minimización de costes, en vez de en minimización de errores.

Veamos un ejemplo; en el servicio de urgencias de un hospital, se desea mejorar su sistema de admisión usando técnicas de aprendizaje inductivo. Cuando un paciente acude a urgencias, una vez evaluado y realizadas las primeras atenciones, se pueden dar tres posibles situaciones que dependen de su estado: ser dado de alta, ser hospitalizado en una planta o ser ingresado en la UCI (Unidad de Cuidados Intensivos). El porcentaje de casos de cada una de estas tres clases es del 86,5%, 13% y 0,5% respectivamente. Usando un algoritmo de minería de datos sobre los datos referentes a antiguos pacientes podríamos obtener un modelo que siempre diera de alta a todos los pacientes. En términos de precisión, éste sería un buen modelo ya que su precisión sería del 86,5%. Sin embargo, este modelo es inútil. Más importante todavía, puede ser peligroso ya que pacientes cuyas condiciones de salud requerirían un ingreso en la UCI serían enviados a sus casas, lo cual es una política desastrosa para el hospital.

El ejemplo anterior pone de manifiesto que necesitamos conocer mejor el tipo de errores y su coste asociado. En los problemas de clasificación se usa para ello una matriz de confusión. A continuación, mostramos una matriz de confusión para un modelo dado en el ejemplo de las urgencias de un hospital. Como podemos observar el modelo predijo para la clase Planta correctamente 36 de los 47 casos actuales, e incorrectamente 11 casos: 9 como Alta y 2 como UCI.

		Real		
		Alta	Planta	UCI
Predicho	Alta	40	9	3
	Planta	10	36	1
	UCI	1	2	15

Si se dispone de información sobre el coste de cada error/acierto en la clasificación, entonces las celdas de la matriz pueden asociarse con el coste de cometer un cierto error de clasificación o de efectuar una clasificación correcta. En este caso, tenemos una matriz de costes de clasificación, como la

que se muestra a continuación siguiendo con el ejemplo de las urgencias del hospital.

		Real		
		Alta	Planta	UCI
Predicho	Alta	0	5,000	500,000
	Planta	300	0	50,000
	UCI	800	500	0

Figura 2.5. Matriz de costes.

Los costes individuales se obtienen multiplicando la frecuencia de errores por su coste (obsérvese que no existe coste para los aciertos). Finalmente, para obtener el coste global se suman todos los costes individuales. En este ejemplo el coste global es 1,599,800. De esta misma forma podemos evaluar varios modelos y seleccionar el modelo que minimice el coste global.

Por desgracia, no siempre se dispone de una matriz de coste que permita adaptar el aprendizaje a ese determinado contexto de coste. Muchas veces, la matriz de coste sólo se conoce durante el tiempo de aplicación y no durante el aprendizaje, generalmente porque los costes varían frecuentemente o son dependientes del contexto. En esta situación, lo que se suele hacer es aprender un conjunto de clasificadores y seleccionar los que mejor se comporten para unas circunstancias o contextos de coste determinados a posteriori. Para ello, la técnica denominada análisis ROC (*Receiver Operating Characteristic*) (Egan, 1975; Swets et al., 2000; Fawcett, 2003) provee herramientas que permiten seleccionar el subconjunto de clasificadores que tienen un comportamiento óptimo para todas las posibilidades de costes.

El análisis ROC se utiliza normalmente para problemas de dos clases (que se suelen denominar clase positiva y clase negativa).

Lógicamente, a partir de los ratios *TPR*, *FNR*, *FPR* y *TNR* cuya definición se vió en la Tabla 2.2, se observa que $TPR = 1 - FNR$, y $FPR = 1 - TNR$. Ambas equivalencias permiten que se puedan seleccionar sólo dos de los porcentajes (*TPR* y *FPR*) y construir un espacio bidimensional con ellos (llamado espacio ROC), limitado por los puntos (0,0) y (1,1). El análisis se basa en representar cada clasificador como un punto en el espacio ROC. En concreto, en el análisis ROC clásico se utiliza el valor *TPR* para el eje de las y, mientras que el eje de las x corresponde al *FPR*.

Por ejemplo, considere un clasificador con la siguiente matriz de confusión:

	Real	
Estimado	30	30
	20	70

A partir de la matriz de confusión, calculamos directamente los valores de los porcentajes *TPR* y *FPR*, en este caso *TPR*=0.6 y *FPR*=0.3. Esto mismo también se puede obtener si representamos la matriz de confusión normalizada por columnas

	Real	
Estimado	0.6	0.3
	0.4	0.7

La siguiente figura muestra este clasificador en el espacio ROC:

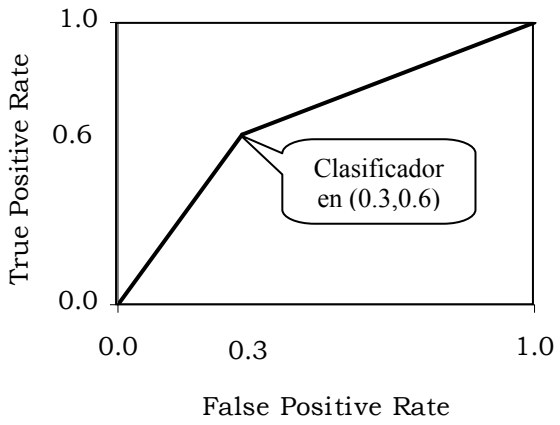


Figura 2.6. Clasificador en el espacio ROC.

Los puntos (0,0) y (1,1) corresponden, respectivamente, al clasificador que predice todo como clase positiva, y al clasificador que asigna todo como clase negativa. Estos clasificadores se conocen como los clasificadores triviales.

Representar un clasificador de esta forma en el espacio ROC tiene interesantes propiedades (Provost y Fawcett, 1997). Lo más importante es que dado un conjunto de clasificadores, se pueden descartar los clasificadores que no caigan dentro de la

superficie convexa formada por todos los clasificadores, junto con los puntos (1,1), (1,0) y (0,0).

Un diagrama ROC típico puede verse en la Figura 2.7. En esta figura se muestran los puntos ROC correspondientes a cinco clasificadores (A, B, C, D y E). Los clasificadores B y E se sitúan en el interior de la superficie convexa por lo que sabemos que siempre habrá algún otro clasificador que tenga mejor comportamiento que estos dos para cualquier contexto de coste.

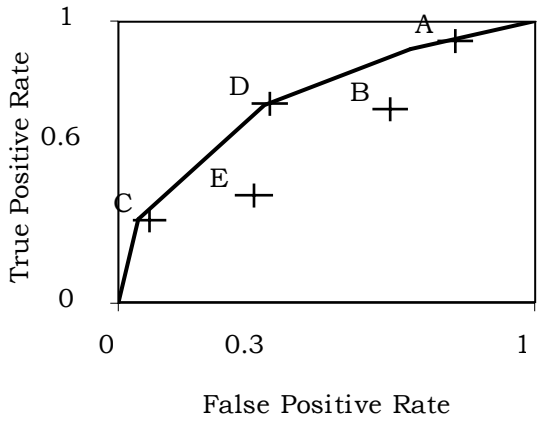
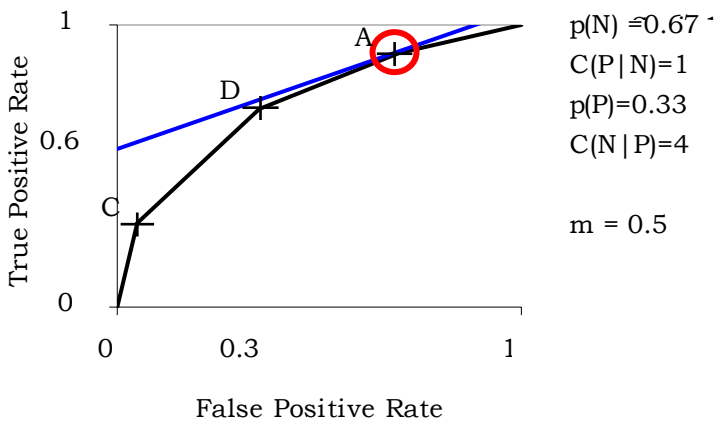


Figura 2.7 Ejemplo de diagrama ROC.

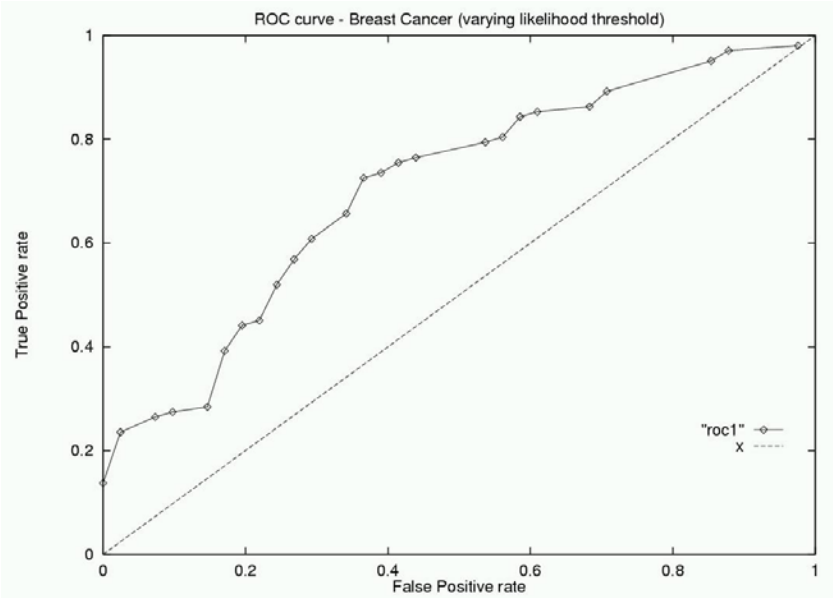
De un diagrama ROC se obtiene una métrica que permite evaluar la calidad de un clasificador: el área bajo la curva de ROC (AUC por *Area Under the ROC Curve*). El AUC evalúa cuán bien un clasificador realiza un *ranking* de sus predicciones, o, dicho de otro modo, evalúa cuán bien un clasificador es capaz de ordenar sus predicciones según la fiabilidad que otroga a las mismas. Una área de 1 representa una calidad perfecta; un área del 0.5 representa una calidad inservible.

Siguiendo con la Figura 2.7, del subconjunto de clasificadores (A, C y D) utilizaremos el modelo con mayor AUC para empezar a aplicarlo con casos reales; si durante la aplicación del modelo podemos estimar la matriz de costes, cambiaremos el modelo a utilizar por el modelo que obtenga mejores prestaciones para la matriz de coste estimada de entre el subconjunto de modelos óptimos. Si conocemos la matriz de costes entonces podemos calcular una pendiente (*skew*) *m* asociada al contexto definido

por dicha matriz. La pendiente se obtiene como m = p(N) · C(P|N) /p(P) · C(N|P), donde C(P|N) es el coste asociado a clasificar un objeto como P cuando realmente es N, C(N|P) es el coste asociado a clasificar un objeto como N cuando realmente es P, p(P) es la probabilidad de que un nuevo dato sea de clase P, p(N) es la probabilidad de que un nuevo dato sea de clase N (p(N) + p(P) = 1). Una vez calculada la pendiente, para seleccionar el clasificador óptimo, dibujamos una línea imaginaria de pendiente m que pase por el punto (0, 1), y deslizamos esa línea (sin perder la pendiente) hacia la curva ROC. El primer punto de la curva con el cual intersecta la línea nos indicará el clasificador con mejores prestaciones para el contexto asociado a la pendiente de la línea imaginaria. Por ejemplo, considerando la matriz de confusión anterior tendremos p(N)=0.67 y p(P)=0.33. Suponiendo los siguientes costes: C(P|N)=1 y C(N|P)=4, se obtiene una pendiente m = (0.67 · 1)/(0.33 · 4) = 0.5. Finalmente, de acuerdo a esta pendiente el clasificador seleccionado es A:



Un clasificador "*soft*" se puede convertir en un clasificador "*crisp*" utilizando un umbral. Por ejemplo: "si score > 0.7 entonces clase A, si no clase B". Con distintos umbrales, tenemos distintos clasificadores, que les dan más o menos importancia a cada una de las clases (sin necesidad de sobremuestreo o submuestreo). Podemos considerar cada umbral como un clasificador diferente y dibujarlos en el espacio ROC. Esto genera una curva como la siguiente



Para realizar la selección del mejor clasificador probabilístico utilizando la curva ROC, se dibujan las curvas de cada uno de los clasificadores. Luego se obtienen sus áreas bajo la curva ROC, entonces el de mayor área es el mejor.

La limitación principal del análisis ROC es que sólo es fácilmente aplicable a problemas de dos clases. Para un problema con c clases, el espacio ROC tiene $c \cdot (c-1)$ dimensiones. Este hecho hace prácticamente inviable el cómputo de superficies convexas para $c>2$, por lo que se han estudiado varias aproximaciones para calcular el AUC de clasificadores en problemas multiclase (Hand y Till, 2001; Ferri et al., 2003). Otra alternativa es la presentada en (Srinivasan, 1999) que consiste en reducir un espacio c-dimensional en bidimensional considerando una clase contra todas las demás.

2.4.3. Revisión de modelos

Los modelos obtenidos en la fase de minería de datos, convenientemente validados e integrados en el uso de la organización no duran siempre. Las condiciones del entorno del problema pueden alterarse, haciendo que un modelo que en un principio

era muy útil, deje de ser válido para su aplicación debido a que no se ha adaptado a la nueva situación.

Por ejemplo, podríamos aprender un modelo para la predicción del abandono de los alumnos de primer curso de una facultad. Este modelo podríamos aplicarlo durante varios años para estimar la matrícula del segundo curso. Obviamente, aunque el comportamiento de los alumnos tendrá algún patrón común durante estos años, existen otros factores que pueden afectar al comportamiento (cambio de plan de estudios en la enseñanza secundaria, incremento en la nota mínima de acceso a la facultad…), y que supondrían una pérdida de precisión del modelo, en caso de que no se efectúe una actualización (revisión) del mismo.

La revisión de modelos se define como el proceso de modificar un modelo inicial para producir un nuevo modelo que sea más preciso en una nueva situación. En general, todos los métodos de revisión de modelos tienen como objetivo revisar el modelo y no construir un nuevo modelo (cambiando alguna de sus partes).

En los sistemas de revisión de modelos estándar: RTLS (Ginsberg, 1990); DUCTOR (Cain, 1991); EITHER (Ourston y Mooney, 1994) y KBDistAl (Yang et al., 1999) se emplean los casos que son incorrectamente predichos por la teoría para realizar algunas modificaciones (como por ejemplo, eliminar/añadir algunas reglas o eliminar/añadir condiciones) que actualicen la teoría.

Otra aproximación diferente es la utilización de algoritmos incrementales. Un algoritmo incremental es capaz de aprender modelos utilizando tan sólo una parte de la información disponible y, posteriormente, utilizar el resto para actualizar o revisar los modelos aprendidos. Si el algoritmo utiliza los ejemplos gradualmente, se denomina incrementalidad vertical. En el caso de que el algoritmo emplee todos los ejemplos, pero la incrementalidad se realice con respecto al número de atributos se llama incrementalidad horizontal.

Usualmente, los métodos de revisión de modelos, especialmente los basados en aprendizaje incremental, intentan determinar los cambios mínimos a la teoría que corrigen la predicción. En algunos casos, se buscan exactamente los cambios mínimos, como por ejemplo (Utgoff, 1994). Este comportamiento tiene

generalmente una consecuencia negativa después de un cierto tiempo: las teorías llegan a ser algo desiguales y consiguen a veces sobreajustarse a las nuevas instancias. Este es especialmente el caso cuando los cambios son muy significativos, como sucede en "*concept drift*" (Widmer y Kubat; 1996).

Capítulo 3

Extracción de modelos comprensibles

Como ya mencionamos en la introducción, la minería de datos tiene como objetivo identificar patrones válidos, novedosos, potencialmente útiles y, finalmente, comprensibles a partir de un gran volúmen de datos. Esta última propiedad, la comprensibilidad del modelo, es necesaria ya que el propósito de la minería de datos es asistir a la toma de decisiones. Si las personas encargadas de tomar las decisiones no pueden entender cómo se obtiene el resultado al aplicar un determinado modelo, entonces éste no puede ser utilizado adecuadamente y se pierde la credibilidad sobre su resultado. A esto se une el hecho que muchos de los usuarios no son expertos en minería de datos, por lo que, los modelos deben estar expresados en un estilo comprensible por ellos.

Es evidente que los algoritmos de minería de datos que generan modelos con buena comprensibilidad son muy deseables. Desafortunadamente, la mayoría de los algoritmos de minería de datos no generan modelos comprensibles y por lo tanto su comprensibilidad tiene que ser alcanzada por otros mecanismos adicionales. Puesto que hay muchas tareas y algoritmos de minería de datos, es difícil que un documento como la presente tesis los cubra todos. Así pues, las discusiones siguientes se restringen a la comprensibilidad de algunos algoritmos de

clasificación, pero en esencia son también aplicable, a otras clases de algoritmos de minería de datos.

Algunos algoritmos de clasificación se califican como comprensibles porque los patrones obtenidos se expresan en una manera explícita. Un caso representativo de estos algoritmos son los árboles de decisión que codifican los patrones bajo la forma de reglas condicionales que pueden examinarse fácilmente. Otros algoritmos de clasificación se califican como incomprensibles porque los patrones se expresan en una forma implícita. Un caso representativo de este tipo son las redes neuronales artificiales que codifican los patrones con pesos de conexión entre ciertos nodos organizados en cierta topología. Actualmente, como veremos, muchos métodos se han desarrollado para mejorar la comprensibilidad de aquellos algoritmos que generan modelos incomprensibles de clasificación, especialmente para las redes neuronales artificiales.

El supuesto de que la comprensibilidad es producida por que un algoritmo que genera explícitamente patrones, no siempre es válida. Por ejemplo, para un árbol de decisión que contiene centenares de hojas, ¿es o no comprensible? Una respuesta cuantitativa a esta pregunta puede ser más razonable que una cualitativa. Así, una medida cuantitativa de la comprensibilidad es necesaria. Tal medida puede también ayudar a solucionar el clásico problema cómo comparar la comprensibilidad de diversos algoritmos.

En definitiva, si disponemos de una medida de comprensibilida, podremos evaluar técnicas de aprendizaje que generen modelos explícitos y también podremos evaluar las técnicas que convierten modelos implícitos en modelos explícitos.

En este último sentido, la meta de convertir cualquier modelo de minería de datos en un conjunto de reglas puede parecer una quimera. En general, hay dos formas factibles de alcanzarla:

- Usar diferentes traductores específicos para convertir un tipo específico de modelo en un conjunto de reglas.

- Usar un traductor general para convertir cualquier modelo (sean redes neuronales, máquinas de vecto-

res soporte, vecinos más cercanos, ...) en un conjunto de reglas.

Ha habido muchas técnicas desarrolladas para la primera aproximación, sobre todo para convertir redes neuronales en reglas. El esquema principal para mejorar la comprensibilidad de las redes neuronales artificiales es la extracción de reglas simbólicas (*rule extraction*) a partir de la red. Esta aproximación tiene sus orígenes en el trabajo de Gallant sobre el sistema experto conexionista (Gallant, 1983; Andrews et al., 1995; Tickle et al., 1998).

La segunda aproximación, aunque sería más aplicable generalmente, no se ha analizado en el mismo grado, probablemente porque no está muy clara la manera en la cual un conjunto de reglas se puede extraer de cualquier clase de modelo, independientemente de su representación.

En concreto, nos interesan aquellas técnicas que consideran los modelos originales como "cajas negras", es decir, que mapean las entradas en las salidas sin tomar en cuenta el detalle interno del modelo. Esto nos permite considerar no solamente redes neuronales sino otros clasificadores, como los obtenidos mediante técnicas de combinación de modelos tales como *bagging*, *boosting*, *stacking*, etc..

En este capítulo describimos los trabajos relacionados con laconversión de un modelo de caja negra a una versión comprensible. En la sección 1 presentamos la extracción de reglas a partir de cajas negras, que corresponde con la primera aproximación anteriormente descrita. La sección 2 presenta la segunda aproximación denominada técnica mimética. Debido a que nuestra tesis se centra en el estudio y extensión del método mimético, realizamos una descripción histórica del desarrollo del mismo. Un tema importante en el esquema de aprendizaje mimético es cómo se generan los datos inventados; por ello, dedicamos la sección 3 a describir los métodos más importantes para la generación de datos inventados. En la sección 4 mostrando las ventajas y limitaciones de la técnica mimética, las mismas que han inspirado la realización de esta tesis.

3.1. Extracción de reglas a partir de modelos de caja negra

La extracción de reglas se define como la obtención de la representación simbólica en forma de reglas que describe el comportamiento (patrón) general de un modelo. Existen numerosas técnicas que extraen reglas de modelos específicos, por ejemplo, redes neuronales, mediante el análisis de la estructura, pesos o características del modelo. Por el contra, existen técnicas que extraen reglas de tipos de modelos específicos, pero no necesitan analizar la estructura interna del modelo, tomándolo así como una "caja negra". Las técnicas de extracción de reglas a partir de modelos de caja negra se denominan técnicas pedagógicas (Andrews et al., 1995).

En esta sección revisamos algunas propuestas o trabajos de extracción de reglas pedagógicas por ser los más relacionados con la técnica mimética.

Muchos métodos del tipo pedagógico se han desarrollado para primero aprender una red neuronal, con la suposición de que producirá un modelo preciso, y después extraer un modelo comprensible de ella, en la forma de un sistema de reglas o un árbol de decisión: *TREPAN* (Craven, 1996); (Andrews et al., 1995); (Saito y Nakano, 1988, 1990); *Rule Extraction As Learning* (Craven y Shavlik, 1994); *Relevance Information* (Johnson et al., 1996); *RULENEG* (Pop et al., 1994); *Binarised Input-Output Rule Extraction* (Taha y Ghosh, 1996); *Decision Tree Extractor* (Boz, 2000).

También se han desarrollado aproximaciones para extraer modelos simples a partir de la combinación de árboles de decisión: usando *arcing* (Breiman y Shang, 1996), usando *bagging* (Domingos, 1997a), usando *boosting* (Estruch et al, 2003).

Otras aproximaciones extraen reglas a partir de la combinación de redes neuronales artificiales (Zhou et al., 2003; Zhou y Jiang, 2003, 2004). En la combinación se utiliza la votación.

Todos estos trabajos sugieren el paradigma de aprender dos veces para desarrollar clasificadores precisos y comprensibles, es decir, una primera etapa de clasificación que tiene como objetivo conseguir una alta precisión, y una segunda etapa en la que se persigue la comprensibilidad.

La mayoría de los algoritmos actuales de extracción de reglas sufren de alta complejidad de cómputo y suelen especializarse para cada técnica. Por lo que se hace necesario contar con algoritmos más simples y generales. Uno de estos es el método mimético.

3.2. El método mimético

Los orígenes del método mimético se remontan al nacimiento del método denominado combinación de múltiples modelos (CMM) (Domingos, 1997a, 1997b, 1998). La principal idea del método CMM es recuperar la pérdida de comprensibilidad por la combinación de múltiples modelos. Para realizar esta recuperación (véase el algoritmo en seudocódigo del método CMM en la Figura 3.1) se reaplica un clasificador base (*L*) para recuperar las fronteras implícitas en la combinación de múltiples modelos. Esto se realiza dando a *L* un nuevo conjunto de datos de entrenamiento, integrado por una gran cantidad de ejemplos (*R*) generados y clasificados según la combinación, más los ejemplos originales. Por otro lado, el modelo aprendido con el clasificador base (*M*) es menos preciso que el modelo combinado ($C_{M1};\ldots;_{Mm}$), especialmente si el poder de representación de la combinación de modelos es más grande que la del clasificador base. Sin embargo, y de forma general, se sabe que la precisión y estabilidad de un modelo se incrementa con el tamaño del conjunto de entrenamiento (debido a la disminución de la varianza (Kohavi y Wolpert, 1996). En el caso de la generación del modelo *M*, el tamaño del conjunto de entrenamiento puede ser tan grande como se quiera, ya que en parte es inventado. Esto hace posible que el modelo sea más preciso y estable que los modelos base de la combinación.

Según Domingos dos puntos significativos se deben tomar en cuenta en el método CMM. El primero es que los ejemplos generados en la fase de meta-aprendizaje deben ser generados de la misma distribución original ("verdadera") y no según la distribución uniforme.

El segundo punto que debe ser notado es que, en CMM, el conjunto de entrenamiento del meta-aprendizaje está compuesto de los ejemplos generados artificialmente y de los originales.

```
Entradas:
    T es el conjunto de entrenamiento,
    L es un algoritmo de aprendizaje,
    C es un procedimiento para combinar modelos,
    m es el número de modelos componentes a generar,
    n ese el número de nuevos ejemplos a generar.

PROCEDIMIENTO CMM (T; L; C; m; n)
    PARA i = 1 A m HACER
        Ti ← variación de T
        Mi ← Modelo producido por aplicar L a Ti
    FIN-PARA
    R ← {}
    PARA j = 1 A n HACER
        x ← generar aleatoriamente un ejemplar
        c ← clase asignada a x por C_{M1;...;Mm} (x)
        R ← R ∪ {(x; c)}
    FIN-PARA

    T ← T ∪ R
    M ← Modelo producido por aplicar L a T

    REGRESAR M
FIN-PROCEDIMIENTO
```

Figura 3.1 El algoritmo de meta-aprendizaje CMM

El trabajo experimental de Domingos se realizó sobre 26 conjuntos del repositorio UCI. El modelo combinado se obtuvo aplicando bagging, mientras que como algoritmo de meta-aprendizaje (clasificador base) se usó C4.5Rules. Los resultados obtenidos demostraron que el segundo modelo tenía una precisión entorno al 60% de la precisión del modelo combinado mientras que su complejidad en términos del número de reglas del modelo era entre 2 y 6 veces mayor que un modelo simple generado usando C4.5Rules a partir de únicamente los datos originales.

Posteriormente en (Ferri et al., 2002) se aborda este mismo problema desde una perspectiva diferente. La solución propuesta se basa en la idea de arquetipo, donde el arquetipo es uno de los modelos simples constituyentes de la combinación, justo aquel que más se aproxima de acuerdo a una medida de similitud) al modelo combinado. Por lo tanto, este modelo simple puede considerarse un representante de la combinación, de

ahí su nombre. También se demostró que la solución obtenida no es 100% equivalente a la combinación, pero en general es razonablemente cercana.

Más recientemente en (Estruch et al, 2003) se analizó experimentalmente el método mimético pero con un modelo combinado diferente al usado en la aproximación CMM. En concreto se aplicó *boosting*, obteniéndose unos resultados consistentes con los de Domingos. En este artículo se presentó la idea de un esquema más general que el método CMM, que denominaron la técnica mimética. Básicamente, la técnica mimética consiste en aprender un primer clasificador muy preciso pero incomprensible, denominado oráculo. Entonces, se usa para etiquetar un conjunto de datos inventados, que resultan ser los datos de entrenamiento de un segundo clasificador. Más concretamente, como puede observarse en la Figura 3.2, la técnica mimética consta de dos etapas.

La primera etapa corresponde al aprendizaje del oráculo. El objetivo de esta etapa es obtener un modelo de una alta precisión aunque sea incomprensible (caja negra). Para ello, se utilizan los datos de entrenamiento originales *T* como entrada al primer algoritmo de aprendizaje, el cual creará este primer clasificador. Ejemplos de algoritmos que se pueden utilizar en esta primer etapa son: las redes neuronales, multiclasificadores (*boosting*, *bagging* o cualquier otra combinación de clasificadores) y máquinas de vectores soporte. Éste es el modelo de referencia en base al cual se realiza el siguiente paso.

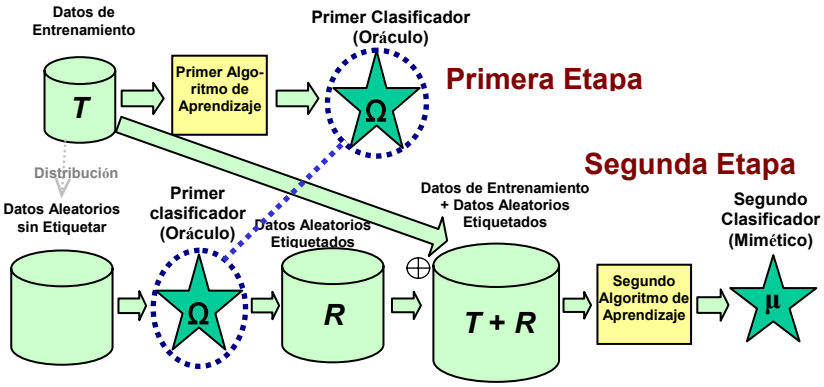


Figura 3.2 La Técnica Mimética

En la segunda etapa (véase la parte inferior de la Figura 3.2) se realiza el aprendizaje mimético. El objetivo de esta etapa es

obtener un modelo que imite el comportamiento del oráculo, buscando conservar su nivel de precisión generando al mismo tiempo un conjunto de reglas comprensibles que permitan el entendimiento de dicho modelo. Tomando como base los datos de entrenamiento originales (por ejemplo para extraer la función de distribución de los mismos) y empleando algún método simple de muestreo se genera un conjunto de datos aleatorio sin etiquetar (R). Luego ese conjunto de datos R se etiqueta utilizando el oráculo. El conjunto de datos etiquetado por el oráculo y el conjunto de datos de entrenamiento original se unen para formar el total de los datos de entrada ($T + R$) para el segundo algoritmo de aprendizaje el cual crea el modelo mimético. El segundo algoritmo de aprendizaje puede ser cualquier clasificador que genere reglas, como, por ejemplo, J48 (la implementación en Weka (Witten y Frank, 2005) del conocido algoritmo de inducción de árboles de decisión C4.5 (Quinlan, 1993)) o See5/C5 (el sucesor del algoritmo C4.5), CN2 u otros algoritmos de aprendizaje de reglas.

Los resultados experimentales de (Estruch et al, 2003) muestran que el uso del conjunto de datos de entrenamiento original (unido con el conjunto de ejemplares aleatorios) es beneficioso desde el punto de vista de la precisión del modelo mimético. También, mostraron que el uso de la distribución a priori para generar el conjunto de datos aleatorios es preferible sobre la distribución normal.

3.3. Generación de los datos inventados

Como hemos descrito en el apartado anterior, el modelo mimético requiere de un conjunto de datos inventados y etiquetados por el oráculo.

El aprendizaje a partir de datos sin etiquetar no se formaliza teóricamente frecuentemente. Sin embargo, existen aplicaciones prácticas que toman ventaja de los datos sin etiquetar (Zhang y Oles, 2000). La idea de utilizar datos sin etiquetar es antigua y se empleó en estadística. En 1968, Hartley y Rao sugieren la combinación de datos etiquetados y sin etiquetar para construir clasificadores con aumento en su precisión siempre y cuando los datos inventados lo sean con la distribución a priori.

Existen diversas formas de generar estos datos inventados siendo las siguientes las más habituales:

Distribución uniforme: Consiste en recopilar para cada atributo nominal el conjunto de todos los posibles valores que puede tomar, mientras que de los atributos continuos se recopilan los valores máximo y mínimo {*min*, *max*} que se conocen del dominio del problema. De esta forma para generar un valor de un atributo nominal se elige aleatoriamente un valor del conjunto de posibles valores usando la misma probabilidad para todos ellos. Para los atributos numéricos, se genera aleatorriamente un valor en el intervalo {*min*, *max*} de valores máximo y mínimo para este atributo. Esta es la distribución más simple ya que requiere disponer de poco conocimiento. Por ello, es una opción generalmente utilizada en caso de desconocer la distribución original.

Distribución a priori: en este caso el objetivo es generar un conjunto inventado sonde la probabilidad de que un ejemplo aparezca en él sea similar a la probabilidad (o frecuencia) de aparición en un conjunto real. Una manera sencilla de obtener esta distribución es generar el valor de cada atributo X_i de un nuevo ejemplo a partir de uno de los valores v_i presente en un ejemplo diferente $R(v_1,\ldots,v_i,\ldots,v_a)$ dentro del conjunto de datos de entrenamiento. Por lo tanto, para esta distribución es necesario disoner del conjunto de datos iniciales a partir de los cuales se inventan otros.

Distribución normal Un método más sofisticado es la distribución normal (Melville y Money, 2005). En ella, los valores de los atributos nominales se seleccionan tomando un valor del conjunto de posibles valores basándose en la frecuencia de ocurrencia de los valores en los datos de entrenamiento (empleando probabilidades acumuladas). En el caso de atributos continuos, el valor numérico se obtiene de una distribución gaussiana definida por la media y la desviación estándar del atributo en cuestión. Ésta es una distribución con exigencias intermedias entre la distribución uniforme y la a priori, ya que se requieren dos valores (la media y la desviación) que describan la distribución original.

Distribución a priori por interpolación: Es similar a la distribución a priori para los atributos nominales, pero para atributos numéricos, miramos un valor $v_{i,1}$ en el conjunto de entrenamiento, pero entonces buscamos los valores más cercanos

$v_{i,2}$ y $v_{i,3}$ de otros ejemplos tales que $v_{i,2} < v_{i,1}$ y $v_{i,3} > v_{i,1}$. Entonces utilizamos una distribución uniforme de $v_{i,2}$ a $v_{i,3}$ para generar un nuevo valor v' que es utilizado como el valor numérico para el argumento i del ejemplo inventado. Si el valor $v_{i,2}$ no existe ($v_{i,1}$ no tiene ningún valor a la izquierda), entonces utilizamos la distribución normal de un solo lado con media en $v_{i,1}$ y un valor de la desviación estándar de la mitad del rango (min_i, max_i) del atributo. Para el valor derecho en la situación contraria, nos comportamos de forma similar. Esta distribución trata de combinar lo mejor de la distribución a priori y la distribución normal.

Para problemas binarios (sólo dos clases), se pueden utilizar otros métodos si consideramos generado un conjunto grande de ejemplos de talla n con cualquiera de los métodos anteriores. En realidad se trata de métodos de muestreo sobre las técnicas anteriores. Para ello, ordenamos los ejemplos de acuerdo a la probabilidad de estimada por el oráculo de que dicho ejemplo sea de una cierta clase (clase positiva). El orden es asignado desde 1 (mejor valor del *ranking*) hasta n (peor valor del *ranking*). Y finalmente aplicamos alguno de los siguientes métodos:

Muestreo por rango absoluto: Se seleccionan los r ejemplos mejores en el *ranking*. El problema de este método es que no es exhaustivo (algunas zonas en el espacio puede que nunca sean elegidas porque el oráculo no se comporta bien allí). Por otra parte, si el tamaño de los datos que disponemos (n) es mucho mayor que el tamaño requerido (r) usados por el método mimético entonces esto es más probable que suceda. Por lo tanto, en este método la relación entre n y r es muy importante.

Muestreo por rango probabilístico: Seleccionar probabilísticamente los r ejemplos asignando a cada ejemplo una probabilidad $Prob(e)=2\cdot(n-i+1)/n\cdot(n+1)$ donde i es el orden del ejemplo e. Este método es exhaustivo (cualquier ejemplo podría salir), puede generar ejemplos repetidos y los resultados serán más o menos los mismos independientemente de la relación entre n y r. Obviamente hay otras formas alternativas de obtener la probabilidad de un ejemplo, pero ésta es una aproximación inicial y fácil.

Muestreo por umbral: Para un ejemplo, sí su probabilidad es mayor o igual que un valor de umbral entonces se acepta el ejemplo en caso contrario se descarta. Se generan ejemplos de

esta forma hasta que se obtengan los r ejemplos. Este método no es exhaustivo.

En esta tesis aplicaremos las distribuciones a priori, uniforme y *bootstrap* por umbral.

3.4. Ventajas y desventajas del método mimético

El método mimético trabaja particularmente bien cuando el "oráculo" es una combinación de clasificadores. Es, por tanto, el primer método general para convertir métodos multiclasificadores en modelos simples. Sin embargo, no se ha consolidado como una técnica general para la extracción de reglas comprensibles a partir de cualquier oráculo. Es decir, hasta el momento, sólo se ha aplicado con oráculos que son combinación de modelos múltiples (*bagging* y *boosting*) y no con otros modelos que tengan muy buena precisión pero de baja comprensibilidad como es el caso de las redes neuronales, los clasificadores naive bayes o incluso cuando el oráculo es un experto humano.

El método mimético puede implementarse fácilmente en cualquier paquete de minería de datos. Sin embargo, todavía no existen implementaciones incorporadas en paquetes o librerías de dominio público, como por ejemplo, Weka.

Una desventaja en el método mimético es que al utilizar poda excesiva en la segunda fase para lograr modelos simples se anula gran parte de la mejora en precisión. Este es un indicio de que faltan muchos factores que estudiar, como por ejemplo, el nivel de poda, el nivel de confianza, el tamaño de los datos inventados, etc. Además de la influencia combinada de los mismos.

Otra desventaja es que esta técnica contrapone las dos propiedades más importantes de los modelos, la precisión y la comprensibilidad. Esto se debe a que al aumentar el número de datos inventados se generan modelos con mejor precisión y, consecuentemente, con mayor número de reglas (menor comprensibilidad) y viceversa.

Los marcos de trabajo para evaluar la calidad de las reglas extraídas son importantes. Como se puede observar Domingos

utiliza un marco con tres parámetros principales para definir la calidad del conocimiento aprendido: precisión, estabilidad y comprensibilidad. En (Ferri et al., 2003) solamente dos precisión y comprensibilidad.

Nosotros estamos interesados en continuar los estudios del método mimético y tenemos el objetivo de lograr buenos compromisos entre comprensibilidad y precisión. Por ello, en los capítulos 5 y 6 aportamos dos marcos para encontrar un compromiso entre comprensibilidad y precisión.

Capítulo 4
Estudio teórico

Como ya hemos dicho en el capítulo anterior, la mimetización es un proceso de aprendizaje que consiste en crear un nuevo clasificador que "mimetice" o imite el comportamiento de otro clasificador preciso pero incomprensible (una red neuronal, una combinación de clasificadores, etc).

En (Estruch y Hernández, 2003) se abordó por primera vez un estudio teórico de la técnica mimética en función de sus características y parámetros, que este capítulo resume y completa.

Por ejemplo, relacionada con esta técnica de aprendizaje surgen muchas preguntas: ¿Cuáles son los parámetros que caracterizan a los modelos miméticos?, ¿Cómo es la relación entre estos parámetros?, ¿Cuán grande debe ser el conjunto de datos inventados para lograr un equilibrio entre precisión y comprensibilidad?, ¿Cómo de similar al oráculo y preciso puede ser el modelo mimético?, ¿Pueden ser los modelos miméticos comprensibles?.

Para poder dar respuesta a estas preguntas analizaremos los límites teóricos y caracterizaremos el comportamiento de la técnica mimética. Tomaremos como base para el estudio las curvas de aprendizaje para diferentes configuraciones del método mimético. Como vimos en el capítulo 2, una curva de aprendizaje es una representación gráfica de la relación de variación que presentan dos variables involucradas en el proceso de aprendizaje. Estas variables pueden ser métricas de validación, parámetros de configuración o valores de entrada del algoritmo de aprendizaje, como por ejemplo, la precisión y el tamaño del conjunto de entrenamiento.

Otro objetivo no menos importante de este capítulo es acompañar con resultados teóricos todos los experimentos realizados en los dos siguientes capítulos y así poder realizar un análisis razonado de estos resultados experimentales. Vamos a considerar en este capítulo la situación mostrada en el capítulo anterior en lo que se dispone no sólo del oráculo sino también de los datos de entrenamiento originales. Sin embargo, hay casos donde el método mimético es útil porque tenemos una caja negra y no tenemos conjunto de entrenamiento. Este caso lo analizamos teóricamente en primer lugar de una forma más breve y lo trataremos experimentalmente en el capítulo 6 de una manera más extensa.

Para abordar estos temas requerimos, primero que nada, un acuerdo en la notación y configuración, presente en la sección 1. Luego, en la sección 2 describimos los resultados generales encontrados, para después, en la sección 3, centrarnos en los árboles de decisión. La sección 4 trata las curvas de aprendizaje según el tamaño del conjunto de datos de entrenamiento, ya que una cuestión importante en el entrenamiento de modelos miméticos es el tamaño del conjunto de datos que se emplea para crear el clasificador mimético. Finalmente, presentamos un resumen de las aportaciones encontradas.

4.1. Notación y configuración

A continuación presentamos algunas definiciones básicas sobre clasificadores. Un dominio de un atributo es cualquier conjunto de valores, ya sean nominales o numéricos. El espacio de evidencia $\mathcal{E}$ es el producto cartesiano $X_1 \times X_2 \times \ldots \times X_m$ donde cada X_i es el dominio[1] del atributo i. El dominio de una clase C es un conjunto finito de c clases denotadas por números naturales: 0, 1, ..., c-1. Un clasificador es una función $f : \mathcal{E} \rightarrow C$. Nos referiremos a la función definida por el clasificador oráculo Ω y por el clasificador mimético μ por f_Ω y f_μ respectivamente. La función objetivo se denota con f_O. Cada elemento $e \in \mathcal{E}$ es un ejemplo sin etiquetar. Un par (e,i) donde $e \in \mathcal{E}$, $i \in C$ es un ejemplo etiquetado. Una muestra (o evidencia o conjunto de datos) E es cualquier subconjunto de $\mathcal{E}$, si la muestra está sin

[1] Un dominio define una dimensión del espacio.

etiquetar, o un conjunto de pares de la forma (e,i) con $e \in \mathcal{E}$, $i \in C$ si la muestra está etiquetada. Usamos la notación E_f para designar el conjunto de datos E etiquetados con la función f. El subíndice puede ser eliminado cuando f esté clara por el contexto. Asumimos que no existen dos ejemplos iguales con diferentes clases. El conjunto de datos de entrenamiento (es decir, el conjunto de datos etiquetados usado en el aprendizaje) lo denotamos como T. El conjunto de datos aleatorios etiquetados con el oráculo se denota por R_Ω (o simplemente R). Un conjunto de prueba o validación es una muestra de un conjunto de datos etiquetados usualmente denotado por V.

Si todos los atributos en $\mathcal{E}$ son numéricos, se puede llamar punto a cada $e \in \mathcal{E}$. Por simplicidad, consideramos (cuando sea necesario) que todos los dominios numéricos X_j están incluidos en el intervalo [0,1]. Dados dos puntos e_1 y e_2, el segmento que conecta a ambos puntos se representa por $\overline{e_1 e_2}$. La longitud del segmento es la distancia euclídea entre e_1 y e_2, denotada por $d(e_1, e_2)$.

Dado el espacio de evidencia $\mathcal{E} = X_1 \times X_2 \times \ldots \times X_m$, denotamos por $\mathcal{E}_\varphi$ al espacio de la subevidencia construido sólo con los atributos nominales, es decir, $\mathcal{E}_\varphi = \Pi X_j$ tal que X_j es nominal. De forma similar, denotamos por $\mathcal{E}_\rho$ el espacio de la subevidencia construido sólo con los atributos numéricos, es decir, $\mathcal{E}_\rho = \Pi X_j$ tal que X_j es numérico. Obviamente, $\mathcal{E}_\varphi$ es finito. Llamamos a $|\mathcal{E}_\varphi|$ la "dimensión nominal" del conjunto de datos. Cada elemento $\alpha \in \mathcal{E}_\varphi$ es llamado un ancla (*anchor*). Si fijamos un ancla entonces la evidencia tiene solamente atributos numéricos variables y podemos ampliar la noción de segmento y de distancia a este caso. Por ejemplo, podemos definir la distancia euclídea entre dos puntos e_1 y e_2, tales que los atributos nominales coincidan como la distancia euclídea considerando solamente los atributos numéricos. Si los atributos nominales de e_1 y de e_2 no son iguales entonces la distancia es infinita. Sin pérdida de generalidad podemos separar los atributos tales que cada ejemplo e puede ser expresado de la manera siguiente $\langle e_\varphi, e_\rho \rangle$ donde $e_\varphi \in \mathcal{E}_\varphi$ y $e_\rho \in \mathcal{E}_\rho$.

En lo que sigue $Prob(v)$ denota la probabilidad de v, y $Prob_w(v)$ la probabilidad de v si w ya sucedió, es decir la probabilidad condicional $Prob(v|w)$.

Definición 1. *Dado un espacio de evidencia $\mathcal{E}$, decimos que un método para generar una muestra aleatoria R es exhaustivo si y sólo si:*

$\mathcal{E}$ tiene solamente atributos nominales y se cumple:

$$\forall e \in \mathcal{E} : Prob(\exists e' \in R, e = e') > 0$$

$\mathcal{E}$ tiene solamente atributos numéricos[2] y se cumple:

$$\forall e_1, e_2 \in \mathcal{E}, e_1 \neq e_2 : Prob(\exists e' \in R, e' \in \overline{e_1 e_2}) > 0$$

$\mathcal{E}$ tiene atributos nominales y numéricos y se cumple:

$$\forall e_\varphi \in \mathcal{E}_\varphi, \forall \tau > 0, \forall e_\rho \in \mathcal{E}_\rho, e_1 = \langle e_\varphi, e_\rho \rangle,$$
$$\forall e_2 \in \mathcal{E}, d(e_1, e_2) < \tau : Prob(\exists e' \in R, e' \in \overline{e_1 e_2}) > 0$$

Las definiciones anteriores tienen como objetivo distinguir entre diferentes métodos de generar datos inventados que cubran o no todo el espacio. De los métodos para generar muestras aleatorias presentados en la sección 3.3 –página 48-, es fácil ver que la distribución uniforme y la distribución de interpolación a priori son métodos exhaustivos. La distribución a priori no es exhaustiva si existen atributos numéricos (porque los valores que no aparecen tienen probabilidad 0).

Para terminar esta sección, definamos la noción de lealtad aplicada a un clasificador.

Definición 2. *Sea C un clasificador, decimos que C es leal si para cualquier ejemplo del conjunto de datos de entrenamiento empleado en el aprendizaje de ese clasificador, C lo clasifica de forma correcta.*

Por ejemplo, un árbol de decisión sin podar es un clasificador leal. Lógicamente un clasificador leal suele realizar sobreajuste en presencia de datos con ruido.

---

[2] La siguiente definición alternativa:

$$\forall \tau > 0, \forall e \in \mathcal{E} : Prob(\exists e' \in R, d(e, e') < \tau$$

no es válida porque el método debe ser exhaustivo en cada dimensión.

4.2. Resultados generales

Dadas las definiciones previas consideramos la situación donde tenemos un espacio de evidencia $\mathcal{E}$, un oráculo Ω y un método exhaustivo para la generación de ejemplos aleatorios de $\mathcal{E}$ etiquetados por Ω. El conjunto de datos inventados lo denotamos por R. A continuación mostramos algunas propiedades generales de la técnica mimética.

La primera propiedad es absolutamente trivial y nos muestra que si todos los atributos son nominales, entonces existe un número de ejemplos finito y por lo tanto, la semántica del oráculo puede ser capturada completamente. Formalmente,

Proposición 1. *Dado un espacio de evidencia* $\mathcal{E} = X_1 \times X_2 \times \ldots \times X_m$ *con solamente atributos* X_i, $1 \leq i \leq m$, *nominales, un oráculo* Ω *y un clasificador mimético leal* μ *aprendido con un conjunto de datos aleatorios R generados por un método exhaustivo y etiquetados con* Ω. *Si* $|R| \to \infty$ *entonces:*

$$\forall e \in \mathcal{E} : f_\mu(e) = f_\Omega(e)$$

Demostración. Conocemos que cada atributo X_i es nominal. Consecuentemente, $\prod_{i=1}^{m} |X_i|$ es el número de posibles ejemplos en $\mathcal{E}$, donde $|X_i|$ indica el número posible de instancias del atributo X_i. Si $|R| \to \infty$, puesto que R se genera por un método exhaustivo, entonces tendremos necesariamente que, como mucho, $R = \mathcal{E}$ (puesto que $\mathcal{E}$ es un conjunto finito[3]), y dado que μ es un clasificador leal que ha empleado a R para aprenderlo, concluimos que $f_\mu(e) = f_\Omega(e)$ para todo $e \in \mathcal{E}$.

□

Este primer resultado ha sido bastante sencillo. Es también bastante obvio que el caso es diferente cuando aparecen números reales en el dominio.

Proposición 2. *Dado un espacio de evidencia* $\mathcal{E}$ *con al menos un atributo numérico* X_i, *y un clasificador mimético leal* μ *aprendido con un conjunto de datos aleatorios R generados por un método exhaustivo. Entonces, existe un oráculo* Ω.*con el que se etiqueta R que satisface que*

---

[3] De hecho $|R|$ nunca puede ser mayor que $|\mathcal{E}|$

$$si \ |R| \to \infty \ \ entonces \ \ \exists e \in \mathcal{E} : f_{\mu}(e) \neq f_{\Omega}(e)$$

Demostración. Considere un oráculo que represente la función $f_{\Omega}(x_1, x_2, x_3,\ldots, x_n)$ = *verdadero* si x_i es racional, si no *falso*. Incluso un conjunto de datos aleatorios extremadamente grande creado con un método exhaustivo de generación nunca generará todos los números racionales (y menos todavía todos los irracionales), por lo que siempre existirá un ejemplo que no se ha visto y para el que el clasificador mimético puede dar una clasificación incorrecta.

□

Esto se relaciona con los teoremas negativos sobre la identificación en el límite (Gold, 1967). De hecho podemos encontrar otros contraejemplos menos pintorescos.

No obstante, parece que cuanto más grande es el conjunto de datos al azar el clasificador mimético podría aproximar mejor al oráculo, incluso cuando hay atributos numéricos. Sin embargo, esto no puede ser demostrado para cualquier función y cualquier clasificador. Necesitamos definir algunas leves restricciones en la función y en el clasificador. Comencemos con la función.

Definición 3. *Dada una función f decimos que un punto e* $\in \mathcal{E}$ *está en una región pura de radio* τ*, denotada por* $Pure_{\tau}$ *(e), si y sólo si*

$$\forall e' \in \mathcal{E} \ (d(e,e') \leq \tau) \to f(e) = f(e')$$

Esto significa que todos los puntos (con una distancia menor que τ) son de la misma clase que *e*. En otras palabras, no existe una frontera más cercana que τ. Es importante observar que τ es un parámetro y tal vez diferente para cada punto.

Definición 4. *Dada una función f, decimos que solamente tiene regiones puras (o simplemente es pura) si y solo si*

$$Prob_{e \in \mathcal{E}}\left[\exists \tau > 0 \wedge Pure_{\tau}(e)\right] = 1$$

No se impone condición adicional para los atributos nominales aparte del hecho de que la condición previa debe ser verdadera para cualquier ancla.

Desde aquí podemos permitirnos un número infinito de puntos en las fronteras, porque el área de estas regiones es cero (τ=0). Sin embargo, no podemos tener un número infinito de puntos "aislados" que son diferentes de sus alrededores, si éstos se reparten alrededor del espacio, incluso si este número infinito es negligible en el espacio entero. Por ejemplo, números naturales entre números reales. La razón es la definición estricta de la región pura.

Para evitar este problema y permitir un conjunto más flexible de funciones, vamos a expresar la definición 3 en términos de probabilidad.

Definición 5. *Dada una función f decimos que un punto e* $\in \mathcal{E}$ *está en una región probabilísticamente pura de radio* τ*, denotado por* $PPure_\tau(e)$*, si y sólo si*

$$Prob_{e' \in \mathcal{E} | d(e,e') \leq \tau} \left[f(e) = f(e') \right] = 1$$

Esto significa que la probabilidad de seleccionar otro punto (con una distancia menor que τ) de la misma clase que *e* es igual a 1. En otras palabras, no existe una frontera más cercana que τ. El término probabilísticamente puro es una generalización e incluye casos donde algún conjunto finito o infinito de puntos "falsos" (sin área) de otra clase pueda estar dentro de la región, pero pueden ser ignorados frente a la mayoría de los puntos que son de la misma clase que *e*. Ahora daremos una definición más flexible (probabilísticamente) de una función pura.

Definición 6. *Decimos que una función f sólo tiene regiones probabilísticas puras (o simplemente es probabilísticamente pura) si y sólo si*

$$Prob_{e \in \mathcal{E}} \left[\exists \tau > 0 \wedge PPure_\tau(e) \right] = 1$$

De nuevo es importante observar que τ puede ser diferente para cada punto. Eso significa que la función es continua (con infinitos puntos en cualquier región) o puede ser discreta pero con una alta resolución (distancia mínima entre puntos) y cualquier forma que podamos desear. Pero, entonces, ¿qué clase de funciones son eliminadas por la definición anterior? La res-

puesta son las funciones para las cuales dada la mayoría de los puntos no podemos encontrar una región pequeña alrededor de él donde la clase sea igual que la del punto. Esta clase de funciones se conoce generalmente como funciones fractales. Por ejemplo, la función siguiente *f* es fractal:

f(x) = *g*(X, 0, 1, verdadero)
g(X, L, R, B) = si (X=(R – L)/2 entonces regresa B
sino si (X< (R – L)/2) entonces *g*(X, L, (R – L)/2, no(B))
sino *g*(X, (R – L)/2, R, no(B))

De hecho, la función *f* previa es tan fractal que su aspecto es "gris" (la mitad de los puntos son de la clase verdadera y la otra mitad son falsos) en cualquier resolución. No obstante, la función anterior está definida solamente para los números racionales. Si consideramos solamente números racionales, entonces la función anterior no es probabilísticamente pura y, lógicamente, tampoco es pura. Sin embargo, si la hacemos completa para los números reales, es decir

$$f'(X) = f(X) \quad si\ X\ es\ racional$$
$$f'(X) = falso\ en\ otro\ caso$$

Ahora *f'* tiene componentes fractales pero, puesto que hay más números irracionales infinitos que racionales, probabilísticamente, tenemos *f'*(X) = falso y por lo tanto, podemos decir que es probabilísticamente pura (aunque no es pura).

En general, la mayoría de los patrones usuales en el aprendizaje computacional y la minería de datos son probabilísticamente puros. Por otra parte, como veremos, con nuestro método mimético, los oráculos implementados en una computadora tienen una resolución limitada, que los hacen probabilísticamente puros o aún simplemente puros. Por lo tanto, las restricciones anteriores se pueden considerar bastante leves.

Una vez dadas estas restricciones en las funciones, pasemos a considerar los algoritmos de aprendizaje. La primera definición es la de algoritmo de aprendizaje "*fence-and-fill*" ("cercar-y-llenar").

Definición 7. *Un algoritmo de aprendizaje "fence-and-fill" es un algoritmo que sólo genera clasificadores cuya función es pura.*

Hay que notar que la definición 7 está definida con la versión no probabilística. Esta no es una restricción importante. Si consideramos que los clasificadores son algoritmos que dan una salida para una entrada en un tiempo finito, entonces son generalmente puros ya que si hay valores numéricos el espacio es infinito y no crean infinitas zonas aisladas. La mayoría de técnicas de aprendizaje: redes neuronales, árboles de decisión, son *fence-and-fill*.

Ahora, dadas las definiciones previas, podemos afirmar un resultado positivo:

Lema 1. *Dado un espacio de evidencia* $\mathcal{E} = X_1 \times X_2 \times \ldots \times X_m$ *tal que* X_i, *es numérico* $\forall i$, $1 \leq i \leq m$, *un oráculo* Ω *tal que su función característica* f_Ω *es probabilísticamente pura, un clasificador mimético* μ *leal y "fence-and-fill" usando un conjunto de datos aleatorios R generado por un método exhaustivo etiquetado con* Ω. *Entonces:*

$$si \ |R| \to \infty \ \ entonces \ \ Prob_{e \in \mathcal{E}}[f_\mu(e) = f_\Omega(e)] = 1$$

Demostración. Como cualquier atributo X_i es numérico, podemos aplicar las definiciones que usan el concepto de distancia.

Puesto que R es generado por un método exhaustivo, tenemos que,

$$\forall \tau > 0, \forall e_1, e_2 \in \mathcal{E}, d(e_1, e_2) < \tau : Prob(\exists e' \in R, e' \in \overline{e_1 e_2}) > 0$$

Ahora bien, como $|R| \to \infty$, entonces

$$\forall \tau > 0, \forall e_1, e_2 \in \mathcal{E}, d(e_1, e_2) < \tau : \exists e' \in R, e' \in \overline{e_1 e_2} \qquad (4.1)$$

Esto significa que hay ejemplos alrededor de cualquier otro de su misma clase.
Puesto que el clasificador mimético es leal:

$$\forall e \in R : f_\mu(e) = f_\Omega(e)$$

Y como el clasificador mimético es "*fence-and-fill*":

$$Prob_{e \in \mathcal{E}}(\exists \tau > 0 \wedge Pure_\tau(e)) = 1$$

Usando la definición de $Pure_\tau(e)$ (Definición 3) tenemos:

$$Prob_{e\in\mathcal{E}}(\exists\tau>0\wedge(\forall e'\in\mathcal{E}(d(e,e')\leq\tau)\rightarrow f_{\mu}(e)=f_{\mu}(e')))=1$$

Por lo tanto, de ser leal, *fence-and-fill* y dado que $R\subseteq\mathcal{E}$ tenemos:

$$Prob_{e\in R}[\exists\tau>0\wedge(\forall e'\in\mathcal{E}(d(e,e')\leq\tau)\rightarrow f_{\mu}(e)=f_{\mu}(e'))]=1 \quad (4.2)$$

Eso significa que el clasificador mimético coincide con el oráculo en la mayoría de los puntos considerados, y sus puntos circundantes tienen la misma clase asignada que el punto central.

Como f_{Ω} es probabilísticamente puro

$$Prob_{e\in\mathcal{E}}\left[\exists\tau>0\wedge PPure_{\tau}(e)\right]=1$$

Extendiendo la definición de $PPure_{\tau}(e)$ tenemos:

$$Prob_{e\in\mathcal{E}}(\exists\tau>0\wedge Prob_{e'\in\mathcal{E}\,|\,d(e,e')\leq\tau}(f_{\Omega}(e)=f_{\Omega}(e'))=1)=1 \quad (4.3)$$

Esto significa que el oráculo sigue a los puntos circundantes a uno dado de misma clase que a éste.

De (4.2) y (4.3) tenemos:

$$Prob_{e\in R}[\exists\tau>0\wedge Prob_{e'\in\mathcal{E}\,|\,d(e,e')\leq\tau}[f_{\mu}(e')=f_{\Omega}(e')]=1]=1$$

Lo cual significa que el clasificador mimético coincide con el oráculo en la mayoría de los puntos considerados ($d(e,e')=0$), y sus puntos circundantes ($0<d(e,e')\leq\tau$).

De esta expresión y (4.1) tenemos que

$$Prob_{e\in\mathcal{E}}[\exists\tau>0\wedge Prob_{e'\in\mathcal{E}\,|\,d(e,e')\leq\tau}[f_{\mu}(e')=f_{\Omega}(e')]=1]=1$$

Lo cual significa que el clasificador mimético coincide con el oráculo en la mayoría de los puntos considerados, y sus puntos circundantes. Esto significa lógicamente que coincide en la mayoría de puntos, es decir:

$$Prob_{e\in\mathcal{E}}[f_\mu(e) = f_\Omega(e)] = 1$$

□

El lema anterior significa que con las condiciones impuestas el error se acercará a cero cuando generamos muchos ejemplos al azar para el clasificador mimético.

Y del lema anterior para las evidencias compuestas exclusivamente por atributos numéricos, podemos ahora establecer el resultado para evidencias compuestas de atributos numéricos y nominales.

Teorema 1. *Dado un espacio de evidencia $\mathcal{E}$ con atributos numéricos y nominales, un oráculo Ω tal que su función característica f_Ω es probabilísticamente pura, y un clasificador mimético μ leal y "fence-and-fill" usando un conjunto de datos aleatorios R generado por un método exhaustivo etiquetado con Ω. Entonces:*

$$si\ \ |R| \to \infty\ \ entonces\ \ \ Prob_{e\in\mathcal{E}}[f_\mu(e) = f_\Omega(e)] = 1$$

Demostración (*esbozo*). Si el espacio $\mathcal{E}$ consta solamente de atributos nominales o numéricos esto se demuestra con la proposición 1 o por el lema 1 respectivamente. Si ambos atributos nominales y numéricos existen en $\mathcal{E}$, podemos dividir este espacio, en el subespacio nominal $\mathcal{E}_\varphi$ y el subespacio numérico $\mathcal{E}_\rho$. Como ya discutimos en el inicio, $\mathcal{E}_\varphi$ es finito y cada elemento $\alpha \in \mathcal{E}_\rho$ es llamado una ancla. Puesto que la forma de generar ejemplos es exhaustiva, tenemos una probabilidad no nula para cada ancla, y por cada ancla tenemos las mismas condiciones que para el lema 1.

□

Hasta aquí, tenemos que para casi cualquier función de interés (probabilísticamente pura) representada por un oráculo, podemos aproximarla tan bien como deseamos con métodos *fence-and-fill* y con un conjunto de datos bastante grande.

4.3. Resultados para árboles de decisión

Debido a que el objetivo de la técnica mimética es obtener una representación comprensible de cualquier oráculo, sería interesante particularizar los resultados anteriores para los árboles

de decisión, pues es el método que usamos generalmente para construir el modelo mimético.

Proposición 3. *El modelo generado por un árbol de decisión sin podar es un clasificador leal y fence-and-fill.*

Demostración. Es leal porque está sin podar, y todas las hojas son puras. Por tanto, todos los ejemplos de entrenamiento son clasificados correctamente.

Es *fence-and-fill* porque el árbol de decisión es puro, ya que, para atributos numéricos, las particiones son de la forma

$$X_i < v \quad | \quad X_i \geq v$$

donde v es un número racional. Esto significa que, al final, el espacio formado por los atributos numéricos de $\mathcal{E}$ tiene una partición basada en regiones hipercúbicas para cada ancla. Esta clase de particiones (hipercubos) son puras para todos los puntos excepto para los puntos de las fronteras. Puesto que hay un número finito de condiciones en un árbol de decisión, el área de las fronteras es despreciable y, por lo tanto tenemos:

$$Prob_{e \in \mathcal{E}}[\exists \tau > 0 \wedge Pure_{\tau}(e)] = 1$$

lo cual es la definición de un clasificador puro. Por lo tanto, por la Definición 7 el clasificador es *fence-and-fill.*

□

Este resultado permite establecer la siguiente conclusión.

Corolario 1. *Dado un espacio de evidencia $\mathcal{E}$ con atributos numéricos y nominales, un oráculo Ω tal que su función característica f_Ω es probabilísticamente pura, y un clasificador μ generado por un árbol de decisión sin podar que imita a Ω usando un conjunto de datos aleatorios R generado por un método exhaustivo y etiquetado con Ω. Entonces:*

$$si \ |R| \to \infty \ \ entonces \ \ Prob_{e \in \mathcal{E}}[f_{\mu}(e) = f_{\Omega}(e)] = 1$$

Demostración. Trivial a partir del Teorema 1 y la Proposición 3.

□

Con los resultados anteriores tenemos que es posible capturar la semántica del oráculo, dado un conjunto de datos inventados suficientemente grande. Por supuesto que sería interesante relacionar la razón del error con el número de ejemplos. Sin embargo, esto depende de la resolución de f_Ω, de la dimensión del espacio y de la manera con la cual se generan los conjuntos de datos al azar. Pensamos que ningún resultado interesante se puede obtener de una coyuntura tan general. Sin embargo, podemos hacer algo interesante si consideramos un marco global pero nos dan las curvas de aprendizaje de los clasificadores. Esto se trata a continuación.

4.4. Curvas de aprendizaje e influencia del conjunto de datos de entrenamiento

En esta sección, vamos a considerar una función real o verdadera f_o (de la cual tenemos un conjunto de entrenamiento T y un conjunto de prueba). La función verdadera es aproximada por un oráculo y, entonces, el clasificador mimético busca aproximarse al oráculo. La primera configuración que vamos a analizar consiste en no usar el conjunto de datos de entrenamiento para la segunda etapa. Este escenario se muestra en la Figura 4.1.

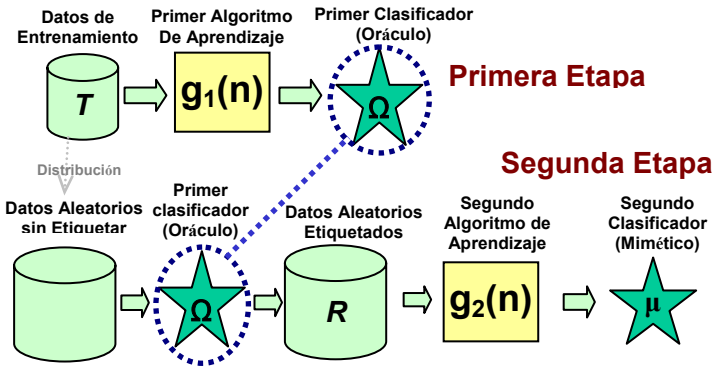


Figura 4.1 Proceso mimético sin conjunto de datos de entrenamiento en la segunda etapa.

Inicialmente, vamos a considerar que la primera etapa de aprendizaje tiene una fidelidad a_1 (porcentaje de T bien clasificado) y la segunda etapa de aprendizaje tiene una fidelidad a_2

con respecto al resultado de la primera etapa. De aquí, podemos intentar establecer la fidelidad global $a_{1:2}$.

Definición 8. *Dada la matriz de confusión normalizada C_{ij} de un clasificador con c clases, donde C_{ij} es el porcentaje de predecir la clase i cuando la clase verdadera es j, tal que si i=j entonces la predicción es correcta, mientras que si i≠j la predicción es incorrecta, decimos que un clasificador tiene los errores equitativamente distribuidos (todas la predicciones incorrectas tienen la misma tasa v) si*

$$\forall i, j \in [1,c], i \neq j : C_{ij} = v \; donde \; 0 \leq v \leq \frac{1}{c-1}$$

La definición 8 puede parecer una restricción muy fuerte, pero de alguna manera representa un caso "medio", es decir aquel en que los errores de los clasificadores en las dos etapas tienen una distribución equitativa.

Como la matriz está normalizada por columnas tenemos que para cada clase j-ésima $1 - C_{jj} = (1-c) \cdot C_{ij}, i \neq j$ con lo que $\forall i, j \; C_{jj} = C_{ii}$, es decir el porcentaje de acierto es el mismo para todas las clases.

Teorema 2. *Dado un escenario de aprendizaje en dos etapas para un problema con c clases, donde el primer clasificador tiene una fidelidad a_1 para este problema y el segundo clasificador tiene una fidelidad a_2 para el problema representado por la primera etapa. Suponiendo que los errores de ambos clasificadores están equitativamente distribuidos según la distribución de clases del problema, se cumple que:*

$$a_{1:2} = a_1 \cdot a_2 + (\frac{1}{c-1}) \cdot (1-a_1) \cdot (1-a_2)$$

Demostración intuitiva. La prueba es sencilla. Para la primera parte de la suma tenemos que considerar los casos donde la función objetivo y la función en la primera etapa coincidan (esto es a_1) y además los casos donde la función en la primera etapa y la función en la segunda etapa coinciden (a_2). La probabilidad de que ambas cosas sucedan viene dada por el primer sumando.

Ahora tendremos que considerar los casos en que las funciones no coinciden. Así, los casos donde la función objetivo y la función en la primera etapa no coincidan, vienen dados por 1-a_1. De todos estos casos tenemos que considerar aspectos en donde la función de la primera etapa y la función de la segunda etapa tampoco no coinciden, es decir, 1–a_2. Algunos de ellos pueden cambiar, corrigiendo el error anterior. Es fácil ver que esto sucede en precisamente 1/(c-1) de los casos si los errores de los clasificadores están bien distribuidos. Consecuentemente, la probabilidad de que la primera etapa falle pero la segunda etapa sea correcta por un error justo en el sentido contrario viene dada por la segunda parte de la suma.

Demostración formal. La fidelidad global viene dada por

$$a_{1:2} = \frac{\sum_{i=1}^{c}\sum_{j=1}^{c} n_i^1 \cdot C_{ij}^1 \cdot C_{ji}^2}{\sum_{i=1}^{c} n_i^1}$$

donde $n^1{}_i$ es el número de ejemplares pertenecientes a la clase i en el primer clasificador, $C^1{}_{ij}$ es la matriz de confusión del primer clasificador (el oráculo) y $C^2{}_{ij}$ es la matriz de confusión del segundo clasificador (el modelo mimético).

Dividamos en dos sumandos (los descritos en la demostración intuitiva)

$$a_{1:2} = \frac{\sum_{i=1}^{c} n_i^1 \cdot C_{ii}^1 \cdot C_{ii}^2 + \sum_{i=1}^{c}\sum_{j=1, i\neq j}^{c} n_i^1 \cdot C_{ij}^1 \cdot C_{ji}^2}{\sum_{i=1}^{c} n_i^1}$$

De acuerdo con la definición 8, los valores de porcentajes de aciertos (C_{ii}) y errores (C_{ij} y $i\neq j$) son iguales en cada una de las matrices de confusión. Si elegimos el primer valor como el valor respresentativo de todos ellos (C_{11} para los aciertos y C_{21} para los errores), entonces

$$a_{1:2} = \frac{C_{11}^1 \cdot C_{11}^2 \cdot \sum_{i=1}^{c} n_i^1 + C_{21}^1 \cdot \sum_{i=1}^{c} n_i^1 \cdot \sum_{j=1, i \neq j}^{c} C_{ji}^2}{\sum_{i=1}^{c} n_i^1}$$

Simplificando tenemos

$$a_{1:2} = C_{11}^1 \cdot C_{11}^2 + C_{21}^1 \cdot \sum_{j=1, i \neq j}^{c} C_{ji}^2 \qquad (4.4)$$

De acuerdo a la definición 8, para la primera clase tendremos

$$1 - C_{11}^1 = (1-c) \cdot C_{21}^1$$

Despejando el error C^1_{21} tenemos

$$C_{21}^1 = \frac{1 - C_{11}^1}{(1-c)}$$

Igualmente para el segundo clasificador

$$\sum_{j=1, i \neq j}^{c} C_{ji}^2 = 1 - C_{11}^2$$

También de la misma definición 8 se deduce que la fidelidad del primer clasificador es

$$C_{11}^1 = C_{22}^1 = C_{33}^1 = a_1$$

De forma análoga, la fidelidad del segundo clasificador es

$$C_{11}^2 = C_{22}^2 = C_{33}^2 = a_2$$

Si sustituimos estas equivalencias en la ecuación (4.4) llegamos a lo que queríamos demostrar

$$a_{1:2} = a_1 \cdot a_2 + (\frac{1}{c-1}) \cdot (1 - a_1) \cdot (1 - a_2)$$

□

Como hemos dicho antes, la condición de que los errores de los clasificadores están equitativamente distribuidos puede parecer un poco exigente. En general, sin embargo, si ambos clasificadores (el oráculo y el mimético) estuvieran sesgados en un sentido similar, el segundo sumando del Teorema 2 no sería muy diferente. Además, la condición es más fuerte cuanto mayor es el número de clases, pero es justamente en estos casos cuando el segundo sumando del teorema 2 es menos importante, ya que el denominador es función del número de clases.

El objetivo de la expresión del teorema 2 es no ignorar el caso de corrección de las dos etapas por fallos simétricos. Dicho de otra manera, aunque estas situaciones serán generalmente iguales o menores al segundo sumando, hemos decidido considerarlas en todos los análisis siguientes.

Finalmente, si a_1 y a_2 son altos podemos particularizar el resultado trivial anterior considerando la "precisión del oráculo" (la similaridad entre la función objetivo y el oráculo), la "fidelidad del mimético con respecto al oráculo" (la similaridad entre el oráculo y el clasificador mimético) y la "precisión mimética" (la similaridad entre la función objetivo y el clasificador mimético). Más formalmente:

Definición 9. La precisión de un oráculo está definida como:

$$Acc_{O,\Omega} = Prob_{e\in\mathcal{E}}(f_O(e) = f_\Omega(e))$$

Definición 10. La fidelidad de un clasificador μ con respecto a otro clasificador Ω esta definida como:

$$Acc_{\Omega,\mu} = Prob_{e\in\mathcal{E}}(f_\Omega(e) = f_\mu(e))$$

Definición 11. La precisión mimética esta definida como:

$$Acc_{O,\mu} = Prob_{e\in\mathcal{E}}(f_O(e) = f_\mu(e))$$

De esto, podemos obtener el siguiente resultado trivial:

Corolario 2. *Dada una función objetivo con c clases, un oráculo, un clasificador mimético y estando los errores de ambos clasificadores equitativamente distribuidos según la distribución de clases del problema, tenemos que:*

$$Acc_{O,\mu} = Acc_{O,\Omega} \cdot Acc_{\Omega,\mu} + (\frac{1}{c-1})(1 - Acc_{O,\Omega}) \cdot (1 - Acc_{\Omega,\mu})$$

Demostración. Es trivial y se obtiene del Teorema 2 y las definiciones 9, 10 y 11.

□

4.4.1. Curvas de aprendizaje

Hasta ahora, hemos considerado un error global y hemos dado por conocido el error cometido por cada etapa para un tamaño dado del conjunto de datos de entrenamiento T (para la primera etapa) y del conjunto de datos inventados R (para la segunda etapa). Con las condiciones asumidas en secciones anteriores, si aumentamos indefinidamente el tamaño de ambos, tendríamos que $a_1 \rightarrow 1$ y $a_2 \rightarrow 1$ y por lo tanto, que $a_{1:2} \rightarrow 1$. Pero, lógicamente el aumento indefinidamente de T no es posible puesto que nos dan generalmente un conjunto de datos de entrenamiento finito y el aumento indefinidamente de R no es posible debido a las restricciones computacionales (espacio y tiempo).

Consecuentemente, necesitamos generalizar la constante "fidelidad" a la variable "fidelidad". Más precisamente, desde ahora, consideramos una "curva de aprendizaje" para cada clasificador, la cual nos dice el valor específico de la fidelidad obtenida por un tamaño dado de un conjunto de datos.

Definición 12. *Dado un clasificador, una curva de aprendizaje de la fidelidad se define como una función* $g : \mathbb{N} \rightarrow \mathbb{R}$ *y* $\forall x : g(x) \in [0,1]$, *tal que dado un conjunto de datos D extraído de la función o problema objetivo* f_O *donde D es usada por el clasificador, entonces la fidelidad del clasificador con respecto a* f_O *está dada por* $g(n)$ *donde* $n=|D|$.

Obviamente, esa función g puede no existir incluso para un solo clasificador y problema, puede ser discontinua y, obviamente, puede depender no solamente del tamaño del conjunto

de datos sino también del orden de aparición de los ejemplos. Por lo tanto, consideramos funciones de "caso-promedio", es decir, funciones que son promedio de todas (o de una muestra representativa de) las combinaciones posibles de ejemplos de cada tamaño o también del comportamiento promedio para varios problemas[4]. No obstante, el uso de estimadores o funciones de curvas de aprendizaje no es nuevo, y se ha utilizado desde hace más de una década, estableciendo límites teóricos (dependiendo de la dimensión de Vapnik-Chervonenkis o de otras dimensiones del problema) y aproximaciones específicas (véase por ejemplo (Haussler et al., 1992; Walpert y Macready, 1997)).

Por lo tanto, partimos de la suposición de que la función puede ser estimada y promediada sobre muchos subconjuntos de la evidencia $\mathcal{E}$o para muchos problemas. Por otra parte, y lo que es interesante, puede ser utilizada para modelar algunos comportamientos y, como veremos después, para entender la razón de por qué algunas veces una técnica particular no trabaja bien o ver qué opciones son mejores.

Las prestaciones de los clasificadores con respecto al número de ejemplos vistos generalmente tiene una forma de una curva con crecimiento exponencial inverso, que comienza por la precisión dada por un clasificador al azar $1/c$ (donde c es el número de clases) y luego crece rápidamente al inicio y después más lentamente hasta alcanzar un máximo. Esta clase de comportamiento puede ser modelado por la siguiente curva de aprendizaje:

Definición 13. *Una curva de aprendizaje de saturación de la fidelidad se define como la función*

$$s_{[a.\alpha]}(n) = (a - \frac{1}{c})(1 - e^{-\alpha n}) + \frac{1}{c}$$

[4] No podemos considerar todos los problemas posibles (al menos todos ellos con la misma probabilidad) debido al teorema *free-lunch* (Walpert y Macready, 1997) sin otras suposiciones adicionales (probabilidad a priori universal, p. ej.), pero podemos considerar una muestra representativa de los problemas (por ejemplo, los conjuntos de datos del repositorio UCI).

donde a es la máxima fidelidad o punto de saturación y α es la tasa de crecimiento (es un valor que define la rápidez con la que la curva alcanza la máxima fidelidad).

Un ejemplo de curva de aprendizaje de saturación $s_{[0.9,0.05]}(n)$ para un problema de 5 clases, se muestra en la Figura 4.2.

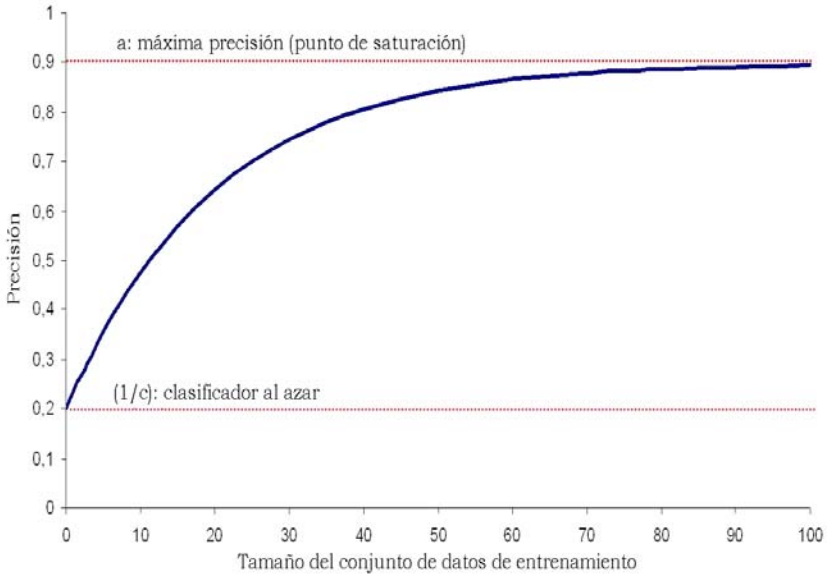


Figura 4.2 Una curva de aprendizaje aproximándose a su punto de saturación.

No es difícil considerar un algoritmo particular de aprendizaje y aprender uno o más modelos usando conjunto de datos de entrenamiento de talla creciente. En muchos casos, la función anterior puede adecuarse bien al comportamiento del algoritmo y del conjunto de datos, permitiendo obtener los parámetros a y α para extrapolar para conjuntos de datos más grandes.

En lo que sigue consideramos funciones de curvas generales para los resultados teóricos y utilizamos a veces las curvas de saturación para los ejemplos.

Y ahora, una vez presentada la función de la curva de aprendizaje, y teniendo un escenario donde se aplica directamente el segundo clasificador podemos dar una caracterización de cuándo es útil usar el método mimético.

Teorema 3. *Dado un escenario de aprendizaje de dos etapas para un problema con c clases, donde el primer clasificador tiene una curva de aprendizaje de fidelidad g_1 para este problema y el segundo clasificador tiene una fidelidad g_2 tanto para el problema original como para el problema representado por la primera etapa, y dado un conjunto de datos T (para la primera etapa, que es correcto con respecto a la función objetivo) de tamaño t=|T| y un conjunto de datos R (para la segunda etapa) de tamaño r= |R| y asumiendo que los errores de ambos clasificadores están equitativamente distribuidos según la distribución de clases del problema, entonces el procedimiento mimético es útil si y sólo si:*

$$g_2(t) < g_1(t) \cdot g_2(r) + (\frac{1}{c-1})(1-g_1(t)) \cdot (1-g_2(r))$$

Demostración. Si no hacemos uso del procedimiento mimético con datos inventados, entonces podemos usar sólo el conjunto de entrenamiento *T* para derivar el segundo clasificador y tenemos una precisión de g_2(t) (la parte izquierda de la desigualdad). Si usamos el procedimiento mimético, por la corolario 2 tenemos que la precisión obtenida es la parte derecha de la desigualdad. Por lo tanto, la técnica mimética es útil cuando ésta última precisión es mejor que la obtenida usando sólo los datos de entrenamiento con el segundo clasificador.

□

Veamos con algunos ejemplos que ilustran cuándo podemos determinar el uso o no de la técnica mimética.

Ejemplo 1

Considere una red neuronal con una función de fidelidad g_1(n) = $s_{[0.9,0.05]}$(n) y un árbol de decisión con una función de fidelidad g_2(n) = $s_{[0.8,0.1]}$(n). El número de clases es 2. Tenemos un conjunto de datos de entrenamiento con 50 ejemplos. ¿Es razonable utilizar el método mimético usando la red neuronal como oráculo y el árbol de decisión como algoritmo para obtener el modelo mimético?

Primero obtengamos las fidelidades resultantes al aplicar directamente los clasificadores. Para la red neuronal, aplicando la definición 13, tenemos

$$s_{[0.9.0.05]}(50) = (0.9 - \frac{1}{2})(1 - e^{-0.05(50)}) + \frac{1}{2} = 0.867$$

Mientras que para el árbol de decisión se obtiene

$$s_{[0.8.0.1]}(50) = (0.8 - \frac{1}{2})(1 - e^{-0.1(50)}) + \frac{1}{2} = 0.775$$

Resulta que usando el árbol de decisión (con un conjunto de datos al azar infinito) se obtiene una fidelidad de

$$s_{[0.8.0.1]}(\infty) = (0.8 - \frac{1}{2})(1 - e^{-0.1(\infty)}) + \frac{1}{2} = (0.3)(1 - 0) + 0.5 = 0.8$$

con respecto a la salida de la red neuronal. Por lo tanto, la fidelidad alcanzada en las dos etapas del método mimético es de 0.694 (0.867×0.8). Como esta precisión es menor que 0.775 (la precisión del árbol de decisión empleado de forma directa) la respuesta es que no es beneficioso usar en esta situación el método mimético.
De hecho, en estas dos curvas no hay un tamaño de entrenamiento para el cual el método mimético es útil. Este es un ejemplo donde el primer clasificador es bueno pero "lento" y el segundo es malo pero "rápido". El mimético es inútil.

Ejemplo 2

Considere un clasificador de regresión logística con una función de fidelidad $g_1(n) = s_{[0.9,0.05]}(n)$ y un árbol de decisión con una función de fidelidad $g_2(n) = s_{[0.8,0.01]}(n)$. El número de clases es 2. Tenemos un conjunto de datos de entrenamiento con 100 ejemplos. ¿Es razonable utilizar el método mimético? ¿Y con 200 ejemplos? ¿Y con 300 ejemplos?

Para 100 ejemplos la respuesta es sí. Resulta que el uso del mimético (con un conjunto de datos al azar infinito) logrará una precisión de 0.721 mientras que el uso directo del segundo clasificador obtiene 0.579 en precisión.

Con 200 ejemplos de entrenamiento tenemos que el mimético (con un conjunto de datos al azar infinito) logrará una precisión de 0.725 mientras que el uso directo del segundo clasificador obtiene 0.719. La respuesta es también sí.

Con 300 ejemplos de entrenamiento tenemos que el mimético (con un conjunto de datos al azar infinito) logrará una precisión de 0.725 mientras que el uso directo del segundo clasifi-

cador obtiene 0.770. La respuesta es no.

Éste es un ejemplo donde el primer clasificador es bueno y rápido y el segundo es malo y lento. En esta situación vemos que en algunos casos el método mimético es útil mientras que en otros no.

Ejemplo 3

Considere un clasificador Bayesiano con una función de fidelidad para la curva de aprendizaje $g_1(n) = s_{[0.8,0.1]}(n)$ y un árbol de decisión con una función de fidelidad para la curva de aprendizaje $g_2(n) = s_{[0.9,0.01]}(n)$. Tenemos un conjunto de datos de entrenamiento con 100 ejemplos. ¿Es razonable utilizar el método mimético? ¿Y con 300 ejemplos?

La respuesta es sí para 100 ejemplos. Resulta que el uso del mimético (con un conjunto de datos al azar infinito) logrará una precisión de 0.725 mientras que el uso directo del segundo clasificador obtiene 0.642 en precisión. Con 300 ejemplos de entrenamiento tenemos que el mimético (con un conjunto de datos al azar infinito) logrará una precisión de 0.725 mientras que el uso directo del segundo clasificador obtiene 0.865. La respuesta es no en este caso. Éste es un ejemplo donde el primer clasificador es malo y rápido y el segundo es bueno y lento. En algunos casos el método mimético es útil.

En las Figuras 4.3 y 4.4 ilustramos dos casos típicos en una gráfica que llamamos "Gráfica de Respuesta Mimética". Esta gráfica muestra tres curvas: la curva de aprendizaje del primer clasificador, la curva de aprendizaje del segundo clasificador si fuera aprendido con diferentes tamaños del conjunto de datos de entrenamiento y, además, mostramos la curva de aprendizaje del clasificador mimético, considerando un conjunto de datos aleatorios infinito. Estos gráficos son bastante ilustrativos, porque podemos ver si el método es inútil (si la curva mimética está debajo de la curva del segundo clasificador) o en caso de ser útil, para qué tamaños de entrenamiento.

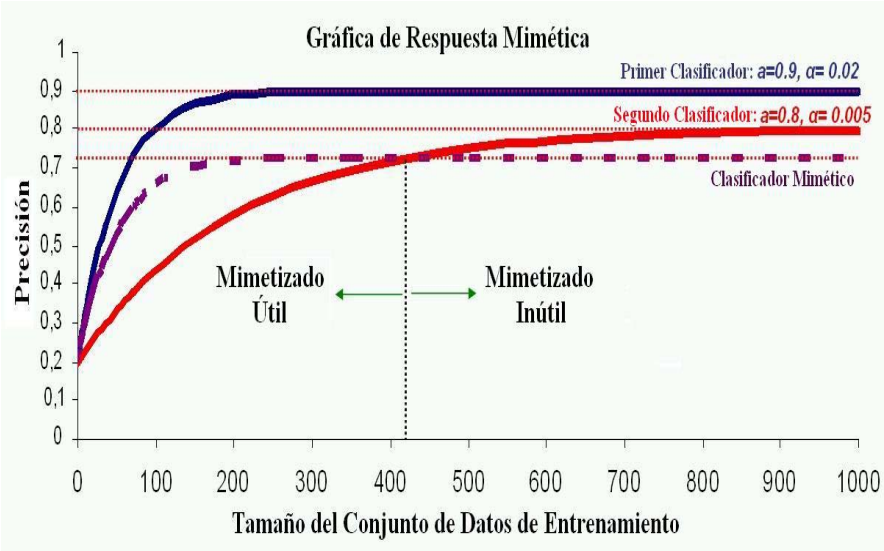


Figura 4.3 Gráfica de respuesta mimética para determinar cuándo el método es útil.

La Figura 4.3 muestra un caso donde el método mimético es útil para conjuntos de datos de entrenamiento de talla pequeña (número de ejemplos menor que 415).

La Figura 4.4 muestra un caso donde el método mimético es inútil independientemente del tamaño del conjunto de entrenamiento.

Figura 4.4 Gráfica de respuesta mimética cuándo el método es inútil.

Según los resultados y los razonamientos anteriores, hemos visto que el elemento más importante de ambos clasificadores es su curva de aprendizaje. Esto es aún más importante que sus precisiones máximas. Hay casos donde clasificadores rápidos y malos pueden ser utilizados para la primera etapa y clasificadores lentos y buenos se pueden utilizar para la segunda, especialmente cuando no hay muchos ejemplos disponibles.

No obstante, debe indicarse que el procedimiento mimético puede ser mejor de lo esperado porque g_2 puede ser mejor para el oráculo que para la función objetivo. Esto puede suceder porque el oráculo es generalmente un dispositivo algorítmico y por lo tanto es puro. Por los resultados de la sección anterior, esto significa que los métodos *fence-and-fill* para la segunda etapa capturarán correctamente la semántica del oráculo para un conjunto de datos al azar grande. Formalicemos este razonamiento.

Lema 2. *En un escenario mimético, si el clasificador de la primera etapa (el oráculo) es probabilísticamente puro y el clasificador de la segunda etapa es puro y leal entonces la curva de aprendizaje del caso promedio del segundo clasificador sigue la condición:*

$$si\ |R| \to \infty\ \ entonces\ \ g'_2(|R|) = 1$$

Usamos g'_2 para denotar este buen comportamiento y sólo sucede con respecto al oráculo. Reservamos g_2 para la curva de aprendizaje con respecto a la función objetivo.

Demostración. Directa por el Teorema 1.

□

Y ahora, a partir de esto, podemos afirmar el teorema siguiente:

Teorema 4. *Dado un escenario mimético con un conjunto de entrenamiento T con tamaño t= |T|. Si el clasificador de la primera etapa (el oráculo) con curva de aprendizaje g_1 es probabilísticamente puro y el clasificador de la segunda etapa con curva de aprendizaje g_2 es puro y leal, y los errores de ambos clasificadores están equitativamente distribuidos según la distribución de clases del problema, entonces el clasificador mimético será útil si y solo si:*

$$g_2(t) < g_1(t)$$

Demostración. Del teorema 3 tenemos:

$$g_2(t) < g_1(t) \cdot g'_2(r) + (\frac{1}{c-1})(1 - g_1(t)) \cdot (1 - g'_2(r))$$

En vista de que podemos generar tantos ejemplos al azar como deseamos, entonces tenemos $|R| \to \infty$. Por lo tanto, del lema 2 tenemos:

$$g_2(t) < g_1(t) \cdot 1 + (\frac{1}{c-1})(1 - g_1(t)) \cdot (1 - 1)$$

$$g_2(t) < g_1(t) + (\frac{1}{c-1})(1 - g_1(t)) \cdot (0)$$

$$g_2(t) < g_1(t)$$

□

Esto es un resultado muy positivo y se puede ver en el gráfico de respuesta mimética que siempre que haya una zona donde la curva del primer clasificador está sobre la curva del segundo clasificador, entonces el método mimético es útil. Obsérvese

que en el gráfico de respuesta mimética mostramos el comportamiento del segundo clasificador respecto a la función objetivo, no respecto al oráculo. Éste es precisamente el aspecto en el que acabamos de discutir, porque el segundo clasificador (mimético) tiende a 1 mientras que el primer clasificador (oráculo) no. Por lo tanto, en un escenario mimético habitual, será preferible usar la técnica mimética a usar la segunda técnica sobre los datos de entrenamiento. Esta es la situación típica cuando tenemos un primer clasificador muy potente pero incomprensible (una red neuronal, un multiclasificador, etc.) y el segundo clasificador es menos potente pero comprensible.

4.4.2. Influencia del conjunto de datos de entrenamiento original

Hecho todo esto, ahora podemos considerar la situación donde, además del conjunto de datos al azar R, el clasificador mimético también utiliza el conjunto de entrenamiento T. Por lo tanto, tenemos un conjunto de datos usado en la segunda etapa $I = R \cup T$. Este conjunto de datos será referido como el conjunto de datos inventados o el conjunto de datos fusión (o unión). Esto se muestra en el Figura 3.2 (en la página 47).

Para esta nueva situación, tenemos que redefinir nuestros resultados previos.

Primero, tenemos que generalizar nuestra noción de curva de aprendizaje para tener en cuenta dos conjuntos de datos de diferentes distribuciones. Incluso considerando valores promedio, está claro que la función $g(n)$ es justamente una idealización que en muchos casos deberá ser solamente aproximada. En el caso de fusionar R y T ambos conjuntos de datos intentan modelar el mismo problema pero se deben considerar dos distribuciones diferentes. Por lo tanto, debemos ampliar la definición de la curva de aprendizaje para la segunda etapa. Primero necesitamos definir un clasificador combinado promedio:

Definición 14. *Dados dos clasificadores f_1 y f_2 y una proporción $r \in [0,1]$. Un clasificador combinado promedio se define como $C_{[r](f_1,f_2)}$, donde las predicciones de C para un ejemplo e son promediadas de manera que:*

$$si\ f_1(e) = f_2(e) = i\ \ entonces\ \ C_{[r](f_1,f_2)}(e) = i$$

$$en\ otro\ caso \begin{cases} Prob(C_{[r](f_1,f_2)}(e) = f_1(e)) = r \\ Prob(C_{[r](f_1,f_2)}(e) = f_2(e)) = 1 - r \end{cases}$$

Definición 15. *Dado un conjunto de entrenamiento de un clasificador con datos de dos distribuciones, una curva de aprendizaje de fidelidad proporcional se define como una función* $g: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}$ *tal que* $\forall x,y : g_1(x,y) \in [0,1]$ *y dado un conjunto de datos* D_1 *extraído de la función o problema objetivo* f_1 *y un conjunto de datos* D_2 *extraído de la función o problema objetivo* f_2*, donde* $D_1 \cup D_2$ *es usada por el clasificador, entonces la fidelidad del clasificador con respecto al clasificador combinado* $C_{[n1/(n1+n2)](f1,f2)}$ *viene dada por* $g(n_1,n_2)$ *donde* $n_1 = |D_1|$ *y* $n_2 = |D_2|$.

Con esta generalización de la curva de aprendizaje, podemos ahora tratar la influencia de la aplicación del conjunto de datos de entrenamiento agregado al conjunto de datos al azar:

Teorema 5. *Consideremos un escenario de aprendizaje de dos etapas con c clases, un conjunto de datos T (para la primera etapa, el cual es considerado correcto con respecto a la función objetivo) con tamaño* $t = |T|$ *y un conjunto de datos* $I = R \cup T$ *(para la segunda etapa) de tamaño* $i = |I|$*, donde R es un conjunto de datos generados aleatoriamente a partir del la primer etapa con un método exhaustivo y de tamaño* $r = |R|$*. Considere también que el primer aprendizaje es puro y tiene una función para la curva de aprendizaje de la fidelidad* g_1 *para la función objetivo, y el segundo aprendizaje tiene fidelidad* $g_2(r,t)$ *para el problema representado por el conjunto de datos inventado. Supongase también que los errores de ambos clasificadores están equitativamente distribuidos según la distribución de clases del problema.*

Para un número suficientemente grande de ejemplos aleatorios y ejemplos de entrenamiento, y considerando que R y T no tienen ejemplos en común, tenemos que la función de la curva de aprendizaje conjunta $g_{1:2}(r,t)$ viene dada por*:*

$$g_{1:2}(r,t) = \frac{r}{r+t} \cdot [g_1(t) \cdot g_2(r,t) + (\frac{1}{c-1})(1-g_1(t)) \cdot (1-g_2(r,t))]$$
$$+ (1 - \frac{r}{r+t}) \cdot g_2(r,t)$$

Demostración. Si consideramos la función de fidelidad del clasificador combinado "promedio" $G_1(r,t)$ antes de la segunda etapa, tenemos:

$$G_1(r,t) = g_1(t) \cdot \frac{r}{r+t} + \frac{t}{r+t}$$

La primera parte tiene el factor $g_1(t)$ porque viene del clasificador de la primera etapa a través de R. La segunda parte viene de la función objetivo directamente a través de T, que se considera correcta.

Y ahora, tenemos una función de la primera etapa $G_1(r,t)$ y una función de la curva de aprendizaje para la segunda etapa $g_2(r,t)$. Del teorema 2 y el corolario 2 tenemos:

$$g_{1:2}(r,t) = G_1(r,t) \cdot g_2(r,t) + (\frac{1}{c-1})(1-G_1(r,t)) \cdot (1-g_2(r,t))$$

Si usamos la expresión de $G_1(r,t)$ en la fórmula de arriba, tenemos:

$$g_{1:2}(r,t) = [g_1(t) \cdot \frac{r}{r+t} + \frac{t}{r+t}] \cdot g_2(r,t) +$$
$$(\frac{1}{c-1})(1-[g_1(t) \cdot \frac{r}{r+t} + \frac{t}{r+t}]) \cdot (1-g_2(r,t)) =$$
$$\frac{r}{r+t} g_1(t) \cdot g_2(r,t) + \frac{t}{r+t} \cdot g_2(r,t) + (\frac{1}{c-1})(\frac{r}{r+t} - g_1(t)\frac{r}{r+t}) \cdot (1-g_2(r,t)) =$$

$$\frac{r}{r+t} g_1(t) \cdot g_2(r,t) + \frac{t}{r+t} \cdot g_2(r,t) + \frac{1}{c-1}\frac{r}{r+t}(1-g_1(t)) \cdot (1-g_2(r,t)) =$$
$$\frac{r}{r+t}[g_1(t) \cdot g_2(r,t) + \frac{1}{c-1}(1-g_1(t)) \cdot (1-g_2(r,t))] + (1-\frac{r}{r+t}) \cdot g_2(r,t)$$

□

El teorema previo establece un modelo general para esta variante del método mimético, donde el conjunto de datos original es usado en ambas etapas. El problema del modelo previo es que $g_2(r,t)$ puede ser difícil de estimar para todo r y t. La siguiente aproximación se basa en la definición de tres modelos específicos, de acuerdo a algunas suposiciones en la curva de aprendizaje $g_2(r,t)$. Estos modelos son:

Modelo 1 (Ideal): Asumimos que $\forall g_2(r,t) = g_2(r+t)$ donde g_2 es la función de la curva de aprendizaje del segundo clasificador. Esto asume que R y T son altamente similares.
Modelo 2 (Optimista): Asumimos que $\forall g_2(r,t) = g_2(max(r,t))$ donde g_2 es la función de la curva de aprendizaje del segundo clasificador. Puesto que r será generalmente mayor que t, simplemente consideraremos $g_2(r,t) = g_2(r)$. Esta suposición se realiza si R y T son similares, pero T agrega complejidad a R y una tasa de aprendizaje lenta.
Modelo 3 (Pesimista): Asumimos que $\forall g_2(r,t) = g_2(min(r,t))$ donde g_2 es la función de la curva de aprendizaje del segundo clasificador. Puesto que t será generalmente menor que r, simplemente consideraremos $g_2(r,t) = g_2(t)$.

Por supuesto podemos considerar modelos aún más pesimistas que el modelo 3, pero asumimos que por lo menos R y T intentan modelar la misma función (es decir el clasificador de la primera etapa, el oráculo, no es muy malo).

Empecemos con el modelo 1.

Proposición 4. *Bajo el modelo 1, el procedimiento mimético es útil si y solo si:*

$$\frac{r}{r+t}\cdot[g_1(t)\cdot g_2(r+t)+(\frac{1}{c-1})(1-g_1(t))\cdot(1-g_2(r+t))]$$
$$+(1-\frac{r}{r+t})\cdot g_2(r+t)$$
$$> g_2(t)$$

Demostración. Usando la suposición del modelo 1 en el teorema 5 tenemos la primera expresión. La segunda expresión corresponde al aprendizaje del segundo clasificador sobre el conjunto de entrenamiento solamente.

□

Proposición 5. *Bajo el modelo 1, para un n dado, si* $g_1(n)<1$ *y* $g_2(n)>1/c$ *entonces es beneficioso utilizar el conjunto de entrenamiento.*

Demostración. Bajo el modelo 1, podemos expresar la precisión final de la estructura mimética usando el conjunto de entrenamiento como:

$$\frac{r}{r+t}\cdot[g_1(t)\cdot g_2(r+t)+(\frac{1}{c-1})(1-g_1(t))\cdot(1-g_2(r+t))]$$
$$+(1-\frac{r}{r+t})\cdot g_2(r+t) \qquad (4.5)$$

Concentrémonos en la segunda parte de la primera expresión y demostremos primero esto:

$$(\frac{1}{c-1})(1-g_1(t))\cdot(1-g_2(r+t))<(1-g_1(t))\cdot g_2(t+r) \qquad (4.6)$$

Ya que $g_1(n)<1$, entonces $(1-g_1(t))>0$, por lo que la expresión anterior se simplifica a:

$$(\frac{1}{c-1})\cdot(1-g_2(t+r))<g_2(r+t)$$

$$(1-g_2(t+r))<(c-1)\cdot g_2(t+r)$$

$$1<c\cdot g_2(t+r)$$

$$1/c<g_2(t+r)$$

que es verdad por la suposición.

Y ahora, considerando (4.5) tenemos que:

$$
\begin{aligned}
&[g_1(t)\cdot g_2(r+t)+(\frac{1}{c-1})(1-g_1(t))\cdot(1-g_2(r+t))]\\
&<[g_1(t)\cdot g_2(r+t)+(1-g_1(t))\cdot g_2(r+t)]\\
&=[g_2(r+t)]
\end{aligned}
$$

Por tanto

$$[g_1(t)\cdot g_2(r+t)+(\frac{1}{c-1})(1-g_1(t))\cdot(1-g_2(r+t))]<g_2(r+t)$$

□

De acuerdo a las proposiciones anteriores, podemos ver que en general se incrementan los casos donde la técnica mimética es útil.

Los resultados confirman nuestra intuición, por lo que realmente no aportan mucha información. Sin embargo, podemos usarlos para clarificar qué hacer en algunos casos.

Ejemplo 4

Considere una red neuronal con una función de fidelidad $g_1(n) = s_{[0.9,0.05]}(n)$ y un árbol de decisión con una función de fidelidad $g_2(n) = s_{[0.8,0.1]}(n)$. Tenemos un conjunto de datos de entrenamiento con 50 ejemplos. ¿Es razonable utilizar el método mimético?

Si sustituimos los valores de la red neuronal en la ecuación de la definición 13, tenemos

$$s_{[0.9.0.05]}(50)=(0.9-\frac{1}{2})(1-e^{-0.05(50)})+\frac{1}{2}=0.867$$

Mientras que el uso directo del segundo clasificador obtiene

$$s_{[0.8.0.1]}(50)=(0.8-\frac{1}{2})(1-e^{-0.1(50)})+\frac{1}{2}=0.775$$

Resulta que el uso del mimético (con un conjunto de da-

tos al azar infinito) logra una fidelidad de

$$s_{[0.8.0.1]}(\infty) = (0.8 - \frac{1}{2})(1 - e^{-0.1(\infty)}) + \frac{1}{2} = 0.3 + 0.5 = 0.8$$

$$g_2(\infty,50) = g_2(50) \cdot \frac{\infty}{\infty + 50} + \frac{50}{\infty + 50}$$
$$g_2(\infty,50) = g_2(50)$$

Ahora calculamos la fidelidad global cuando tenemos r $\rightarrow \infty$ utilizando la ecuación del teorema 5, tenemos

$$g_{1:2}(\infty,50) = \frac{\infty}{\infty + 50} \cdot [g_1(50) \cdot g_2(\infty,50) + (\frac{1}{2-1})(1 - g_1(50)) \cdot (1 - g_2(\infty,50))] + (1 - \frac{\infty}{\infty + 50}) \cdot g_2(\infty,50)$$

$$g_{1:2}(\infty,50) = (g_1(50) \cdot g_2(\infty,50) + (\frac{1}{2-1})(1 - g_1(50)) \cdot (1 - g_2(\infty,50))$$

Como la fidelidad del clasificador combinado "promedio" es

$$g_2(\infty,50) = g_2(50)$$

entonces

$$\begin{aligned} g_{1:2}(\infty,50) &= (g_1(50) \cdot g_2(50) + (1 - g_1(50)) \cdot (1 - g_2(50)) = \\ &= (0.867 \cdot 0.775) + (1 - 0.867) \cdot (1 - 0.775) = \\ &= 0.672 + (0.133) \cdot (0.225) = \\ &= 0.672 + 0.03 = \\ &= 0.702 \end{aligned}$$

Resulta que la fidelidad del segundo clasificador con un conjunto de datos aleatorios infinito es de 0.702, por lo que combinando las dos etapas del proceso mimético se

obtiene una precisión de 0.609 (0.867×0.702). Si dibujamos la curva de aprendizaje del mimético podemos observar un máximo en precisión de 0.75 con un tamaño del conjunto de datos aleatorios de 30 ejemplos mientras que el uso directo del segundo clasificador se obtiene una precisión de 0.775. La respuesta es no, debido a que la precisión del mimético es menor que la del segundo clasificador sólo (0.75<0.775) en el mejor de los casos. De hecho, para estas dos curvas no hay tamaño de entrenamiento para el cual el método mimético sea útil. Éste es un ejemplo donde es bueno el primer clasificador pero lento y el segundo es malo pero rápido. El mimético es inútil. Estos resultados son iguales a los obtenidos sin el conjunto de datos de entrenamiento.

Ejemplo 5

Considere un clasificador de regresión logística con una función de fidelidad $g_1(n) = s_{[0.9,0.05]}(n)$ y un árbol de decisión con una función de fidelidad $g_2(n) = s_{[0.8,0.01]}(n)$. Tenemos un conjunto de datos de entrenamiento con 100 ejemplos. ¿Es razonable utilizar el método mimético? ¿Y con 200 ejemplos? ¿Y con 300 ejemplos?

Para 100 ejemplos la respuesta es sí. Resulta que la precisión del segundo clasificador tiene un máximo con un conjunto de datos aleatorios de 500 ejemplos (con un conjunto de datos al azar infinito es menor) dando al menos 0.735 de precisión tras las dos etapas (sin el entrenamiento es 0.721) mientras que con el uso directo del segundo clasificador con el conjunto de entrenamiento se obtiene 0.579 en precisión.

Con 200 ejemplos de entrenamiento tenemos que el mimético tiene un máximo con un tamaño del conjunto de datos aleatorio de 500 ejemplos (con un conjunto de datos al azar infinito es menor) dando como mucho 0.745 de precisión (sin el entrenamiento era 0.725) mientras que con el uso directo del segundo clasificador se obtiene 0.719 de precisión. La respuesta es sí. Con 200 ejemplos de entrenamiento tendríamos que sería útil con el conjunto de datos de entrenamiento pero inútil sin él.

Con 300 ejemplos de entrenamiento tenemos que el mi-

mético tiene un máximo con un tamaño del conjunto de datos aleatorio de 500 ejemplos (con un conjunto de datos al azar infinito es menor) dando 0.752 de precisión (sin el entrenamiento daría 0.725) mientras que con el uso directo del segundo clasificador se obtiene 0.770 de precisión. La respuesta es no.

Éste es un ejemplo donde el primer clasificador es bueno y rápido y el segundo es malo y lento. En esta situación vemos que en algunos casos el método mimético es útil mientras que en otros no. En comparación con ejemplos de secciones anteriores, el uso del conjunto de datos de entrenamiento genera más casos útiles.

Ejemplo 6

Considere un clasificador Bayesiano con una función de fidelidad $g_1(n) = s_{[0.8,0.1]}(n)$ y un árbol de decisión con una función de fidelidad $g_2(n) = s_{[0.9,0.01]}(n)$. Tenemos un conjunto de datos de entrenamiento con 100 ejemplos. ¿Es razonable utilizar el método mimético? ¿Y con 150 ejemplos? ¿Y con 200 ejemplos? ¿Y con 300 ejemplos?

La respuesta a la primera pregunta es sí. Resulta que el uso del mimético tiene un máximo con un conjunto de datos aleatorios de 500 ejemplos (con un conjunto de datos al azar infinito es menor) dando al menos 0.751 de precisión (sin el entrenamiento es 0.725) mientras que el uso directo del segundo clasificador se obtiene 0.642.

Con 150 ejemplos de entrenamiento tenemos que el mimético tiene un máximo con un tamaño del conjunto de datos aleatorio de 500 ejemplos (con un conjunto de datos al azar infinito es menor) dando a lo mucho 0.763 de precisión (sin el entrenamiento dio 0.725) mientras que el uso directo del segundo clasificador obtiene 0.744. La respuesta es sí (sin el conjunto de entrenamiento la respuesta era no).

Con 200 ejemplos de entrenamiento tenemos que el mimético tiene un máximo con un tamaño del conjunto de datos aleatorio de 500 ejemplos (con un conjunto de datos al azar infinito es menor) dando como mucho 0.773 (sin el entrenamiento daría 0.725) mientras que con el uso directo del segundo clasificador se obtiene 0.805. La

respuesta es no.

Con 300 ejemplos de entrenamiento tenemos que el mimético tiene un máximo con un tamaño del conjunto de datos aleatorio de 200 ejemplos (con un conjunto de datos al azar infinito es menor) dando al menos 0.799 de precisión (sin el entrenamiento dio 0.725) mientras que con el uso directo del segundo clasificador se obtiene 0.865. La respuesta es no.

Este es un ejemplo donde el primer clasificador es malo y rápido y el segundo es bueno y lento. En esta situación vemos que en algunos casos el método mimético es útil mientras que en otros no. En comparación con los ejemplos de las secciones anteriores, el uso del conjunto de datos de entrenamiento genera más casos útiles.

Resumiendo, puede haber casos donde el incremento de los ejemplos aleatorios es siempre bueno. Existen otros casos donde la técnica mimética no se puede usar. Pero, según lo demostrado en algunos ejemplos anteriores y a diferencia de lo que sucedió con el método mimético sin el conjunto de datos de entrenamiento, cuando utilizamos el conjunto de datos de entrenamiento junto con el conjunto de datos aleatorios hay algunos casos donde es buena la técnica mimética hasta cierta cantidad de datos aleatorios pero a partir de allí el aumento de los datos aleatorios ya no es beneficioso.

Es interesante notar que para algunas combinaciones de curvas de aprendizaje puede haber tres zonas diferentes. La siguiente curva generalizada de respuesta (véase Figura 4.5) muestra en tres dimensiones la influencia del tamaño del conjunto de datos aleatorio (el primer clasificador tiene una curva de aprendizaje $s_{[0.8,0.1]}$ y el segundo $s_{[0.7,0.02]}$):

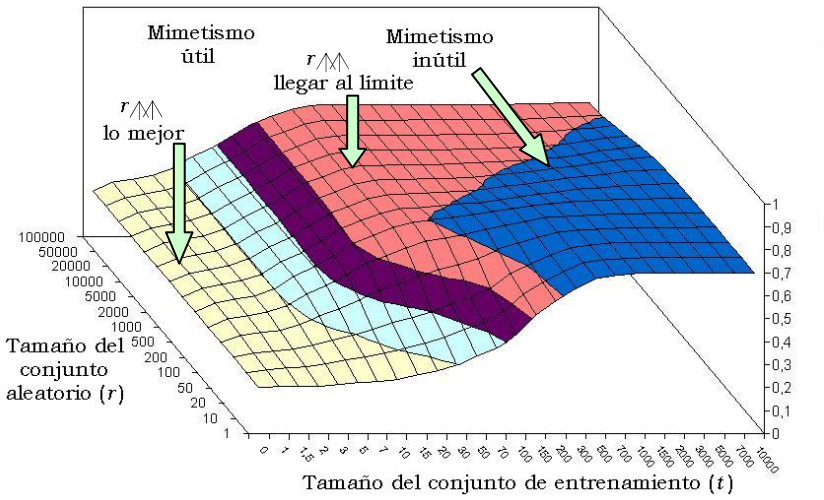


Figura 4.5 Gráfica de respuesta mimética generalizada cuando usamos el conjunto de datos de entrenamiento.

Aunque parece contraintuitivo, hay una zona para la cual el aumento del conjunto de datos aleatorio no es beneficioso. Esto es una lástima, porque éste es uno de los parámetros más fáciles que se pueden modificar para aumentar la calidad del proceso entero.

Afortunadamente, parece que puede haber una salida en estas zonas extrañas. La idea se basa en la observación de que el gráfico anterior siempre está aumentando para cualquier (r,t) a $(r{\cdot}k\ ,t{\cdot}k)$ cuando $k > 1$. Probémoslo:

Proposición 6. *Bajo el modelo 1 y funciones no decrecientes de la curva de aprendizaje, si incrementamos r y t en la misma proporción* (*es decir* $(\cdot r \cdot k\ ,\ t \cdot k)$ *donde* $k>1$) *entonces tenemos una precisión total igual o mayor.*

Demostración. Del modelo 1 tenemos la expresión (4.5).
Es trivial ver que la proporción $r/(r+t)$ permanece constante después de incrementar (r a $r{\cdot}k$ y t a $t{\cdot}k$). Por lo tanto, tenemos que demostrar que

$$g_1(t)\cdot g_2(r+t)+(\frac{1}{c-1})(1-g_1(t))\cdot(1-g_2(r+t))$$

no puede decrecer cuando r y t aumentan, lo cual es trivial porque ambos g_1 y g_2 son no decrecientes. También tenemos que demostrar que:

$$g_2(r+t)$$

no puede decrecer cuando r y t aumentan, lo cual es también trivial porque g_1 y g_2 son no decrecientes.

□

Esto justifica la idea de duplicar ejemplos del conjunto de datos originales. De esta forma, obligaremos a dar mayor relevancia al conjunto de datos originales en el aprendizaje del segundo clasificador. Esto significa que podemos tener un nuevo $t'=k\cdot t$ y un nuevo $r'=k\cdot r$. Claro está que repitiendo el conjunto de datos originales tal vez no tengamos buenos efectos (especialmente si repetimos los datos varias veces), pero podemos tener algunas situaciones donde esto sea beneficioso.

Por supuesto hay un cuarto modelo, además de los vistos en la página 80, que asume que el primer clasificador es probabilísticamente puro y que se satisfacen las condiciones del Teorema 4, y por lo que el método mimético es útil en muchos más casos. Es fácil ver que los resultados se pueden adaptar al caso donde tenemos además el conjunto de entrenamiento. Veamos que el Lema 2 sigue cumpliéndose cuando se usan los datos de entrenamiento en la segunda etapa de la técnica mimética.

Lema 3. *Dado un escenario mimético donde el conjunto de entrenamiento es empleado en el clasificador de la segunda etapa. Si el clasificador de la primera etapa (el oráculo) es probabilísticamente puro y el clasificador de la segunda etapa es puro y leal entonces la curva de aprendizaje del caso promedio del segundo clasificador satisface la condición*:

$$si\ \ |R| \to \infty\ \ entonces\ \ \ g'_2(|R|+|T|)=1$$

Usamos g'_2 para denotar que este buen comportamiento solamente sucede con respecto al oráculo. Reservamos g_2 para la curva de aprendizaje con respecto a la función objetivo.

Demostración. Del teorema 1 y considerando que cualquier clasificador probabilísticamente puro es también probabilística-

mente puro cuando adicionamos un conjunto de casos finitos (representados por *T*).

□

Y ahora, de esto, podemos afirmar el siguiente teorema similar al teorema 4.

Teorema 6. *Dado un escenario mimético con un conjunto de entrenamiento T de tamaño t=|T|. Si el clasificador de la primera etapa es probabilísticamente puro y el clasificador de la segunda etapa es puro y leal, y la segunda etapa utiliza a ambos conjuntos R y T entonces el clasificador mimético es útil si y solo si:*

$$g_2(t) < g_1(t)$$

Demostración. Del teorema 5 tenemos

$$g_{1:2}(r,t) = \frac{r}{r+t} \cdot [g_1(t) \cdot g'_2(r,t) + (\frac{1}{c-1})(1-g_1(t)) \cdot (1-g'_2(r,t))] + (1-\frac{r}{r+t}) \cdot g'_2(r,t)$$

Para que el proceso mimético sea útil se debe cumplir

$$g_2(t) < g_{1:2}(r,t)$$

Consideremos que podemos generar tantos ejemplos aleatorios como necesitemos, entonces tenemos $|R| \to \infty$. Por lo tanto, del lema 3 tenemos:

$$g_2(t) < \frac{r}{r+t} \cdot [g_1(t) \cdot 1 + (\frac{1}{c-1})(1-g_1(t)) \cdot (1-1)] + (1-\frac{r}{r+t}) \cdot 1$$

Pero también $r \to \infty$ y por lo tanto:

$$g_2(t) < 1 \cdot [g_1(t) + (\frac{1}{c-1})(1-g_1(t)) \cdot (0)] + 0$$

Simplificando obtenemos

$$g_2(t) < g_1(t)$$

□

Obsérvese que en este caso obtenemos el mismo resultado con o sin el uso del conjunto de datos de entrenamiento, pero los resultados anteriores nos dicen que incluso en esta situación ideal, necesitaríamos menos ejemplos, en general, si usamos el conjunto de entrenamiento en la segunda etapa que si no lo usáramos.

4.4.3. Observaciones adicionales

El teorema anterior demuestra que un clasificador malo puede funcionar mejor para un conjunto de datos pequeño y se puede utilizar como un oráculo para, en segundo lugar, ver un buen clasificador que actúe bien para muestras grandes. Por ejemplo, parece que las redes neuronales junto con el aprendizaje de un árbol de decisión no es la mejor combinación (porque las redes neuronales requieren generalmente una gran cantidad de ejemplos), pero que una combinación de clasificadores + un árbol de decisión puede ser una buena combinación (puesto que los métodos de combinación evitan el sobreajuste y son buenos para muestras pequeñas).

Finalmente, hay algunos detalles que hemos dejado fuera en la discusión anterior. Primero hemos asumido que el conjunto de datos aleatorios *R* y el conjunto de entrenamiento *T* determinado no tienen ningún ejemplo en común. Esto es fácil si tenemos atributos numéricos y métodos generadores de datos aleatorios exhaustivos, pero puede no ser una suposición razonable en otros casos. Si hay ejemplos comunes, puede suceder que ambas clases sean la misma[5]. En este caso, los resultados anteriores son prácticamente idénticos a los demostrados. Por el contrario, si la clase del ejemplo *e* en *R* es diferente de la clase de *e* en *T*, entonces podemos hacer tres cosas diferentes para construir el conjunto de datos inventados:

- Corregir los errores en *R* con los de *T* y usar los buenos ejemplos.

[5] Esto es posible en el caso de que en la primera etapa el algoritmo sea leal.

- Mantener ambos conjuntos de datos (ahora tendremos inconsistencias).

- Mantener solamente la clase de *R*.

Vemos razonable considerar la primera opción como la mejor. Si realizamos esto, los resultados previos permanecen, por lo tanto esta corrección es buena, y el uso del conjunto de entrenamiento (junto con el conjunto de datos aleatorios) se refuerza, porque el primer término de la expresión en el teorema 5 se incrementará ligeramente.

Una segunda cuestión es que los ejemplos en el conjunto de datos de entrenamiento están obtenidos de una distribución a priori real, mientras que los ejemplos en *R* se generan, en el mejor de los casos, con una distribución a priori simulada. Esto significa que los ejemplos similares a los de *T* serán más probables de pertenecer a la distribución objetivo que los ejemplos similares a *R*. De nuevo, esto influye positivamente en el uso del conjunto de datos original en la segunda etapa.

4.5. Discusión y conclusiones

En este capítulo hemos analizado algunas cuestiones teóricas referentes al método mimético. En la primera parte hemos analizado las preguntas siguientes:

- Si tenemos un buen oráculo, ¿podemos aproximarnos a él con un clasificador diferente? La respuesta ha sido sí, bajo algunas suposiciones razonables. Así, si el oráculo no es fractal, el mimetizador puede ajustarse al detalle que se quiera (por ejemplo, un árbol de decisión no podado con un conjunto de datos inventados arbitrariamente grande) éste puede alcanzar el 100% de fidelidad respecto del oráculo.

- ¿Las suposiciones se sostienen para los árboles de decisión? La respuesta es sí.

En la segunda parte hemos dado respuesta a las preguntas siguientes:

- Si conocemos las precisiones del oráculo y la fidelidad del clasificador mimético, ¿puedo computar la precisión total? Hemos dado una fórmula para hacer esto.

- ¿Es útil agregar el conjunto de entrenamiento al conjunto de datos aleatorios para la segunda etapa? La respuesta es sí.

- ¿Cuáles son los casos donde el método mimético no es bueno? Eso depende del modelo, pero la única situación donde el mimético puede ser desestimado es cuando la precisión del oráculo, obtenida del entrenamiento fijado solamente, es peor que la precisión del clasificador mimético, obtenido del conjunto de entrenamiento solamente.

- Dado un conjunto de entrenamiento y las curvas de aprendizaje de dos clasificadores, ¿cuántos ejemplos al azar se necesitan para tener cierta precisión? Hemos dado una ecuación (de la definición 13) para solucionar esta clase de problemas.

- ¿Cómo deben ser el primer algoritmo de aprendizaje y el segundo algoritmo de aprendizaje en un escenario mimético? El primero debe ser rápido. El segundo debe ser altamente preciso con conjuntos de datos grandes. Un algoritmo leal y *fence-and-fill* para la segunda etapa sería una buena opción puesto que el primero es generalmente puro.

Obviamente, hay muchas otras preguntas abiertas. Algunas de ellas relacionadas a la metodología usada en este trabajo son:

- si R no es un conjunto sino un multiconjunto y tenemos ejemplos duplicados, ¿cómo afecta? Pensamos que no afecta significativamente para los resultados anteriores.

- si T no es un conjunto sino un multiconjunto y tenemos ejemplos duplicados, ¿cómo afecta? Pensamos que no afecta significativamente para los resultados anteriores.

En este capítulo hemos realizado un análisis que nos ha permitido describir teóricamente el comportamiento mimético. El siguiente paso es examinarlo experimentalmente. Los resulta-

dos experimentales darán un retrato más preciso de cómo los clasificadores miméticos trabajan y cuáles son sus límites.

Capítulo 5

Análisis con datos de entrenamiento

En este capítulo vamos estudiar con mayor profundidad y detalle el método mimético cuando además de los datos inventados se utilizan los datos de entrenamiento originales, como podemos ver en la Figura 5.1. El objetivo es mejorar el método mimético para que se pueda obtener mejor precisión y comprensibilidad de los modelos creados por esta técnica.

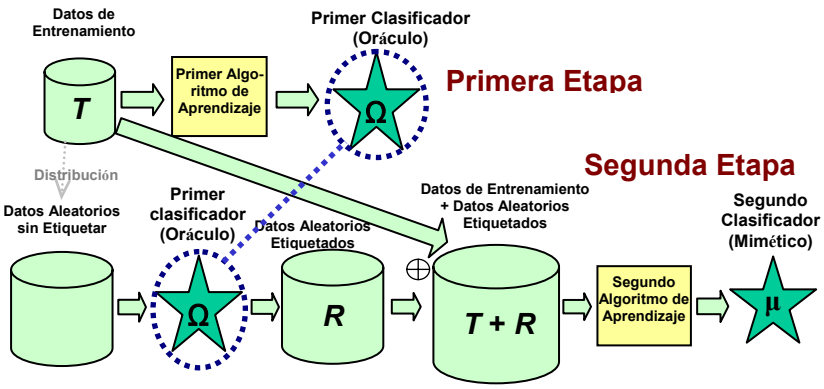


Figura 5.1 La Técnica Mimética con datos de entrenamiento.

Resumiendo lo que se ha estudiado del método mimético hasta ahora en trabajos anteriores, diremos que:

- los tamaños del conjunto de datos inventados ($|R|$) utilizados en los estudios han sido de 100, 1.000, 10.000 y 100.000 aunque no se ha estudiado detalladamente el comportamiento respecto a la dimensionalidad del problema (ni teórica ni experimentalmente).

- los datos aleatorios sólo han sido generados con la distribución a priori.

- los oráculos (Ω) que se han estudiado son *boosting* y *bagging*. Es decir, la técnica mimética sólo se ha aplicado a oráculos que son combinación de clasificadores.

- sólo se han estudiado los árboles de decisión como clasificador de la segunda etapa (μ), y obteniéndose modelos miméticos cuyos conjuntos de reglas fueron del orden de dos o más veces las generadas por un árbol de decisión creado con el algoritmo C4.5 usando para el entrenamiento únicamente los datos originales.

- se demostró experimentalmente que el uso de los datos de entrenamiento originales son beneficiosos para el aumento de la precisión del modelo mimético. Este hecho ha sido corroborado teóricamente en el capítulo anterior.

Aunque tal y como acabamos de comentar se han analizado varias cuestiones del método mimético, como la precisión del modelo y la utilidad de usar el conjunto de entrenamiento original para generar el modelo mimético, existen otros aspectos que pueden influir en el comportamiento del método que no han sido todavía estudiados:

- el método no se ha aplicado a oráculos que no sean combinación de clasificadores, por ejemplo las redes neuronales. Con esto se reforzaría la hipótesis de que el método puede aplicarse a cualquier oráculo de los que se consideran una caja negra (son no comprensibles).

- la relación entre la comprensibilidad del modelo mimético y de algunos factores como el grado de poda del algoritmo de aprendizaje utilizado para generarlo, el número de los ejemplos generados al azar y el uso de la confianza del oráculo en los ejemplos inventados no ha sido analizada.

Con la idea de ampliar nuestro conocimiento sobre el método mimético nos centramos en este capítulo en los dos aspectos antes mencionados. En primer lugar, ampliaremos las técnicas usadas para la primera etapa (generación del oráculo) considerando ahora red neuronal y ampliando el estudio del algoritmo *boosting*. El estudio de los factores mencionados en el segundo punto se hará de forma experimental: la influencia de la poda, el tamaño del conjunto de datos inventados y el nivel de confianza de los ejemplos para obtener conjuntos de reglas comprensibles sin reducir mucho la precisión del modelo.

Para este último objetivo definimos una métrica de "calidad" que permita establecer un *ranking* entre los diferentes modelos miméticos comparados. Un *ranking* es un proceso de ordenación (posicionamiento) de un modelo en una escala ordinal en relación a otros modelos, comparando su calidad con respecto a un determinado parámetro o combinación de ellos, en nuestro caso la métrica de calidad.

Los parámetros de interés que definen la métrica de calidad son: la precisión y la comprensibilidad (representada simplemente por el número de reglas del modelo mimético). Esta métrica debe tener una base adimensional y no tener una dependencia absoluta con los valores de los parámetros de interés, por lo que la normalizaremos.

La métrica de "calidad" (Q) que definimos a continuación pondera la ganancia media de precisión con el incremento de reglas del modelo mimético con respecto a un modelo de referencia. Más formalmente, definamos la ganancia en precisión como

$$GanaAcc = \frac{Acc(Mim) - Acc(Ref)}{Acc(Ref)}$$

donde "modelo de referencia" (Ref) representa a un modelo comprensible aprendido directamente con el conjunto de datos original, como por ejemplo un árbol de decisión aprendido con C4.5, Mim representa el modelo mimético (posiblemente también obtenido con el algoritmo C4.5). $Acc(M)$ es la precisión del modelo M (donde M puede ser Mim o Ref)

De forma similar definamos la ganancia respecto a las reglas como

$$GanaReglas = \frac{Reglas(Mim) - Reglas(Ref)}{Reglas(Ref)}$$

donde *Reglas*(*M*) es el número de reglas del modelo *M*.

Finalmente la métrica de calidad es

$$Q = \frac{GanaAcc}{max(GanaReglas, \psi)} \qquad (5.1)$$

$$\psi = 0.0001$$

Debido a que en el conjunto de datos de entrenamiento del modelo mimético tienen los datos originales junto con los inventados su precisión debería ser mejor que la precisión del modelo de referencia, el cual se obtiene usando sólo los datos originales suponiendo que tenemos un buen oráculo. Obviamente, si con el procedimiento mimético no tenemos mejor precisión que el de referencia, es decir, sí *Acc*(*Mim*) < *Acc*(*Ref*), entonces no es adecuado utilizar la técnica mimética, tal y como ya mostramos en el capítulo 4. El rango de posibles valores de la métrica de calidad son los números reales, pero sólo nos interesan sus valores positivos ya que un valor de calidad negativo significa que el método mimético es inútil ya que no supera la precisión del modelo de referencia. Obviamente a mayor valor (positivo) de la métrica mejor calidad del modelo. El caso en el que las reglas del modelo de referencia son mayores que las del mimético la ganancia en reglas (*GanaReglas*) también da un valor negativo, pero es una situación conveniente, aunque rara, en los experimentos realizado más adelante no hemos encontrado esta situación. Como veremos, los factores que afectan a esta métrica de calidad *Q* son múltiples y complejos.

La métrica de calidad tiene relación con el principio MML (ver sección 2.2.2 página 26) existiendo varias correspondencias. Así, la precisión en la métrica de calidad es equivalente al error en el principio MML, el cual determina el tamaño (del mensaje) de los datos dado el modelo. También se utiliza el número de reglas del modelo mimético como el divisor en la fórmula (5.1) mientras que en el principio MML es la longitud (del mensaje) del modelo. Finalmente, la métrica de calidad es un cociente entre estos parámetros y en el principio MML es una suma de términos. En realidad, la relación entre ambas medidas es lo-

garítmica y darán *rankings* similares. En este capítulo optamos por esta métrica de calidad y no el MML, porque éste es más sensible a la codificación utilizada y dificulta la inclusión del modelo de referencia.

Este capítulo está organizado como sigue. La sección 1 muestra la configuración experimental que se ha utilizado para realizar el análisis de la técnica mimética. En la sección 2 estudiamos la influencia de la poda en la calidad y la fidelidad de los clasificadores miméticos usando redes neuronales y *boosting* como oráculos. La relación entre el tamaño del conjunto de datos inventados y la calidad del modelo mimético se analiza en la sección 3. Asimismo, derivamos una función para estimar el tamaño óptimo del conjunto de datos aleatorios para cada problema. En la sección 4 se modifica el conjunto de datos inventados tomando en cuenta la confianza del oráculo al etiquetar los ejemplares. En la sección 5 se presenta un experimento que combina algunos de los resultados obtenidos en los experimentos anteriores. Finalmente, se incluye un análisis y una discusión sobre los resultados obtenidos en las secciones anteriores.

5.1. Configuración experimental

En esta sección presentamos la configuración experimental usada para el análisis del método mimético descrito en este capítulo. Para los experimentos hemos empleado 20 conjuntos de datos (véase la Tabla 5.1) del repositorio de la UCI (Black y Merz, 1998). Como puede observarse hemos intentado tener en cuenta cuantas más situaciones posibles mejor: atributos nominales y/o numéricos, conjuntos de datos pequeños y grandes (tanto en cuanto a número de atributos como de instancias), valores faltantes y varias clases. Para la generación del conjunto de datos inventados usamos la distribución a priori (para más detalle véase la sección 3.3).

Consideramos dos clases de oráculos: una red neuronal y *boosting*, usando las implementaciones de estos algoritmos en el paquete de minería de datos Weka (*MultilayerPerceptron* y *AdaBoostM1*, respectivamente). También se emplea el algoritmo J48 de Weka para construir los clasificadores de referencia y mimético. En adelante, denotamos la red neuronal como RN, el *boosting* como Boost, el clasificador de referencia como J48 y el

clasificador mimético como Mim. Todos los experimentos se realizaron usando validación cruzada de 10×10.Finalmente, cuando mostramos los resultados medios de todos los conjuntos de datos, utilizaremos la media aritmética de todos ellos.

Tabla 5.1. Información de los conjuntos de datos empleados en los experimentos

No.	Conjunto de datos	Atr.	Atr. Num.	Atr. Nom.	Clases	Tamaño	Faltan
1	anneal	38	6	32	6	898	No
2	audiology	69	0	69	24	226	Sí
3	balance-scale	4	4	0	3	625	No
4	breast-cancer	9	0	9	2	286	Sí
5	cmc	9	2	7	3	1,473	No
6	colic	22	7	15	2	368	Sí
7	diabetes	8	8	0	2	768	No
8	hayes-roth	4	0	4	3	132	No
9	hepatitis	19	6	13	2	155	Sí
10	iris	4	4	0	3	150	No
11	letter	16	16	0	26	20,000	No
12	monks1	6	0	6	2	556	No
13	monks2	6	0	6	2	601	No
14	monks3	6	0	6	2	554	No
15	mushroom	22	0	22	2	8,124	Sí
16	sick	29	7	22	2	3,772	Sí
17	vote	16	0	16	2	435	Sí
18	vowel	13	10	3	11	990	No
19	waveform-5000	40	40	0	3	5,000	No
20	zoo	17	1	16	7	101	No

5.2. Análisis de la poda

La poda es una de las primeras y más simples modificaciones que se han ideado para mejorar el comportamiento de los árboles de decisión. La poda siempre incrementa el error sobre los datos de entrenamiento (Quinlan, 1993), sin embargo, hace más pequeño al árbol de decisión y si se realiza adecuadamente puede reducir el error sobre los datos de prueba. Por lo tanto, es una de las primeras herramientas, sino la primera, que ha de estudiarse para mejorar la métrica de calidad que compara precisión y número de reglas.

En esta sección analizamos el uso de diversos grados de poda en el algoritmo Mim y cómo afectan a la métrica de calidad y a la fidelidad del clasificador mimético. Para hacer esto, realizamos varios experimentos que modifican el umbral de la confianza para podar en el algoritmo J48, consideramos los valores siguientes: 0.0001, 0.001, 0.01, 0.02, 0.05, 0.10, 0.15, 0.20,

0.25, y 0.3. Un valor de poda define el error que permitimos al seleccionar un ejemplo para el entrenamiento del modelo, es decir, cuando el valor de poda es pequeño el algoritmo reduce al máximo el conjunto de datos de entrenamiento quedándose solamente con los ejemplos que clasifica con alta confianza. En todos los casos, hemos fijado el tamaño del conjunto de datos aleatorios inventados a 10,000. Este número es suficientemente grande para que los resultados sean independientes del tamaño de la muestra.

La Figura 5.2 muestra los resultados medios de precisión obtenidos para todos los conjuntos de datos cuando el oráculo es una red neuronal (Mim RN) y cuando el oráculo es *boosting* (Mim Boost). También hemos incluido como referencia la precisión obtenida para los oráculos y por el algoritmo J48 aprendido con el conjunto de datos originales de entrenamiento y los modelos miméticos sin poda (Mim RN sin podar y Mim Boost sin podar).

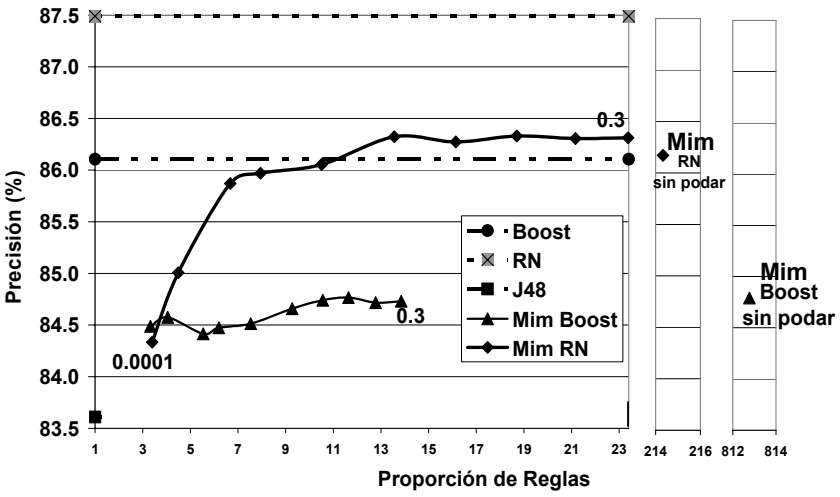


Figura 5.2. Precisión contra Proporción de Reglas de Mim para diferentes grados de poda

Como podemos ver, Mim tiene una precisión mejor cuando el oráculo es la red neuronal. También, ambos oráculos son mejores que el J48. En la Figura 5.2 podemos ver que el aumento en precisión alcanza prácticamente su máximo con un grado de poda de 0.01 con una proporción de reglas alrededor de 7

veces más que el clasificador de referencia J48. Este punto "óptimo" es corroborado por la Figura 5.3, la cual muestra la métrica de calidad para Mim RN. Obsérvese que valores altos de poda perjudica la métrica de calidad porque se aumenta el número de reglas.

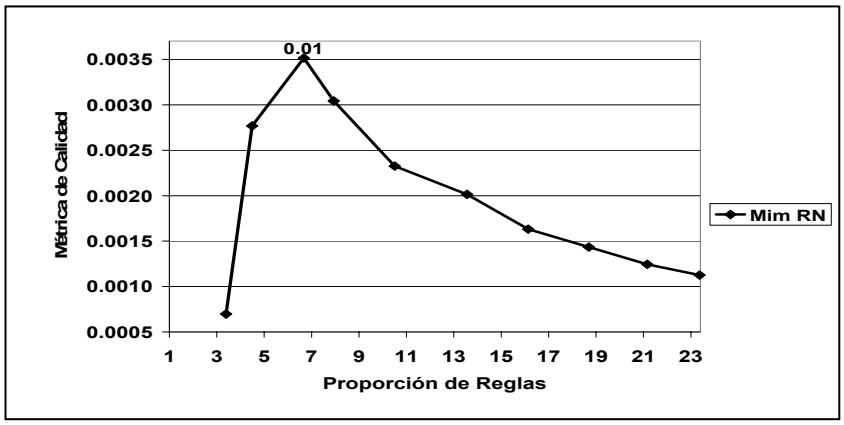


Figura 5.3. Métrica de calidad vs proporción de reglas de Mim RN para diferentes grados de poda.

En las figuras anteriores, hemos considerado los valores medios para todos los conjuntos de datos, así que los resultados muestran el comportamiento medio de los oráculos y de los clasificadores miméticos, por lo que existen conjuntos de datos para los que la precisión de la RN o el Boost es mejor que la precisión del J48 y viceversa. Sin embargo, sería interesante estudiar qué sucede con el algoritmo Mim cuando consideramos solamente aquellos conjuntos de datos para los cuales el oráculo mejora la precisión del algoritmo J48. En estos casos, el uso de un clasificador mimético en vez de un árbol simple de decisión se justifica mejor. Una forma de ver a cuál de los dos oráculos imita mejor el método mimético es comparando las coincidencias entre ambos clasificadores, es decir, analizando su fidelidad. La Figura 5.4 muestra que ahora el oráculo al que la técnica mimética se ajusta más con grados fuertes de poda es el *boosting*. Esto se debe al hecho de que es más fácil imitar un oráculo como *boosting* que se basa en la combinación del mismo algoritmo (generando patrones rectangulares con los lados paralelos a los ejes) que el clasificador mimético que imita un oráculo como la red neuronal cuyas regiones son más diferentes.

En ambos casos, no obstante, vemos que para acercarnos al número de reglas del J48 se pierde mucha fidelidad (y, por tanto, precisión).

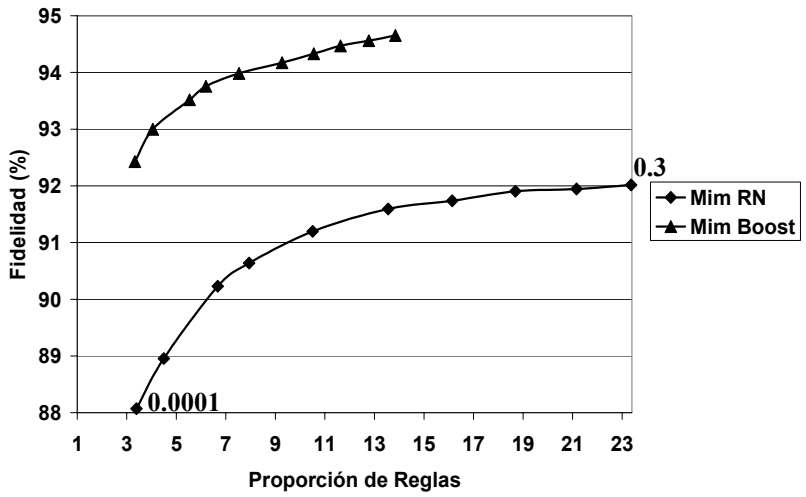


Figura 5.4. Fidelidad contra Proporción de Reglas de Mim para varios grados de poda cuando el oráculo mejora en precisión al J48

De todos los resultados anteriores, se concluye que podando (con el método de poda incluido en J48) se tiene muy poca maniobrabilidad para conseguir buenos resultados de precisión y tener pocas reglas. Las mejores calidades se obtienen con un aumento de casi 2 puntos en exactitud pero con una proporción de reglas de 7, es decir los árboles de decisión son 7 veces más grandes que el de referencia. Por lo tanto, en la sección siguiente, estudiamos la influencia de otros factores más interesantes tales como el tamaño del conjunto de datos inventados.

5.3. Análisis del tamaño de los datos inventados

Uno de los factores que más influyen en el tamaño de un árbol de decisión es el tamaño de los datos que se utilizan para entrenarlo. Los trabajos previos sobre los clasificadores miméticos (ver sección 3.1) han considerado un tamaño fijo para el conjunto de datos inventados (generalmente entre 1,000 y

10,000). Sin embargo, está claro que este valor es relativamente pequeño para conjuntos de datos tales como, por ejemplo, "*letter*" (que tiene 20,000 instancias) y relativamente grande para otros como "*hayes-roth*" (que consta de 132 instancias). En esta sección analizamos mejor la relación entre el tamaño del conjunto de datos inventados y la calidad del modelo mimético (en términos de la métrica de calidad definida al principio del capítulo).

En un análisis global de los resultados, en la Figura 5.5 mostramos la precisión promedio de los modelos miméticos con respecto al aumento del tamaño del conjunto de datos inventados. Aproximadamente hasta un tamaño de datos inventados de 3000 se observan los mayores incrementos en la precisión.

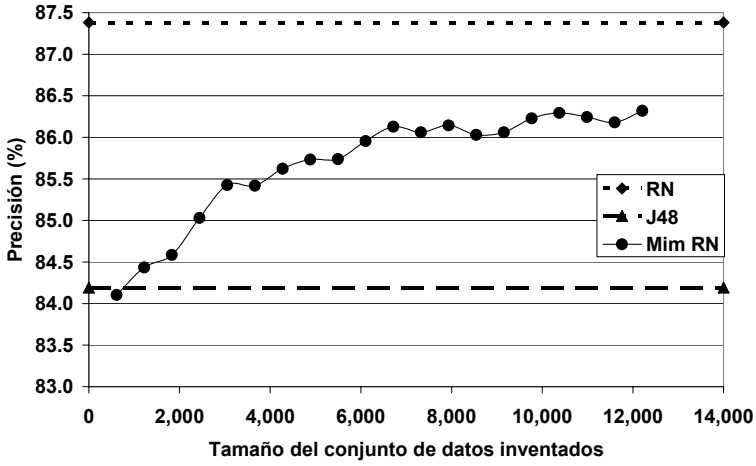


Figura 5.5. Precisión contra Tamaño del conjunto de datos inventados.

Ahora profundicemos en nuestro análisis y en vez de considerar el tamaño de los datos inventados usemos la proporción de las reglas del modelo mimético respecto a las reglas de un árbol de decisión construido con tan sólo los datos originales. En la Figura 5.6 mostramos cómo se comporta la precisión promedio con respecto a la proporción de reglas. Adicionalmente se muestran las precisiones de la red neuronal y la del árbol de decisión de referencia. El incremento en la ganancia de precisión va disminuyendo conforme aumenta la proporción de reglas. Dado que no existe una relación lineal entre la precisión y la proporción de reglas esto nos permite obtener un punto óptimo donde se maximice la métrica de calidad, como se puede ver en la Figura 5.7.

Para estudiar este factor realizamos experimentos con varios tamaños del conjunto de datos inventado, desde $0.3n$ hasta $6n$, donde n es el tamaño del conjunto de datos de entrenamiento original. Cada incremento es de 0.3, haciendo un total de 20 diferentes tamaños por cada conjunto de datos. Utilizamos la red neuronal como oráculo y J48 como clasificador mimético.

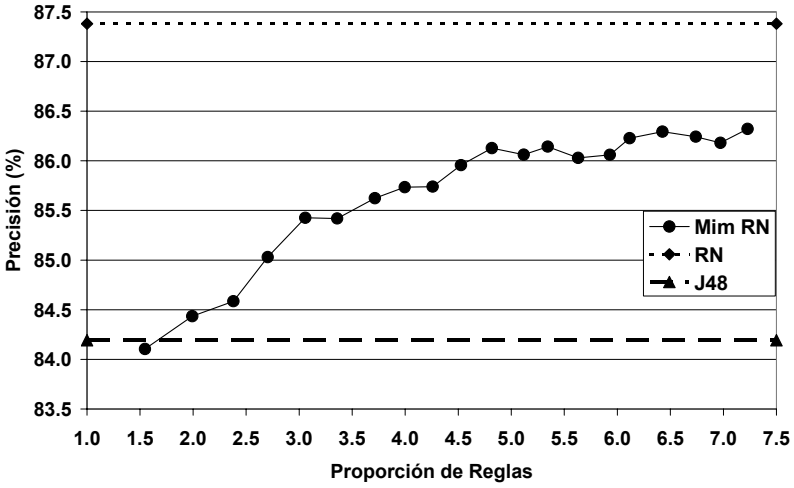


Figura 5.6. Precisión contra Proporción de Reglas de Mim para diferentes tamaños

La Figura 5.7 muestra el comportamiento del método mimético en términos de la métrica de calidad en relación al aumento del tamaño del conjunto de datos inventado. Como podemos ver, hay un punto máximo en (3.06, 0.0071), con un conjunto de datos inventados 1.5 veces más grande que el conjunto de datos de entrenamiento, lo cual significa que, como esperábamos, conjuntos de datos muy pequeños proporcionan precisiones muy bajas y número de reglas pequeño, generando valores de las métricas de calidad menores de 0.006. Por otro lado, no es beneficioso generar conjuntos de datos demasiado grandes ya que la precisión no varía mucho y sin embargo el tamaño del modelo (su número de reglas) aumenta considerablemente, y por lo tanto, la métrica de calidad empieza a descender conforme el tamaño del conjunto de datos de entrenamiento aumenta.

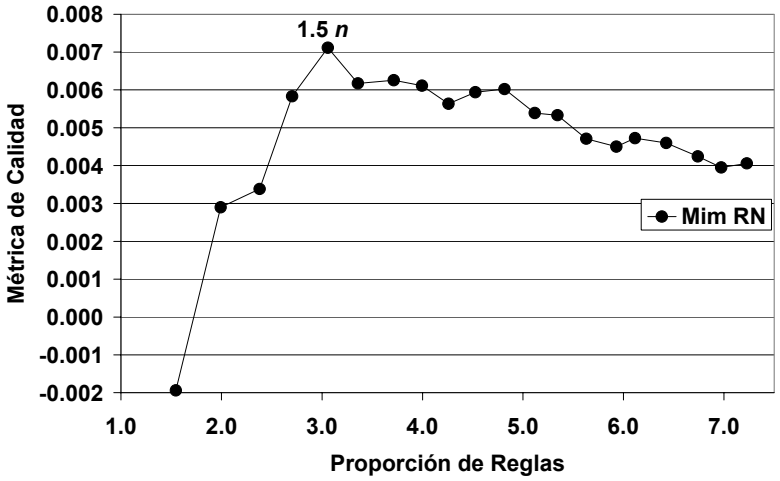


Figura 5.7. Métrica de Calidad contra Proporción de Reglas de Mim para diferentes tamaños

Teniendo esto en mente y aunque $1.5n$ pueda servir como regla general, este valor es muy dependiente del conjunto de datos. Con el fin de ajustarse más intentamos estimar el tamaño óptimo para cada conjunto de datos. Para ello, primero, con los experimentos anteriores, determinamos el tamaño del conjunto de datos inventados para el cual la métrica de calidad da el mejor valor. Con esto, tenemos un mejor valor de la métrica de calidad, siendo diferente para cada uno de los 20 conjuntos de datos. Lo siguiente es utilizar estos datos para estimar una función que calcule el tamaño del conjunto de datos aleatorio que sería óptimo para un nuevo conjunto de datos.

Primero se calcula la relación entre el tamaño del conjunto de datos inventados con el tamaño del conjunto de entrenamiento original. Llamemos a dicha relación el factor φ. Obviamente, para estimar φ sólo contamos con la información procedente del problema y plasmado en el conjunto de datos de entrenamiento, como es el número de clases, tipo de los atributos, etc.. Más formalmente, el factor φ puede estimarse con la función:

$$\varphi = f(AtrNum, AtrNom, Clases, Tamaño)$$

donde *AtrNum* es el número de atributos numéricos, *AtrNom* es el número de atributos nominales, *Clases* el número de clases y *Tamaño* el tamaño del conjunto de datos de entrenamiento original.

Debido al número pequeño de conjuntos de datos usados para esta estimación (20 conjuntos de datos) utilizamos una técnica de modelación simple: regresión lineal múltiple (con y sin coeficiente independiente), siendo la función siguiente la de mejor coeficiente de determinación (R^2 = 0.34):

$$\begin{aligned}\varphi = 0.05097 \cdot AtrNum - 0.01436 \cdot AtrNom \\ + 0.17077 \cdot Clases + 0.00003 \cdot Tamaño\end{aligned} \tag{5.2}$$

Los atributos numéricos tienen más importancia que los atributos nominales, mientras que el número de clases es también muy importante, mostrando que los grados de libertad del problema son clave para el tamaño óptimo.

Ahora, utilizamos la función (5.2) para estimar el tamaño óptimo del conjunto de datos inventados como sigue:

$$n = \varphi \cdot Tamaño \tag{5.3}$$

Para validar esta fórmula (5.3) hemos estimado los tamaños óptimos con 5 conjuntos de datos nuevos, que se muestran en la Tabla 5.2.

Tabla 5.2. Información sobre los conjuntos de datos de validación.

No.	Conjunto de datos	Atributos		Clases	Tamaño
		Numéricos	Nominales		
1	Autos	15	10	7	205
2	CarsW	0	6	4	1728
3	Credit-a	6	9	2	690
4	Heart-c	6	7	5	303
5	Hypothyroid	7	22	4	3772

La Tabla 5.3 muestra los resultados obtenidos usando los cálculos anteriores. El significado de cada columna es:

- No. es el número de identificación del conjunto de datos (coincide con la columna de igual nombre en la Tabla 5.2).
- *n max* es el tamaño cuando la métrica de calidad es máxima (real u óptima), usando diferentes tamaños sobre los datos reales.
- *n calc* es el tamaño calculado (estimado) con la fórmula (5.3).

Respecto a las columnas relacionadas con la métrica de calidad tenemos que:

- *Q n max* es la métrica de calidad máxima obtenida al experimentar con diferentes tamaños.

- *Q n calc* es la métrica de calidad estimada, obtenida con un modelo mimético entrenado con un conjunto de datos de tamaño *n calc*.

- *Q 150%* es la métrica de calidad para un tamaño fijo del conjunto de datos inventados 150% del tamaño del conjunto de datos original (es decir, 1.5*n* para todos).

De los cinco casos, en dos la técnica mimética es útil (independientemente de la técnica para estimar el tamaño). Recordemos que un valor negativo sólo puede darse por una precisión peor del mimético que del modelo de referencia. Las medias se calculan utilizando tanto los valores positivos como negativos. Por tanto, estas situaciones no satisfactorias están contempladas por la media.

En la Figura 5.8 mostramos gráficamente la métrica de calidad para estos 5 conjuntos de datos, la métrica de calidad en la gráfica utiliza la siguiente simbología: para el valor actual el prefijo ds y para el valor calculado (estimado) el prefijo dsc, seguido por el número del conjunto de datos.

Como podemos ver en la Figura 5.8 y en la Tabla 5.3, la estimación no es perfecta, y los valores estimados (*Q n calc*) están generalmente por debajo de los valores verdaderos (*Q n max*). Sin embargo, si vemos los valores medios, la calidad media obtenida usando esta estimación es perceptiblemente mejor (0.018) que la obtenida por un tamaño fijo inventado del conjunto de datos del 150% (0.01). Esto corrobora la idea de considerar un tamaño apropiado para cada conjunto de datos, dependiendo de, por lo menos, los factores anteriores (número de atributos nominales y numéricos, número de clases, y, por supuesto, del número de ejemplos del conjunto de entrenamiento).

Tabla 5.3. Tamaños y calidades estimadas y máximas para los conjuntos de datos de validación.

No.	Tamaño		Métrica de calidad (Q)		
	n max	*n calc*	*Q n max*	*Q n calc*	*Q 150%* Tamaño
1	1051	374	-0.01716	-0.03600	-0,02439
2	466	1124	0.14232	0.09200	0,05983
3	2235	372	-0.00042	-0.00335	-0,00084
4	81	324	0.07948	0.04000	0,01813
5	1018	3172	-0.00132	-0.00230	-0,00219
Media			**0.04058**	**0.01807**	**0.010108**

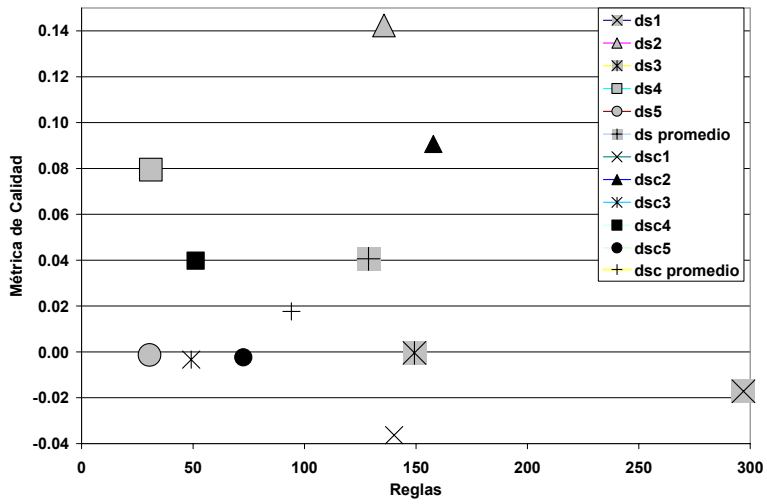


Figura 5.8. Métrica de Calidad contra Número de Reglas en los conjuntos de datos de validación

5.4. Uso de la confianza en el método mimético

Muchos clasificadores (ya sean obtenidos por técnicas de aprendizaje automático) o manualmente, en base al conocimiento de un experto, pueden acompañar sus predicciones de un valor de confianza. Este valor de confianza suele definirse como un valor real entre 0 y 1, y se suele obtener a partir de la probabilidad estimada para la clase con máxima probabilidad.

Cuando un conjunto de datos inventados se utiliza para aprender un modelo mimético, se etiqueta generalmente sin considerar la confianza del oráculo sobre este conjunto de ejemplos. Parece razonable pensar que la calidad del modelo mimético mejoraría si utilizamos solamente aquellos ejemplos del conjunto de datos inventados para los cuales el oráculo da una alta confianza. En esta sección analizamos cómo la confianza de los ejemplos del conjunto de datos inventados se puede utilizar para mejorar la técnica mimética.

Para hacer esto, una vez que hemos generado el conjunto de datos inventados R y añadido los datos de entrenamiento T, procesamos este nuevo conjunto de datos de la manera siguiente. Primero, quitamos de $R+T$ todos los ejemplares cuyo valor de confianza esté por debajo de un umbral de confianza t_c. Nótese que los ejemplos de T nunca serán eliminados porque tienen un valor de confianza de 1. Después, quitamos los ejemplos repetidos. Finalmente, duplicamos los ejemplos restantes cierto número de veces dependiendo de su valor de confianza. El conjunto de datos que resulta, que lo denotamos como D_{RT}, se utiliza para entrenar al modelo mimético. La Figura 5.9 ilustra este proceso.

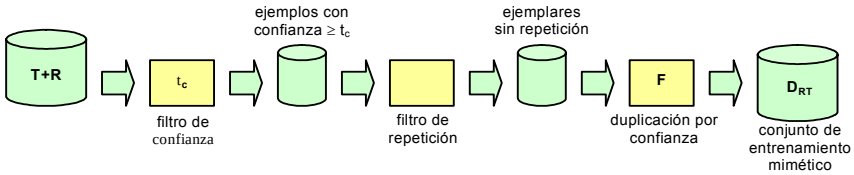


Figura 5.9. Trasformación del conjunto de entrenamiento para la técnica mimética

El número de las veces que un ejemplo debe ocurrir en D_{RT} se define como sigue: dado C_e el valor de confianza de un ejemplo e y un factor F de repetición, entonces el número de ocurrencias de e en D_{RT} es:

$$occ(e) = round(C_e \cdot F)$$

Puesto que el objetivo de la función *occ* es determinar si el ejemplo debe duplicarse o no, cada ejemplo e para el cual $occ(e)=0$ permanece en D_{RT}. Nótese que, cuando el factor de repetición aumenta, los ejemplos con alta confianza llegan a ser más significativos, y pueden ocurrir más de una vez en D_{RT}.

Para comprobar nuestra hipótesis de que el uso que hemos descrito la confianza del oráculo en los ejemplos puede mejorar el comportamiento del método mimético generando mejores modelos, hemos realizado distintos experimentos.

Los umbrales t_c de confianza considerados en esta evaluación experimental han sido: 0, 0.3, 0.9, 0.95, 0.98, 0.99 y 1.0, y para cada uno de ellos hemos utilizado un factor de repetición que varía de 1 a 4. El tamaño del conjunto de datos inventados R es 10,000. Para estos experimentos hemos utilizado todos los conjuntos de datos incluidos en la sección 1 (véase Tabla 5.1, en la página 102), una red neuronal como oráculo. En estos experimentos el valor de la confianza se obtiene de las probabilidades utilizando Laplace en la red neuronal. La Figura 5.10 muestra estos resultados. La precisión del modelo de referencia (J48), del oráculo (RN) así como la precisión del método mimético original (es decir, sin procesamiento del conjunto de datos inventado, Mim) también se han incluido como referencia.

Observamos que en el caso del umbral de confianza t_c =1.0 y del factor de repetición F=1, en general el proceso no agrega ejemplos inventados a T y el resultado es muy similar al modelo de referencia (J48). En algunos problemas, incluso conseguimos conjuntos de datos D_{RT} más pequeños que los conjuntos de datos de entrenamiento originales. La explicación de esto es que en algunos conjuntos de datos originales existen ejemplares repetidos y con el proceso de eliminación de repetidos éstos desaparecieron. Obviamente, el caso para t_c =1.0 es útil para ver si hay muchos ejemplos en R con confianza = 1 pero los resultados son muy similares al J48 original, por lo que esta combinación no la analizaremos mucho más.

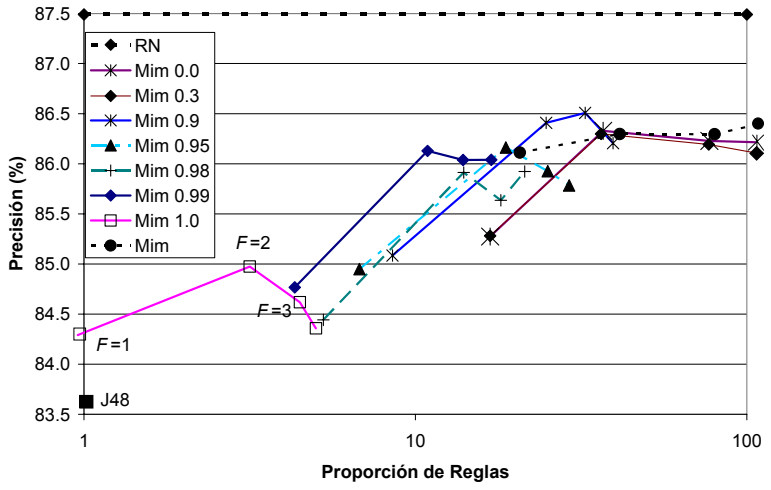


Figura 5.10. Precisión contra Proporción de Reglas del modelo mimético dependiendo de un umbral de confianza y un factor de repetición.

Para el caso de t_c=0.99 y de F=1, conseguimos un tamaño de ejemplos inventados en D_{RT} alrededor de 3,000. Si contrastamos este valor con los 7,000 ejemplos inventados cuando utilizamos t_c=0.0 (se eliminaron aproximadamente 3,000 ejemplos inventados porque se repiten), podemos ver que un porcentaje importante de ejemplos tienen una confianza igual o mayor de 0.99 para la RN (cerca del 50%). Consecuentemente, con t_c=0.99 tenemos una situación intermedia la cual está más a la izquierda de la Figura 5.10 que el Mim original. Adicionalmente, el aumento en precisión se alcanza casi totalmente con este caso (86.1 para F=2).

Con respecto al factor de repetición, el comportamiento es absolutamente similar para todos los casos, pero tiene diversas interpretaciones. Por ejemplo, para t_c=0.99 y para F=1 todos los ejemplos son incluidos una vez y para F=2 todos los ejemplos se incluyen dos veces. El aumento importante de la precisión entre estos dos casos se puede justificar por el hecho de que J48 tiene una limitación en el número mínimo de ejemplos por nodo, y esta duplicación permite que J48 sea más detallado.

A medida que se disminuye el umbral, tenemos que el número de reglas aumenta y el comportamiento es más similar (y por tanto, menos útil) que el mimético tradicional. No obstante, con un t_c=0.98 se obtiene incluso mejor precisión que el mimético tradicional.

Para confirmar estas aseveraciones mostramos en la Figura 5.11 la métrica de calidad para estos experimentos. Como podemos ver, la mejor métrica de calidad se obtiene usando un umbral de confianza de 0.99 (Mim 0.99) con un factor de repetición de 2.

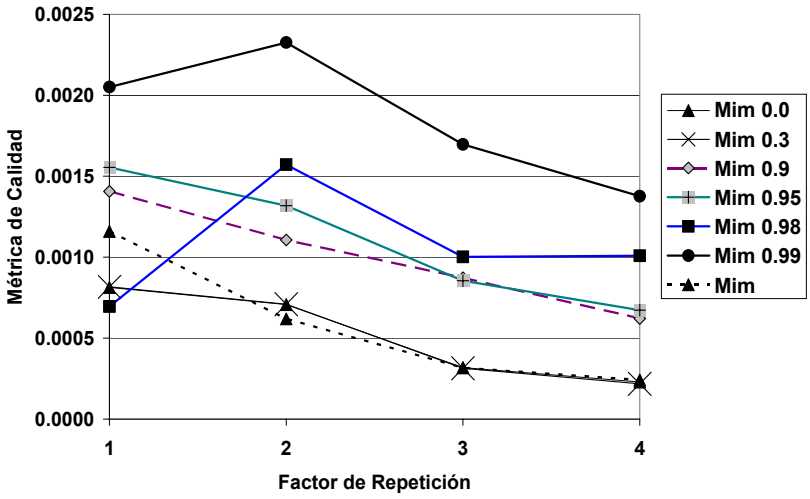


Figura 5.11. Métrica de Calidad contra Factor de Repetición para diferentes umbrales de confianza

5.5. Búsqueda del caso óptimo por combinación de factores

Debido a la complejidad del problema de encontrar un compromiso entre precisión y comprensibilidad, hemos optado hasta ahora por estudiar por separado los factores que consideramos más relevantes para conseguir nuestro objetivo. Sin embargo, estos factores no son independientes entre sí en cuanto a su efecto sobre el modelo mimético. Por tal motivo dedicamos esta sección a realizar combinaciones para primero demostrar que la combinación de factores produce una mejora en la relación entre precisión y comprensibilidad, y segundo realizar una estimación del caso óptimo.

Para cumplir con estos objetivos, realizamos una evaluación experimental combinando algunos de los resultados obtenidos

en los experimentos anteriores. Más concretamente, utilizamos los mejores valores obtenidos de forma individual para cada factor: los factores de la poda de 0.01, el tamaño del conjunto de datos inventados fue fijado al valor predicho por el modelo estimado en la sección 4, el factor de repetición se fijó a 2 y el nivel de confianza a 0.99. Además, también se realizaron experimentos variando ligeramente estos valores (para contrastar la influencia que podía tener algunos factores de forma conjunta, pocas reglas y baja precisión). Por ejemplo, también usamos un factor de poda de 0.1. La Tabla 5.4 muestra los resultados de ambos escenarios.

Tabla 5.4. Resultados experimentales obtenidos por la combinación de varios factores: poda, tamaño del conjunto de datos inventados y confianza. Prop Significa proporción de reglas y Prec es la precisión.

No.	RN	J48		Mim 0.01			Mim 0.1		
	Prec	Prec	Reglas	Prec	Reglas	Prop	Prec	Reglas	Prop
1	98.89	98.56	39.50	98.18	48.60	1.23	98.39	53.05	1.34
2	83.21	77.33	30.20	85.05	53.30	1.77	85.76	53.10	1.76
3	90.84	78.40	39.60	77.88	32.83	0.83	79.60	52.45	1.32
4	67.96	74.08	7.50	70.39	3.17	0.42	72.17	19.15	2.55
5	50.86	51.57	155.70	54.51	47.67	0.31	51.69	228.55	1.47
6	81.94	85.13	5.50	84.40	5.20	0.95	85.44	6.25	1.14
7	74.42	74.19	19.20	74.53	27.20	1.42	72.53	63.50	3.31
8	81.20	68.58	19.00	74.74	22.63	1.19	77.25	24.65	1.30
9	80.06	79.43	9.40	79.35	2.27	0.24	79.63	5.55	0.59
10	96.81	94.96	4.70	95.33	4.77	1.01	94.67	4.65	0.99
11	82.08	87.98	1,158.10	87.65	1,037.20	0.90	85.69	16,655.10	14.38
12	100.00	97.12	30.10	100.00	28.00	0.93	100.00	28.00	0.93
13	100.00	63.29	24.50	65.72	1.00	0.04	65.72	1.00	0.04
14	98.49	98.92	14.00	96.95	9.97	0.71	98.92	13.70	0.98
15	100.00	100.00	25.00	99.90	90.10	3.60	99.98	161.70	6.47
16	96.84	98.68	28.60	98.32	10.65	0.37	98.32	10.65	0.37
17	94.49	96.55	5.80	95.49	2.33	0.40	96.22	5.70	0.98
18	93.15	79.75	128.00	79.50	155.75	1.22	82.93	494.9	3.87
19	95.02	92.39	8.30	92.82	12.03	1.45	92.68	14.15	1.70
20	83.54	75.36	290.70	76.54	177.10	0.61	76.02	1450.2	4.99
Media	87.49	83.61		84.36		0.98	84.68		2.52

Los resultados con el nivel de poda con 0.01 muestran que los tres factores principales considerados (grado de poda, tamaño del conjunto de datos, nivel de confianza), si se utilizan juntos, pueden reducir drásticamente el número de reglas. De hecho, los resultados medios muestran que el número de reglas están por debajo del número de reglas del J48 con sus parámetros predeterminados, es decir, obtuvimos una proporción promedio de reglas (0.98) menor que 1. En este escenario, sin embargo, el aumento en precisión es suave (de 83.61 a 84.36). El cuadro cambia cuando vemos los resultados con el nivel de poda en 0.1. En este caso, la precisión aumenta a 84.68 con un tamaño

de los modelos que contiene 2.52 veces más reglas que el modelo original J48.

Estos resultados son perceptiblemente mejores que los obtenidos por los trabajos previos sobre los clasificadores miméticos (Domingos, 1997a, 1997b, 1998; Estruch et al, 2003). Aunque los conjuntos de datos eran diferentes y el oráculo usado era diferente también, tenemos, por ejemplo, que en (Domingos, 1998) se obtuvo un aumento de 1.5 puntos en precisión (de 75.9 a 77.4) (usando como referencia C4.5, que es, en general, peor que J48) con un aumento en el tamaño del modelo de 4.9, teniendo una calidad de 0.0051. Nótese que, con un aumento similar de la complejidad (4.68) obtuvimos 2.4 puntos del aumento de la precisión (de 83.61 a 86.01, véase la Figura 5.6) en la sección 3 con un tamaño de conjunto de datos inventados del 330% (calidad 0.006). Y acabamos de ver que podemos obtener aumentos significativos con solamente una proporción de reglas de 2.5 (calidad 0.0084).

5.6. Conclusiones

Resumiendo, del análisis por separado de los factores que afectan a la calidad de los modelos obtenidos con el método mimético (especialmente la poda), parecía que era casi imposible mejorar la métrica de calidad. La reducción del número de reglas implica sistemáticamente una reducción de la precisión y viceversa. Sin embargo, el estudio de factores tales como el tamaño del conjunto de datos inventados y de la modificación de la distribución de ejemplos (por selección y repetición) son los factores más importantes para producir mejoras significativas en precisión y comprensibilidad de los modelos miméticos.

Hemos propuesto una fórmula (5.1) normalizada para calcular la calidad de los modelos miméticos, la cual ha dado buenos resultados experimentales y a diferencia del principio MML es más fácil de aplicar.

El algoritmo base para la generación de los modelos miméticos ha sido el J48, incluido en Weka y hemos demostrado que su implementación de la poda no permite mejorar sustancialmente la métrica de calidad de los modelos miméticos.

Hemos estudiado en mayor profundidad el tamaño del conjunto de datos inventados y mostrado que no existe un tamaño constante del conjunto de datos inventados para los cuales se maximice la métrica de calidad. Hemos proporcionado una forma de determinar aproximadamente el tamaño del conjunto de datos inventados en función de la cantidad de atributos numéricos y nominales, número de clases y tamaño del conjunto de datos de entrenamiento originales que permita obtener mejores resultados en la métrica de calidad (fórmulas (5.2) y (5.3)).

Hemos demostrado que es posible encontrar un modelo óptimo que negocie precisión y comprensibilidad, al realizar una buena combinación del grado de poda, el número de los ejemplos generados al azar y el nivel de confianza de la clase predicha por el oráculo.

En este capítulo demostramos que la técnica mimética permite extraer conjuntos de reglas cortos a partir de cualquier modelo de caja negra, y usar este conjunto de reglas como la explicación de la caja negra.

Adicionalmente, este capítulo profundiza en la forma en la que los clasificadores miméticos trabajan. Una de las principales contribuciones del capítulo es la modificación de la distribución de los ejemplos, utilizando el nivel de confianza de la clase predicha por el oráculo. La segunda aportación ha sido el mostrar que se pueden obtener buenos resultados si se ajustan adecuadamente los factores que influyen en el modelo mimético.

Hasta ahora, los únicos factores que se habían tenido en cuenta eran el tamaño de los datos inventados. Hemos mostrado cómo existen otros factores que también influyen en esta técnica, más concretamente, usar poda y utilizar el nivel de confianza de la clase predicha por el oráculo para modificar la distribución de los ejemplos.

Capítulo 6

Análisis sin datos de entrenamiento

En muchas aplicaciones reales se utilizan modelos del entorno que rara vez están acompañados de los datos de soporte o de entrenamiento a partir de los cuales se derivaron. Esto se puede deber a muchas razones. El modelo pudo haber sido construido por un experto humano sin aplicar ninguna técnica de aprendizaje computacional. También puede ser que los datos usados para construir el modelo simplemente ya no existan por pérdida o daño de los mismos o, más probablemente, porque ha pasado bastante tiempo. Ésta es una situación real en muchas áreas tales como la ingeniería, la diagnosis, los procesos de fabricación, los negocios, la medicina, etc.

Por tanto, las técnicas de aprendizaje automático y de minería de datos, aunque cada vez más útiles y frecuentes en la adquisición de conocimiento a partir de datos, no se pueden aplicar si tenemos ya un modelo que queremos adaptar o transformar, pero no disponemos de los datos originales.

Más aún, incluso en algunas aplicaciones el uso de técnicas automáticas para obtener conocimiento plantea algunos inconvenientes.

Una de estas áreas de aplicación es la construcción y, en especial, el mantenimiento de los sistemas expertos.

Como ya se dijo en el capítulo 1, uno de los problemas principales en los sistemas expertos es el cuello de botella en la adquisición del conocimiento, debido principalmente a que muchos expertos no pueden expresar su conocimiento en forma de reglas claras e inequívocas. Los expertos emplean generalmente reglas explícitas y reglas empíricas (sin ningún fundamento lógico). Incluso en caso de que el experto pueda escribir todo su conocimiento, esto representa un alto esfuerzo, puede ser un proceso donde se desperdicia mucho tiempo y es difícil de mantener. Además, el resultado es a veces un modelo trascrito que no se puede aplicar de una forma completamente automatizada ya que todavía presenta una cierta ambigüedad.

En algunos ámbitos (por ejemplo, el diagnóstico) el problema puede plantearse en forma de casos descritos por una serie fija de atributos (de cualquiera tipo: nominal o numérico) y por un valor dependiente (también de tipo: nominal o numérico). Los modelos creados por los expertos predicen el valor dependiente de acuerdo con el resto de los atributos. Esta estructura de modelo es similar a la de los modelos predictivos en aprendizaje computacional, donde tenemos casos expresados a partir de varias variables de entrada y una variable de salida.

En este caso, el problema de la adquisición del conocimiento puede ser resuelto por técnicas de aprendizaje computacional. La idea es que preguntemos a un experto para obtener los datos a partir de los cuales construir nuestro modelo. Existen varios métodos para hacer esto, y a esta clase de aprendizaje se le llama "*query learning*" (Angluin, 1987), donde el experto actúa como un oráculo.

Sin embargo, la aproximación "*query learning*", en general, tiene algunas desventajas: no muchas técnicas de aprendizaje computacional están diseñadas para aprender realizando preguntas, en algunos casos las preguntas pueden ser complejas, y, la más importante, muchos paradigmas de *query learning* asumen que el experto está presente en el entrenamiento, y da las respuestas inmediatamente. Si no, los algoritmos de aprendizaje tendrían que esperar horas o días, lo cual haría que el problema de la disponibilidad del experto fuera todavía peor. Adicionalmente, en los análisis existentes de *query learning*, no está presente el análisis de la complejidad del modelo.

En este capítulo investigamos el uso del método mimético para la adquisición del conocimiento. En este caso, no existen unos

datos iniciales de entrenamiento (tal y como hemos estado suponiendo hasta ahora), sino que se usan datos sin etiquetar para los cuales el experto debe proveer el valor de salida. El resultado es un conjunto de datos ya etiquetados, el cual puede ser empleado para entrenar el modelo mimético usando cualquier método disponible de aprendizaje computacional o paquete de minería de datos (ver Figura 6.1). En este caso, el objetivo es que el modelo mimético se aproxime al comportamiento del experto.

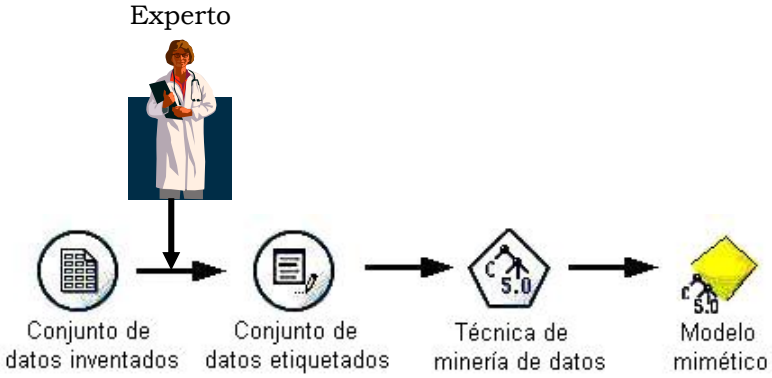


Figura 6.1. Proceso mimético con un experto como oráculo

De acuerdo a lo presentado en los capítulos anteriores, el modelo mimético será tanto más preciso cuanto más grande sea el número de ejemplos que podemos pedir al experto que nos etiquete.

Sin embargo, no existe una técnica de adquisición de conocimiento donde podamos tener una disponibilidad infinita del experto. Por lo tanto, tenemos que analizar cuántos casos requerimos para conseguir un modelo que capture el conocimiento del experto con una alta precisión y comprensibilidad, pero con un número mínimo de preguntas.

También hemos visto que, a diferencia de la precisión, la comprensibilidad del modelo mimético diminuye al aumentar el número de casos, ya que cuantos más casos más complejo es el modelo. Una forma de solucionar este dilema es utilizar la poda (para aquellas técnicas que dispongan de esta opción), pero esto requeriría generar muchos más casos de los que realmente necesitamos, saturando a nuestro experto inútilmente. En lugar de ello mostraremos que se puede controlar la comprensibilidad de nuestros modelos miméticos (en términos

del número de reglas) estimando el número de casos que pedimos etiquetar al experto, es decir, iremos pidiendo al experto que etiquete conjuntos de datos y en base a ellos estimaremos la cantidad de casos necesarios para lograr maximizar a la vez precisión y comprensibilidad, sin exceder la cantidad de casos necesarios para lograr este objetivo.

Para ello, ideamos un método basado en curvas de aprendizaje que permiten predecir el número de casos que serán requeridos tomando en cuenta el compromiso entre precisión y comprensibilidad de los modelos creados.

En resumen, en este capítulo proponemos el uso de la técnica mimética de una forma diferente a la del capítulo anterior, asumiendo que el oráculo es un experto humano. Entonces, este experto etiqueta el conjunto de datos inventados, que será el único conjunto de datos empleado para entrenar al modelo mimético, es decir, no contamos con los datos originales.

El capítulo está organizado como sigue. En las secciones 1 y 2, introducimos una aproximación basada en el principio de longitud mínima de mensaje (MML) para determinar el tamaño óptimo del conjunto de datos inventados. La sección 3 muestra la configuración experimental que se usa en la sección 4 para la evaluación experimental de nuestra aproximación. La sección 5 presenta cómo utilizar esta aproximación en la práctica para un problema dado de adquisición de conocimiento. Finalmente, se describen las conclusiones de este estudio.

6.1. Análisis del modelo mimético desde la perspectiva del principio MML

Ya hemos mostrado en el capítulo anterior que los siguientes factores del modelo mimético están relacionados: el tamaño del conjunto de datos inventados empleado para entrenar el modelo mimético, su número de reglas y su precisión. Así pues, la precisión aumenta cuando el tamaño de los datos inventados aumenta, mientras que se obtienen pocas reglas (y así, mayor comprensibilidad del modelo) usando conjuntos de datos inventados más pequeños. De estos resultados se deduce que es necesario encontrar un compromiso entre la precisión y el número de reglas (comprensibilidad) en el modelo mimético. El factor que es más conveniente para realizar este estudio es el

tamaño del conjunto de datos inventados, puesto que de él dependen la precisión y el número de reglas.

Dado que pretendemos aplicar la técnica mimética para la adquisición del conocimiento, nuestro oráculo es un experto y por lo tanto la generación del modelo mimético puede abordarse como un problema típico del aprendizaje a partir de datos. Una manera muy común de analizar la relación entre el tamaño del modelo y el nivel de error es el principio de la longitud mínima de mensaje (MML, ver sección 2.2.2 de la página 26). Nosotros vamos a aplicar la filosofía de este principio para analizar el modelo mimético, más concretamente, para determinar el tamaño óptimo del conjunto de datos inventados que maximiza la precisión y la comprensibilidad del modelo mimético.

Dado que en la situación que planteamos el aprendizaje del modelo requiere de un etiquetado previo de los ejemplos por parte de un experto, modificamos el principio MML enunciado en la subsección 2.2.1 para tener en cuenta el coste del experto. Así, dado un modelo M aprendido mediante la técnica mimética usando un conjunto de datos inventados R etiquetados por un experto humano, definimos el coste de M como:

$$Cost(M) = MsgLen(M) + MsgLen(R|M) + Query(R) \qquad (6.1)$$

Nótese que, la ecuación anterior sólo difiere de la ecuación (2.1) -página 28- en el término *Query*(R) que representa el coste de preguntar al experto para etiquetar los ejemplos del conjunto R. Esta modificación es necesaria para tener en cuenta que la disponibilidad del experto es limitada. Si recordamos, $MsgLen(M)$ es el tamaño del modelo (que puede ser expresado en términos del número de reglas del mismo); $MsgLen(R|M)$ son los datos que no puede clasificar bien el modelo M, es decir los errores del mismo (cuyo número está inversamente relacionado con la precisión). En consecuencia, el problema de determinar un tamaño óptimo para R se puede ver como un proceso de optimización cuyo objetivo sea maximizar la precisión (representada por $MsgLen(R|M)$) y la comprensibilidad (relacionada con $MsgLen(M)$) del modelo dadas algunas restricciones (el número de preguntas) que también se incluyen en la ecuación. Como veremos después, estas restricciones están conectadas con el resto de parámetros a través de las curvas de aprendizaje del modelo mimético, que representan la variabilidad del

número reglas y los errores del modelo con respecto al tamaño del conjunto de datos inventados.

6.2. Optimización del tamaño de R usando el principio MML modificado

En esta sección presentamos nuestra aproximación basada en la modificación del principio MML incluida en la sección anterior para determinar el tamaño óptimo del conjunto de datos inventados usado para aprender el modelo mimético. Para optimizar dicha fórmula hemos de obtener una relación entre las distintas variables involucradas. Para ello, en primer lugar, deduciremos las expresiones que relacionan la precisión (a través del error) y la comprensibilidad (a través del número de reglas) con el tamaño del conjunto de datos inventados usando las curvas de aprendizaje para varios conjuntos de datos del repositorio UCI.

A continuación, usaremos dichas expresiones en la ecuación (6.1).

6.2.1. Curvas de aprendizaje para modelos miméticos

Tal y como se explicó en el capítulo 2 y 4, las curvas de aprendizaje de un modelo representan la relación entre los factores de interés. Para los modelos miméticos, las curvas de aprendizaje de interés muestran la relación entre el número de reglas (*NR*) y el porcentaje de error (*Error*) con respecto al tamaño del conjunto de datos (*S*). Para obtenerlas, hemos inducido cien modelos miméticos (para cada conjunto de datos del repositorio UCI incluidos en la Tabla 6.1) variando el tamaño *S* desde un tamaño inicial que coincide con el tamaño del conjunto de datos de entrenamiento en el UCI, y aumentando éste un 5% cada vez. Las Figuras 6.2 y 6.3 muestran, como ejemplo, las curvas obtenidas para el conjunto de datos *balance-scale*.

En general, la curva de aprendizaje *Error vs. Tamaño* puede describirse mediante la ecuación

$$Error = \alpha \cdot \ln(S) + \beta \qquad (6.2)$$

donde α y β son constantes (que determinaremos por una regresión lineal con escala logarítmica en el tamaño). Como vemos en la Figura 6.2, el ajuste de esta función al comportamiento de la curva real es casi perfecto.

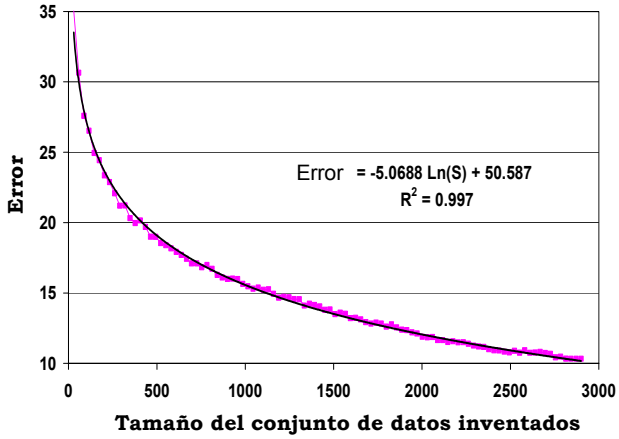


Figura 6.2. Error vs. Tamaño para el conjunto de datos *balance-scale*

Análogamente, la curva de aprendizaje *Número de Reglas vs. Tamaño* (Oates y Jensen, 1997) puede describirse como

$$NR = \delta \cdot S + \lambda \tag{6.3}$$

donde δ y λ son constantes (también determinadas por regresión lineal). Ambas funciones las utilizaremos en la siguiente subsección para obtener el tamaño óptimo de *R*.

La Figura 6.3, reafirma la clara relación lineal que existe entre el número de reglas del modelo y el tamaño del conjunto de datos inventados, como hemos supuesto en la ecuación (6.3).

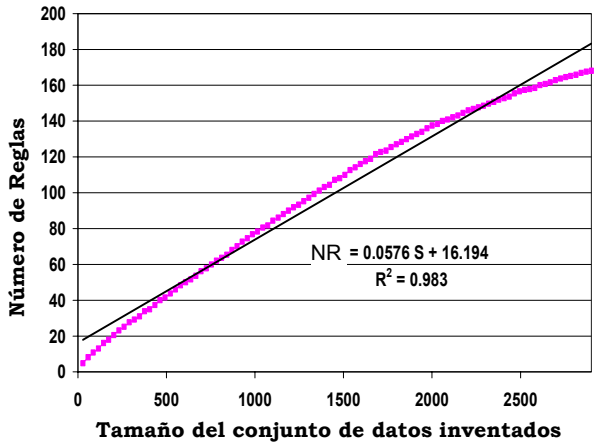


Figura 6.3. Número de reglas vs. Tamaño para el conjunto de datos *balance-scale*

En la sección 4 presentaremos los detalles de las curvas de aprendizaje, en particular los valores de los parámetros α, β, δ y λ estimados para cada conjunto de datos.

6.2.2. Cálculo del tamaño óptimo del conjunto de datos inventados

Ahora relacionaremos las ecuaciones de las curvas de aprendizaje con la función objetivo. Después, calcularemos los valores críticos (máximo y mínimo) de dicha función. Un valor mínimo de la función objetivo será el tamaño óptimo del conjunto de datos inventados.

Considerando que un modelo M consta de NR reglas y que el coste de codificar cada regla es cr, el coste de codificar M puede ser aproximado por:

$$MsgLen(M) \approx NR \cdot cr \tag{6.4}$$

Asimismo, si existen $Error$ errores y el coste de codificar cada uno de ellos es ce, entonces la longitud del mensaje del conjunto de datos R dado M se puede también aproximar por:

$$MsgLen(R|M) \approx Error \cdot ce \tag{6.5}$$

Finalmente, el coste de preguntar al experto es

$$Query(R) \approx S \cdot cq \tag{6.6}$$

cq es el coste de etiquetar un ejemplo.

Remplazando (6.4), (6.5) y (6.6) en (6.1) tenemos que:

$$Cost(M) \approx NR \cdot cr + Error \cdot ce + S \cdot cq \tag{6.7}$$

Incluyendo (6.2) y (6.3) en (6.7) se tiene

$$Cost(M) \approx (\delta \cdot S + \lambda) \cdot cr + (\alpha \cdot Ln(S) + \beta) \cdot ce + S \cdot cq$$

Finalmente tenemos

$$Cost(M) \approx (\delta \cdot cr + cq) \cdot S + \lambda \cdot cr + \alpha \cdot ce \cdot Ln(S) + \beta \cdot ce$$

Ahora, para obtener sus puntos críticos determinamos la primera derivada y la igualamos a cero. La solución a esta ecuación nos permite calcula el tamaño óptimo.

La primera derivada respecto a S es

$$\frac{d(Cost(Modelo))}{dS} = \delta \cdot cr + cq + \frac{\alpha \cdot ce}{S}$$

La ordenada del punto crítico es

$$S_{opt} = -\frac{\alpha \cdot ce}{\delta \cdot cr + cq} \tag{6.8}$$

Para saber si este punto es un valor máximo o un mínimo se requiere el signo de la curvatura que se obtiene a través de la segunda derivada, la cual es

$$\frac{d^2(Cost(Modelo))}{dS^2} = -\frac{\alpha \cdot ce}{S^2}$$

El signo de esta expresión depende del signo de α dado que *ce* y *S* son siempre positivos. Si α es negativo entonces el signo de la doble derivada es positivo, esto significa que hemos encontrado un valor mínimo, es decir, tenemos el punto donde disminuimos el coste de representación del modelo. Si el signo de α es positivo esto nos indica que es un punto máximo y no interesa para nuestro problema. El signo positivo de α nos estaría indicando que al ir aumentando el número de datos de entrenamiento la precisión del modelo obtenido tendería a bajar y esto sería una anomalía. Por lo tanto queda demostrado que sí es posible estimar con la ecuación (6.8) el tamaño óptimo del conjunto de datos de entrenamiento en un modelo mimético.

Ahora, estamos listos para utilizar estos cálculos en aplicaciones prácticas (como veremos en la sección 5). Así, para un problema dado, generamos modelos miméticos usando conjuntos de datos inventados incrementando su tamaño cada vez. Entonces, aproximamos las curvas de aprendizaje y computamos sus parámetros. Finalmente, aplicamos la fórmula (6.8) para obtener el tamaño óptimo del conjunto de datos inventados.

6.3. Evaluación experimental

En primer lugar, presentamos la configuración utilizada para la evaluación experimental de nuestra aproximación. En vez de usar a expertos verdaderos, utilizamos los conjuntos de datos del repositorio UCI, de los cuales aprendemos una red neuronal que actúe como nuestro "experto". Hemos elegido este escenario porque la red neuronal puede capturar patrones complejos y es absolutamente diferente de la técnica que utilizamos para el modelo mimético, un árbol de decisión.

6.3.1. Configuración experimental

Para cubrir diferentes situaciones y clases de problemas, usaremos 18 conjuntos de datos (mostrados en la Tabla 6.1) del repositorio UCI (Black y Merz, 1998).

Para la generación del conjunto de datos inventados utilizamos la distribución uniforme.

Tabla 6.1. Conjuntos de datos usados en los experimentos

No.	Conjunto de datos	Atr. Num.	Atr. Nom.	Clases	Tamaño
1	anneal	6	32	6	898
2	audiology	0	69	24	226
3	balance-scale	4	0	3	625
4	breast-cancer	0	9	2	286
5	cmc	2	7	3	1,473
6	colic	7	15	2	368
7	diabetes	8	0	2	768
8	hayes-roth	0	4	3	132
9	hepatitis	6	13	2	155
10	iris	4	0	3	150
11	monks1	0	6	2	556
12	monks2	0	6	2	601
13	monks3	0	6	2	554
14	sick	7	22	2	3,772
15	vote	0	16	2	435
16	vowel	10	3	11	990
17	waveform-5000	40	0	3	5,000
18	zoo	1	16	7	101

La red neuronal, que "actúa" como experto, es el método *MultilayerPerceptron* incluido en el paquete de minería de datos Weka (Witten y Frank, 2005), con los parámetros por defecto. Los clasificadores miméticos se construyen con el algoritmo J48 (incluido también en Weka) con su configuración predeterminada. Finalmente, cuando mostramos los resultados promedios de muchos conjuntos de datos, utilizaremos la media aritmética de todos los conjuntos de datos. Para las precisiones y el número de las reglas mostradas en todos los experimentos, utilizamos la validación cruzada de diez repeticiones con diez pliegues.

6.3.2. Experimentos

Dada la configuración experimental anterior, en esta sección demostramos cómo podemos estimar confiablemente las curvas de aprendizaje con tan sólo tres modelos. Esto permitirá pedir al experto el conjunto de datos a etiquetar, no más y no menos que lo necesario, y sin estar interactuando con el algoritmo.

En particular, deseamos estimar los parámetros α y δ presentes en la ecuación (6.8). En la sección 6.2.1 hemos indicado que las curvas de aprendizaje se obtienen a partir de los datos correspondientes a cien modelos miméticos. Sin embargo, desde el punto de vista práctico en esta sección demostraremos que es suficiente con generar tres modelos para establecer los parámetros de las curvas. Concretamente, usaremos tamaños para el conjunto de datos inventados del 5%, 150% y de 450% del tamaño del conjunto original de datos.

La Tabla 6.2 muestra los parámetros α y δ estimados usando cien modelos para cada uno de los 18 conjuntos de datos. También mostramos los coeficientes de determinación.

Tabla 6.2. Parámetros y coeficientes de determinación para las curvas de aprendizaje miméticas creada con cien puntos (n=100).

No. Conjunto de Datos	n=100			
	Error vs Tamaño		Reglas vs Tamaño	
	α	R^2	δ	R^2
1	-4.0964	0.90	0.0723	0.99
2	-5.9717	0.95	0.1726	0.99
3	-5.0688	1.00	0.0576	0.98
4	0.1917	0.09	0.0939	1.00
5	-0.7527	0.61	0.0933	1.00
6	-1.1518	0.58	0.0082	0.97
7	-0.8269	0.79	0.0436	1.00
8	-6.1926	0.82	0.0583	0.93
9	-2.2169	0.85	0.0436	0.99
10	-4.9763	0.82	0.0468	0.99
11	-5.5677	0.77	0.0022	0.16
12	-12.035	0.91	0.1055	0.92
13	-1.4338	0.42	0.0021	0.67
14	-0.2049	0.78	0.0064	0.99
15	-0.9203	0.81	0.0255	0.99
16	-4.4329	0.99	0.2139	1.00
17	-2.6815	0.98	0.1089	1.00
18	-8.8899	0.95	0.0955	0.97
Media		0.82		0.91

El valor promedio del coeficiente de determinación es 0.82 para el *Error* vs. *Tamaño* y 0.91 para *Reglas* vs. *Tamaño*, lo cual significa que el 82% de la variabilidad del error es explicada por el tamaño y el 94% de la variabilidad del número de reglas es explicado también por el tamaño.

La Tabla 6.3 muestra los parámetros α y δ para los 18 conjuntos de datos, estimados usando solamente tres modelos para cada uno. También mostramos los coeficientes de determinación. El valor promedio de este coeficiente es 0.97 para el *Error* vs. *Tamaño* y 0.94 para *Reglas* vs. *Tamaño*, lo cual significa que el 97% de la variabilidad del error es explicada por el tamaño y el 94% de la variabilidad del número de reglas es explicada también por el tamaño.

Tabla 6.3 Parámetros y coeficientes de determinación para las curvas de aprendizaje miméticas creadas con tres puntos (n = 3)

No.	n=3			
	Error vs Tamaño		Reglas vs Tamaño	
	α	R^2	δ	R^2
1	-4.5711	0.97	0.0776	0.98
2	-7.9902	0.99	0.1808	0.99
3	-5.4376	1.00	0.0591	0.99
4	-0.4968	0.74	0.0949	1.00
5	-1.5936	0.93	0.0956	1.00
6	-2.66	0.95	0.0097	0.96
7	-1.2611	0.98	0.0445	1.00
8	-7.197	0.96	0.0699	0.93
9	-3.628	0.98	0.0456	0.99
10	-8.9288	0.97	0.0498	0.98
11	-7.6452	0.95	0.0081	0.49
12	-7.5818	0.95	0.1045	0.97
13	-4.1454	0.94	0.0043	0.73
14	-0.2817	1.00	0.0068	0.99
15	-1.5628	0.99	0.0267	0.99
16	-4.9703	1.00	0.2144	1.00
17	-3.1991	0.99	0.1099	1.00
18	-10.865	0.99	0.1083	0.99
Media		0.97		0.94

A simple vista parece que las diferencias no son significativas. Para confirmar este resultado y mostrar que podemos estimar una curva a partir de solamente tres modelos de tamaños diferentes, comparamos el coste del modelo usando la ecuación (6.7) y los parámetros estimados para 3 modelos con el mejor

coste posible (se elige el coste más bajo obtenido de los 100 modelos).

La Tabla 6.4 muestra los costes mínimos estimados con tan sólo tres modelos miméticos (empleando un valor que recomendamos de 10 en el factor $ce/(cr+cq/\delta)$), y el mejor coste de entre 100 modelos miméticos. Con estos costes realizaremos una prueba t-*Student* pareada.

Tabla 6.4 Costes mínimos estimados para 3 modelos y el observado de entre 100 modelos.

No.	Coste mínimo	
	Estimado para 3 modelos	El mejor de 100 modelos
1	213.15	202.922
2	525.72	481.648
3	224.54	227.275
4	326.45	307.864
5	568.83	543.152
6	172.27	179.361
7	298.85	289.335
8	191.32	218.889
9	226.46	225.327
10	23.83	90.752
12	250.30	297.48
14	67.79	61.349
15	85.04	74.915
16	810.62	782.523
17	443.81	427.34
18	142.27	158.163
Media	285.70	285.52

Para calcular el valor t de *Student* necesitamos: la media de las diferencias d es -0.19, y su desviación típica s_d es 28.43 en las 18 parejas. El error estándar de la media de las diferencias es:

$$es_d = \frac{28.43}{\sqrt{18-1}} = 6.9$$

Por lo que el valor de t será ahora

$$t = \frac{d}{es_d} = \frac{-0.19}{6.9} = -0.028$$

El valor de t en tablas para 17 grados de libertad y un nivel de confianza de 0.01 es de 2.947. Como la t calculada es menor que la t en tablas se demuestra que no hay diferencia significativa entre la estimación del coste con tres modelos miméticos y el coste mínimo de 100 modelos miméticos.

6.4. Procedimiento de aplicación

Una vez que hemos demostrado que tres tamaños son suficientes para obtener un punto óptimo casi idéntico al examinando con 100 tamaños, apliquemos este resultado general a nuestro problema real: reducir al mínimo el número de entrevistas al experto sin exceder el número de los casos requeridos que se etiquetarán.

Como nuestro objetivo es estimar el número de ejemplares a etiquetar por un experto sin exceder del número necesario de ejemplares que maximicen precisión y comprensibilidad no podemos pedir que nos etiquete una cantidad grande de ejemplares. El proceso es, pedirle que etiquete pocos ejemplares (por ejemplo 20) y con éstos crear 2 modelos para luego calcular los parámetros de las curvas de aprendizaje. Proponiendo ciertos costes unitarios y empleando la ecuación (6.8) se estima el número de ejemplares. Entraremos en un ciclo repitiendo el etiquetar la cantidad de ejemplares estimados y construyendo un nuevo modelo que aunado con el resto de los modelos construidos anteriormente permitan realizar una estimación más precisa del tamaño óptimo.

La descripción algorítmica de este procedimiento lo tenemos en la Figura 6.4.

El algoritmo comienza pidiendo 20 ejemplos al experto (se aprenden dos modelos, uno usando sólo 10 ejemplos y el otro con los 20 ejemplos). Con esto computamos una primera curva, de la cual podemos estimar el valor óptimo de *S* (Sopt, esto se hace en la función Estima_Optimo()). Si el tamaño estimado es más pequeño que el número de los ejemplos pedidos al experto (situación extraña) o a la desviación entre el valor estimado y el número de ejemplos es pequeño entonces terminamos. Si no, pedimos al experto los casos restantes hasta el tamaño óptimo estimado.

```
tamaños= {10, 20}; // conjunto con número inicial de ejemplos
margen= 0.1; // porcentaje del error con respecto al tamaño óptimo.
i= 20;
MIENTRAS ( verdad ) {
  Pregunta_Experto_Hasta(i);
  Sopt= Estima_Optimo(tamaños);
   SI ((Sopt < i) o (i/Sopt > 1 - margen))
     ALTO;
   SINO {
     i= Sopt;
     tamaños= tamaños ∪ { i }; }
}
```

Figura 6.4. Procedimiento para estimar el tamaño óptimo

Un ejemplo de traza del algoritmo anterior para el conjunto de datos "*zoo*" es como sigue:

Iteración	*i*	*opt*	*i/opt*
1	20	207	0.1
2	207	340	0.6
3	340	353	0.97 (ALTO)

Como podemos ver, ejecutamos solamente tres iteraciones (tres entrevistas al experto) y sin exceder el valor óptimo (en este caso es finalmente alrededor de 350), podemos encontrar el punto óptimo para el problema de adquisición del conocimiento. Resultados similares se obtienen para otros conjuntos de datos.

6.5. Conclusiones

En este capítulo hemos analizado un escenario donde la adquisición del conocimiento se hace con preguntas simples (varios casos de salida) a uno o más expertos. Hemos obtenido una aproximación práctica, fácil de implementar y general en muchas situaciones. Es posible aplicarla especialmente en diagnosis y en los problemas donde los casos están bien estructurados (definidos por medio de una tabla de atributos/casos) y además el modelo del experto no es fácilmente explicable con una docena de reglas.

En estas situaciones, la técnica mimética es útil y fácil de emplear, puesto que lo único que necesitamos es al experto, unos ciertos datos sin etiqueta (que se pueden generar con una simple distribución uniforme) y cualquier técnica de aprendizaje computacional. Como sabemos, cuanto mayor es el conjunto de datos que el experto puede etiquetar, mejores serán los resultados. Esto genera un dilema entre la calidad del modelo que captura el conocimiento del experto, el tamaño del modelo (y por lo tanto, su comprensibilidad) y el coste/disponibilidad del experto. Usando curvas de aprendizaje para varios conjuntos de datos, hemos visto que el coste del experto (tamaño del conjunto de datos) y el tamaño del modelo se correlacionan linealmente (por lo menos para los árboles de la decisión), y por lo tanto podemos simplificar nuestro análisis basado en el principio MML a solo dos factores: la precisión y el tamaño del modelo/datos.

Dados estos resultados generales hemos ideado una metodología para estimar el número de los casos necesarios para obtener el modelo "óptimo", en términos del compromiso entre la precisión, el número de preguntas y el tamaño del modelo. Hemos visto que, generalmente, podemos aproximar este valor de forma razonable con sólo tres iteraciones, lo que significa que pediremos algunos casos al experto en una primera entrevista, y en la segunda o tercera entrevista sabremos casi exactamente el número de casos requeridos.

Los resultados y la metodología propuesta son un paso adelante en la adquisición de conocimiento a través del aprendizaje computacional de forma mucho más práctica y fácil. Este paso puede ayudar a solucionar el cuello de botella de la adquisición de conocimiento.

Capítulo 7

Contextualización con revisión

En un contexto inicial un modelo se debe validar muy a fondo antes de ser aplicado o empleado en un cierto dominio. Con el paso del tiempo puede ocurrir que el modelo empiece a comportarse mal (por ejemplo, su precisión puede disminuir considerablemente) porque se ha producido un cambio del contexto de aplicación, siendo éste diferente del contexto inicial. Esta situación es habitual y entra dentro del problema de mantenimiento de modelos o teorías. La aproximación más natural cuando un modelo ya no se ajusta al contexto actual es la revisión del mismo para que considere la nueva situación. La revisión de teorías, como vimos en el capítulo 2, puede realizarse de muchas maneras, desde la modificación puntual intentando minimizar los cambios hasta revisiones más drásticas con el fin de adaptarse mejor. La característica común es que todas las técnicas intentan preservar o aprovechar el modelo existente ya que parece lógico que, en general, reutilizar un modelo existente es más inteligente que generar un modelo nuevo desde cero.

En este capítulo proponemos reutilizar el modelo original[1] de una manera diferente basándonos en la técnica mimética. En concreto, utilizaremos el modelo inicial como oráculo para generar un conjunto de datos amplio y grande que represente la situación original. Entonces combinaremos este conjunto de

---

[1] Utilizamos los siguientes adjetivos de forma sinónima: inicial, original, antiguo y viejo para calificar las siguientes palabras: modelo, dato, situación, contexto y región.

datos con los nuevos datos originados de la nueva situación. Para balancear la importancia de la situación original y la nueva utilizamos sobremuestreo. Entonces, con el conjunto de datos resultante de la unión, aprendemos un modelo que considere la situación original (de la cual tenemos mucho más detalle) y la nueva situación (de la cual tenemos generalmente menos detalle). En este capítulo, analizamos por qué el modelo mimético puede ser mejor que el modelo antiguo, mejor que un modelo nuevo construido con sólo los nuevos datos y mejor que la combinación de ambos.

Nuestra aproximación trata una situación frecuente que se presenta cuando tenemos modelos antiguos o sistemas expertos trabajando, y deseamos ponerlos al día sin usar técnicas de ingeniería inversa o técnicas incrementales específicas de revisión de modelos. Además, supondremos que no tenemos los datos iniciales (si han existido), y que los nuevos datos etiquetados son escasos. Nótese que este escenario es bastante habitual en la realidad. De hecho, si los nuevos datos cubrieran todo el problema y fueran lo suficientemente numerosos sería más sencillo reentrenar el modelo con los datos nuevos. Asimismo, remarcamos que nuestra configuración no requiere ninguna técnica de aprendizaje particular o incremental, no necesita desarrollar o cambiar partes del viejo modelo y la revisión se puede automatizar con un paquete de minería de datos o de aprendizaje automático. Obviamente, para aplicar la técnica las dos condiciones generales para la revisión de modelos deben existir: el modelo antiguo se comporta peor que inicialmente al evaluarlo con los nuevos datos (por lo que la revisión es necesaria) y la cantidad de nuevos datos no es grande o no es suficiente para obtener un buen modelo nuevo (la revisión es apropiada).

Por lo que conocemos, la técnica mimética no se ha utilizado para la revisión de modelos ni para lo que se conoce como "*concept drift*" (véase la sección 2.4.3). La técnica mimética hace posible la revisión de modelos sin virtualmente ninguna suposición: no hay necesidad de los datos antiguos, tampoco tenemos que saber su distribución original, cualquier técnica de aprendizaje computacional puede ser utilizada, y no se requiere ningún tamaño mínimo para los nuevos datos (por el contrario, la técnica es especialmente beneficiosa cuando hay pocos datos disponibles).

Este capítulo está organizado como sigue. En la sección 1 enumeramos las diferentes formas de revisar un modelo usando técnicas del aprendizaje computacional. En la sección 2 presentamos un escenario donde la nueva situación consiste en nuevos datos en una región particular que no era inicialmente significativa o que simplemente era inexistente para el modelo original. Utilizamos este escenario para capturar la idea de la "revisión accionada por novedad" y analizamos cuándo la técnica mimética podría ser aplicable. En la sección 3 realizamos una evaluación experimental que demuestra que el modelo mimético supera, en general, al antiguo modelo y al nuevo modelo. Luego, en la sección 4, discutimos estos resultados desde una perspectiva más general. Finalmente, presentamos las conclusiones del capítulo.

7.1. Revisión de modelos a través de aprendizaje

En general, podemos utilizar el aprendizaje (no incremental) para la revisión de modelos de las siguientes formas (véase la Figura 7.1):

a) Entrenar un nuevo modelo con los nuevos datos. En este caso, el modelo original es substituido por el nuevo modelo. La debilidad principal de esta aproximación es que se requiere tener a mano una gran cantidad de nuevos datos para que el nuevo modelo sea más preciso que el modelo original.

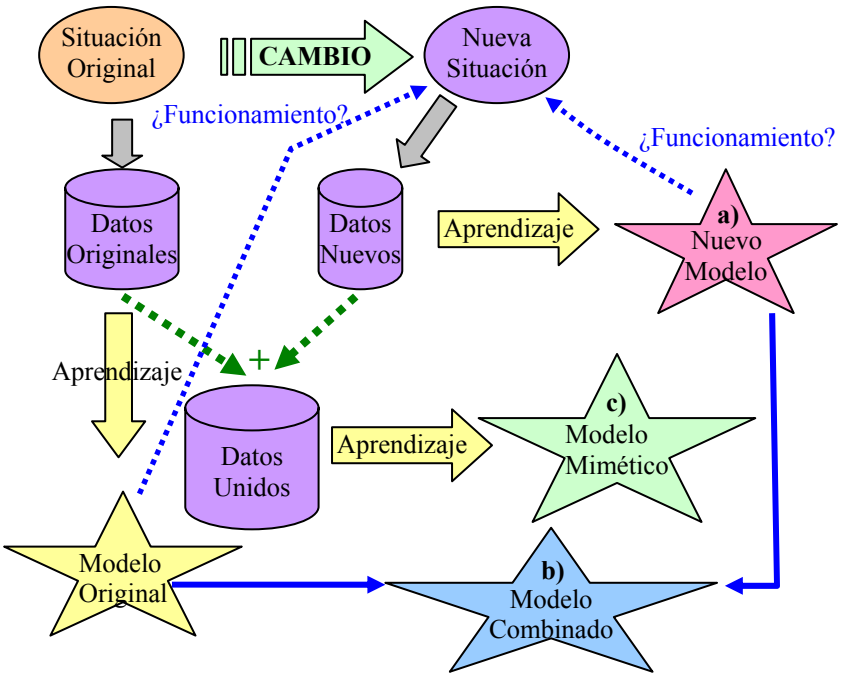


Figura 7.1 Modelos alternativos en un escenario de cambio.

b) Entrenar un nuevo modelo con los nuevos datos (opción a) y combinarlo con el modelo antiguo. Aunque, en general, la precisión se incrementa en un modelo combinado (en relación con la precisión de sus modelos componentes), este resultado se ha establecido cuando hay una gran cantidad de modelos simples diferentes, pero es incierto qué sucede con la combinación de solamente dos modelos. Otra desventaja de esta opción es que un modelo combinado sería incomprensible, lo cual puede ser un serio inconveniente para varios dominios de aplicación. Una forma alternativa a la combinación de una colección de modelos es construir un modelo simple (modelo híbrido) tomando piezas de ellos. En este caso, el modelo híbrido puede ser comprensible (por ejemplo, si son árboles de decisión). Sin embargo, esta aproximación es menos popular que los métodos de combinación de clasificadores porque, en general, para obtener modelo híbridos con mejor funcionamiento que sus componentes es necesaria más información de los clasificadores (Blockeel y Struyf, 2001) o del problema (Bridwell et al., 2005) para seleccionar las piezas en la construcción del modelo híbrido.

c) Entrenar un nuevo modelo con la fusión de los datos originales y nuevos. Esta opción requiere que los datos originales y los nuevos estén disponibles. En este caso procedemos a fusionar los datos iniciales y los nuevos, para así generar un nuevo conjunto de datos para después entrenar un modelo global.

Resumiendo, cada una de las alternativas expuestas tiene inconvenientes: la opción a) requiere disponer de una buena cantidad de datos de la situación nueva, la opción b) requiere información adicional si se quiere que el modelo obtenido sea comprensible y, finalmente, la opción c) necesita disponer de los datos originales (situación original).

En las secciones siguientes presentamos una nueva variante de la opción c) que cubre la situación en la cual los datos de entrenamiento del modelo original no están disponibles a la hora de la revisión del modelo. Es en este caso que proponemos utilizar la técnica mimética puesto que no necesita de los datos originales (como se ha mostrado en el capítulo 6). En este caso, el conjunto de datos inventados son etiquetados por el modelo original substituyendo al conjunto de datos original faltante. Este conjunto, junto con los datos nuevos, se usa para generar el modelo mimético. Asimismo, comparamos los resultados obtenidos por nuestra propuesta con los modelos generados por las otras aproximaciones.

7.2. Escenario con novedad

En esta sección presentamos un escenario común a la revisión de modelos donde los cambios ocurren sobre todo en las áreas que no eran muy importantes en la situación original. Este escenario es frecuente siempre que un sistema tenga que "cubrir" o "tratar" una nueva área, y generalmente se llama revisión accionada por la novedad.

Diferentes ejemplos de esta situación se pueden encontrar especialmente en los sistemas expertos, los sistemas basados en conocimiento o los modelos creados por humanos que tienen que ser revisados debido a una nueva situación (un nuevo mercado, un nuevo producto, un nuevo problema, una nueva región, etc.). Para poner un ejemplo específico, nos referimos a un sistema experto desarrollado para diagnosticar parásitos y

enfermedades en la cosecha del arroz y para sugerir medidas de prevención/curación (Plant y Stone, 1991). El conocimiento fue adquirido de científicos de la universidad agrícola de Andhra Pradesh, dirección de investigación del arroz, en la India. Uno de los problemas de este sistema basado en el conocimiento es que para cubrir nuevas situaciones, debe constantemente pedir a los científicos las nuevas reglas para los parásitos y las enfermedades para los nuevos tipos de cosechas. Algunos científicos que participaron en la situación anterior ya no participan en situaciones subsecuentes. Así pues, aquí vemos casos donde no solamente no tenemos los datos viejos (el modelo fue hecho por los expertos humanos) sino que también ya no tenemos a algunos de los expertos humanos. En (Plant y Stone, 1991), se reconoce que cualquier manera de automatizar la revisión y la actualización, o de utilizar una cierta clase de ingeniería inversa para mantener los modelos sería muy útil para los sistemas del conocimiento, no solamente en agricultura, sino en muchas otras áreas.

Para analizar este escenario de una manera más general abstraemos del mismo dos situaciones: una situación original y una situación final o futura. El momento en el cual la necesidad del cambio se acciona sucede en un cierto punto entre las dos, como se muestra en la Figura 7.2.

Como discutimos en la sección 7.1, la idea es comparar el modelo mimético con el nuevo modelo, el antiguo modelo y su combinación. Para hacer esto simulamos el escenario accionado por novedad como sigue. Dado un conjunto de datos D, identificamos aleatoriamente un subconjunto de ejemplos de D, que denotamos como D_C, que pertenecen a una región arbitraria. Esta región se define generando una condición C sobre un atributo. Si el atributo seleccionado x es nominal entonces las condiciones son de la forma $x = v$, donde el valor v (aleatoriamente elegido) pertenece al dominio de x. Similarmente, las condiciones sobre atributos numéricos son de la forma $x <= v$ o $x > v$, donde el valor v pertenece al intervalo $[x_{min}, x_{max}]$ siendo x_{min} y x_{max} los valores mínimo y máximo del atributo x en D. Ambos, el valor y el atributo, se determinan aleatoriamente. Ahora, la condición C permite definir el conjunto de datos originales O quitando de D los ejemplos que cumplen C, es decir, $O = D - D_C$. Por lo tanto, O no cubre un área que sí lo hace D (la cual esta definida por D_C). Análogamente, para construir el conjunto de datos nuevos, seleccionamos la misma proporción de ejemplos de O y de D_C (utilizamos sobremuestreo en caso de

necesitarlo). La condición C se elige de tal forma que el tamaño de D_C esté entre el 20% y el 30% del tamaño de D. Esto hace que generalmente la condición C involucre a un sólo atributo. En la sección 3, aplicaremos este procedimiento para obtener el escenario accionado por novedad en la revisión de modelos a varios conjuntos de datos del repositorio UCI (Black y Merz, 1998).

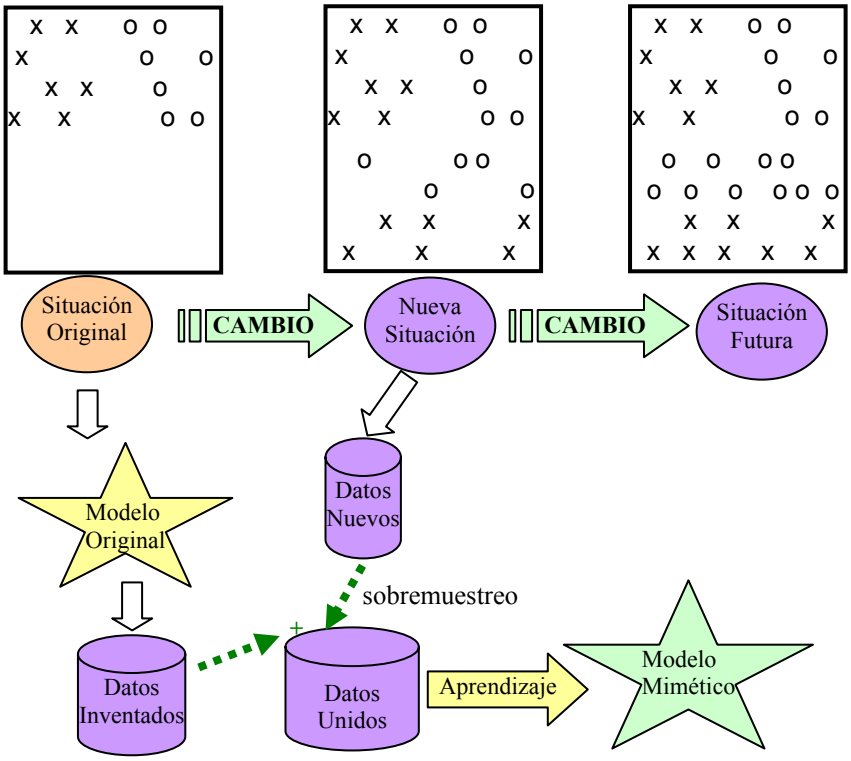


Figura 7.2. Revisión mimética en un escenario accionado por novedad

Los cuatro modelos que consideraremos se obtienen de la siguiente manera:

Modelo Antiguo: M_{Old}. Es el modelo original entrenado con los datos existentes en la situación original (el conjunto de datos original O).

Nuevo Modelo: M_{New}. Es el nuevo modelo entrenado con pocos datos en la nueva situación (T_{New}).

Modelo Combinado: M_{Comb}. Es el modelo obtenido por la combinación de M_{New} y M_{Old}. Esta combinación se hace utilizando sus probabilidades estimadas como peso en la combinación. Esta técnica de voto mayoritario consiste en entrenar a cada clasificador individualmente. Entonces, si ambos clasificadores divergen para un ejemplo, la clase con mayor peso (mayor probabilidad de los dos clasificadores M_{New} y M_{Old}) es la que se emite como resultado final de la combinación. Llamaremos T_{New} al conjunto de datos nuevos antes de efectuar el sobremuestreo.

Modelo Mimético: M_{Mim}. Se obtiene usando el siguiente conjunto de entrenamiento T_{Joint}: primero generamos un conjunto de datos inventados (T_{Inv}) suficientemente grande y etiquetado por el modelo original (M_{Old}), y entonces le añadimos el conjunto de datos nuevos (usando diversos grados de sobremuestreo para balancear ambas partes).

En el último caso, T_{Inv} se genera con una distribución uniforme. No podemos utilizar ninguna distribución basada en los datos porque partimos de la hipótesis que no tenemos los datos originales y que los nuevos datos son escasos y obtenidos de una nueva distribución. Así pues, en lo que sigue, asumimos que sabemos solamente los valores posibles para los atributos nominales y el rango (mínimo–máximo) para los atributos numéricos.

En adelante, denotamos el modelo viejo evaluado con datos viejos (originales) como $M_{Old}(Old)$, el modelo viejo evaluado con nuevos datos como $M_{Old}(New)$ o simplemente M_{Old}, el nuevo modelo evaluado con nuevos datos como $M_{New}(New)$ o simplemente M_{New}, el modelo combinado evaluado con nuevos datos como $M_{Comb}(New)$ o simplemente M_{Comb} y el modelo mimético evaluado con nuevos datos $M_{Mim}(New)$ o simplemente M_{Mim}. Es decir, si no se indica con qué datos se evalúa el modelo se presupone que es con datos nuevos.

Una cuestión importante para el aprendizaje del modelo mimético es determinar la proporción de ejemplos que tenemos que utilizar de la vieja situación y de la nueva situación. Podemos conseguir una idea aproximada de esta proporción razonando de la siguiente forma. Consideremos el caso en el cual la vieja situación y la nueva situación son totalmente independientes. En este escenario, el viejo modelo se comportará como un modelo al azar para los nuevos datos. Denotamos este modelo

como M_{Random}, y su comportamiento con los nuevos datos es justamente $M_{Random}(New) = 1/nC$, donde nC es el número de clases en el problema[2]. Así pues, si sabemos la precisión obtenida por el modelo para los datos viejos, es decir $M_{Old}(Old)$ y el "mal" comportamiento para los nuevos datos, es decir $M_{Old}(New)$, podemos estimar la proporción α de ejemplos que vienen de la vieja situación y de la nueva situación como sigue:

$$M_{Old}(New) = \alpha M_{Old}(Old) + (1-\alpha) M_{Random}(New)$$

$$\alpha = \frac{M_{Old}(New) - M_{Random}(New)}{M_{Old}(Old) - M_{Random}(New)} \tag{7.1}$$

Una vez que hemos obtenido α podemos pensar que obviamente la nueva situación y la vieja situación no son en general totalmente independientes. En general, son positivamente dependientes, así que el viejo modelo se comportará mejor que el modelo al azar. Consecuentemente, tenemos que tomar este valor de α como un mínimo, pero aún así es una buena aproximación de la proporción de ejemplos inventados y de ejemplos que vienen de la nueva situación, porque la tendencia es a que en el futuro la nueva situación cobre más importancia.

Así pues, en principio generamos el conjunto de datos inventados tan grande como sea posible (puesto que esto es bueno para la técnica mimética) y después utilizamos sobremuestreo (por repetición de ejemplos) con los nuevos datos para aproximar la proporción de ejemplos necesarios al valor de α estimado. Por ejemplo, si tenemos α = 0.8, y generamos 2000 ejemplos inventados, y tenemos solamente 100 nuevos ejemplos, entonces tenemos que sobremuestrear los nuevos ejemplos 5 veces, para conseguir una proporción de 0.8 (2000 contra 500). Formalmente:

$$\alpha = \frac{|T_{Inv}|}{|T_{Inv}| + fs \cdot |T_{New}|} \tag{7.2}$$

donde fs es el factor de sobremuestreo.

---

[2] Esto es solamente cierto si en ambos casos las clases están equilibradas. En caso contario, el comportamiento sería peor, aunque si las clases están desequilibradas de forma semejante (definición 8 en el capítulo 4, página 66) el comportamiento sería mejor.

7.3. Evaluación experimental

En esta sección presentamos algunos experimentos para analizar el uso de la técnica mimética para la revisión de modelos en el escenario de novedad, que acabamos de describir en la sección anterior. Asimismo, compararemos el modelo mimético generado con los modelos viejo (M_{Old}), nuevo (M_{New}) y combinado (M_{Comb}).

7.3.1. Configuración experimental

Para los experimentos hemos empleado 10 conjuntos de datos (véase la Tabla 7.1) del repositorio UCI (Black y Merz, 1998). Como ya habíamos dicho antes, para la generación del conjunto de datos inventados utilizamos la distribución uniforme, porque tenemos que considerar el caso donde no existen los datos originales y tampoco la distribución original.

Tabla 7.1. Información de los conjuntos de datos utilizados en los experimentos

No.	Conjunto de Datos	Atributos		Clases	Tamaño	Datos Faltantes
		Num.	Nom.			
1	anneal	6	32	6	898	No
2	audiology	0	69	24	226	Sí
3	balance-scale	4	0	3	625	No
4	breast-cancer	0	9	2	286	Sí
5	colic	7	15	2	368	Sí
6	diabetes	8	0	2	768	No
7	hepatitis	6	13	2	155	Sí
8	iris	4	0	3	150	No
9	vowel	10	3	11	990	No
10	zoo	1	16	7	101	No

Todos los modelos son clasificadores construidos con el algoritmo J48 incluido en Weka (Witten y Frank, 2005). Para los modelos miméticos usamos diferentes grados de sobremuestreo: M_{Mim1} cuando no usamos sobremuestreo, M_{Mim2} con un factor de sobremuestreo = 2, y M_{Mim5} con un factor de sobremuestreo = 5. Para todos los experimentos usamos una validación cruzada de 10 pliegues con 500 repeticiones. Con este número de repeticiones logramos disminuir la varianza en los resultados de los experimentos. Las pruebas t de *Student* se

realizan en el nivel de confianza del 5 en los 5000 (500×10) modelos de cada tipo, usando la corrección propuesta por (Bouckaert y Frank, 2004). Finalmente, cuando mostramos los resultados promedios, utilizamos la media aritmética.

7.3.2. Evaluación empírica

Tal como hemos descrito en la sección 2, generamos aleatoriamente una condición para cada repetición y conjunto de datos tal que cubra alrededor del 20% al 30% del conjunto de datos. Con esto construimos el conjunto de datos original *O*, que tendrá lógicamente alrededor del 70-80% del conjunto de datos *D* de la UCI. La mayoría de (pero no todo) este conjunto de datos se utiliza para entrenar al modelo antiguo y se llama T_{Old}. No utilizamos el *O* entero, puesto que reservamos una parte para construir T_{New}, con lo que evitamos tener ejemplos comunes en ambos conjuntos de datos. Finalmente, el nuevo conjunto de datos T_{New} se construye tal que su tamaño es el 10% del tamaño de T_{Old}. El contenido de T_{New} se establece de la siguiente forma: la mitad es una muestra escogida al azar de T_{Old} (como hemos dicho antes) y la otra mitad viene de D_C, es decir de los ejemplos del conjunto de datos que cumplen la condición *C*. Asimismo, y tal como mostraremos en detalle en esta misma sección, los conjunto de validación Old y New se obtienen a partir de T_{Old} y T_{New}.

La primera cosa a analizar con este escenario es si la revisión del modelo es necesaria o no. Puesto que ha aparecido una nueva área (en una extensión de entre un 20% a un 30%) y ésta tiene la mitad del peso en T_{New}, se espera que la precisión de $M_{Old}(New)$ sea perceptiblemente menor que la de $M_{Old}(Old)$. Como podemos ver en las columnas segunda ($M_{Old}(Old)$) y tercera ($M_{Old}(New)$) de la Tabla 7.2, la precisión del modelo original con los nuevos ejemplos baja perceptiblemente en general. Por lo tanto, la revisión puede ser detectada y por tal razón la revisión del modelo es requerida.

Tabla 7.2. Precisión del modelo antiguo empleando los datos originales y nuevos, así como el modelo nuevo con los datos nuevos.

Conjunto de datos	$M_{Old}(Old)$	$M_{Old}(New)$	M_{New}
anneal	98.52	89.23	90.78
audiology	78.57	61.31	53.39
balance-scale	78.32	75.86	72.43
breast-cancer	75.11	68.75	67.70
colic	85.15	84.60	79.64
diabetes	73.90	71.46	68.97
hepatitis	83.44	76.00	76.55
iris	93.44	93.38	81.67
vowel	78.50	57.39	46.27
zoo	93.75	66.20	59.84
Media	***83.87***	***74.42***	***69.72***

La última columna de la Tabla 7.2 muestra que un modelo nuevo entrenado con los datos nuevos todavía se comporta peor en general que el modelo original, debido principalmente a que el número de datos nuevos es muy bajo como para obtener un buen modelo. Asimismo, la precisión del modelo original sobre los nuevos datos es inferior (en casi 9.5 puntos) a la precisión de este modelo con los datos viejos, lo que justifica la necesidad de la revisión del modelo y confirma que la forma de construir el escenario de revisión que hemos descrito en la sección anterior (determinación de la zona de novedad a partir de una condición aleatoria) es adecuada para la situación que estamos tratando.

Ahora, el siguiente modelo a considerar es el modelo mimético. Para ello el conjunto de datos inventados se une con T_{New} para formar un conjunto de datos T_{Joint}, el cual se utiliza para entrenar al modelo mimético. Con el propósito de poder comprobar que el valor estimado de α determina el mejor sobremuestreo variamos el porcentaje de los ejemplos que vienen de T_{Inv} y de T_{New} usando varios grados de sobremuestreo previamente predeterminados: 1, 2 y 5, dando tres modelos miméticos diferentes denotados, como hemos dicho antes por Mim1, Mim2 y Mim5 respectivamente. La Tabla 7.3 muestra el conjunto de datos de entrenamiento y de prueba usado en cada modelo. El conjunto de prueba para los modelos miméticos lo constituyen un 10 % de los datos nuevos y ninguno de los datos de T_{Inv} debido a que estos últimos son inventados a diferencia de los de T_{New} que sí corresponden a los datos que esperamos en un futuro.

Finalmente, el tamaño del conjunto de datos inventados en la técnica mimética es un factor que debe ser analizado antes de poder hacer una comparación completa con todos los modelos posibles. Como hemos demostrado en capítulos anteriores, el tamaño del conjunto de datos inventados es una cuestión clave para el éxito del modelo mimético. Generalmente, la recomendación es hacer este conjunto de datos tan grande como sea posible.

Tabla 7.3. Configuración del entrenamiento y validación de los modelos miméticos.

Validación	Modelo	Entrenamiento	Prueba
$M_{Old}(Old)$	M_{Old}	90% T_{Old}	10% T_{Old}
$M_{Old}(New)$	M_{Old}	90% T_{Old}	T_{New}
M_{New}	M_{New}	90% T_{New}	10% T_{New}
M_{Mim1}	M_{Mim1}	T_{Inv} + 90% T_{New}	10% T_{New}
M_{Mim2}	M_{Mim2}	T_{Inv} + dos veces 90% T_{New}	10% T_{New}
M_{Mim5}	M_{Mim5}	T_{Inv} + cinco veces 90% T_{New}	10% T_{New}
M_{Comb}	M_{Comb}	-	10% T_{New}

Para analizar la importancia de este tamaño, hemos construido T_{Inv} de diversos tamaños, 1000, 2000, 3000, 4000 y 5000. No aplicamos la obtención del tamaño óptimo del capítulo anterior pues aquí nuestro objetivo es solamente obtener la mejor precisión. Hemos observado que, aunque estos diversos tamaños cambian la proporción de ejemplos que vienen de la vieja situación y de la nueva situación (y esto es tanto para M_{Mim1}, como para M_{Mim2} y M_{Mim5}), en general los resultados son completamente estables a los diversos tamaños de T_{Inv}. Una explicación a este comportamiento es que los tamaños de los conjuntos de datos son, en general, pequeños (ver Tabla 7.1) y con estos tamaños de datos inventados se llega a una zona de estabilidad de la precisión.

Podemos ver el comportamiento estable en tres casos: un caso donde el modelo mimético sobrepasa al antiguo y al nuevo modelo, un caso donde incluso sobrepasa al modelo antiguo con los datos antiguos, y un caso donde el modelo mimético M_{Mim5} no logra sobrepasar en precisión al nuevo modelo.

El primer caso, donde son mejores los modelos miméticos que el antiguo y nuevo modelo se muestra en la Figura 7.3.

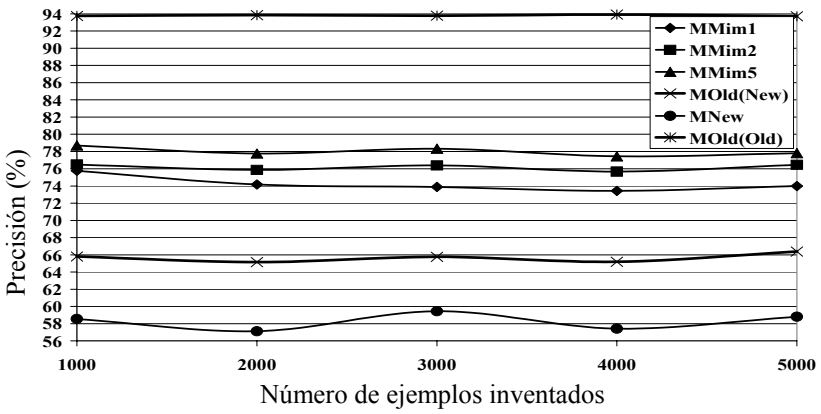


Figura 7.3. Comportamiento dependiendo de $|T_{Inv}|$ para el conjunto de datos *zoo*.

El segundo caso donde incluso los modelos miméticos sobrepasan al modelo antiguo con los datos originales se muestra en la Figura 7.4.

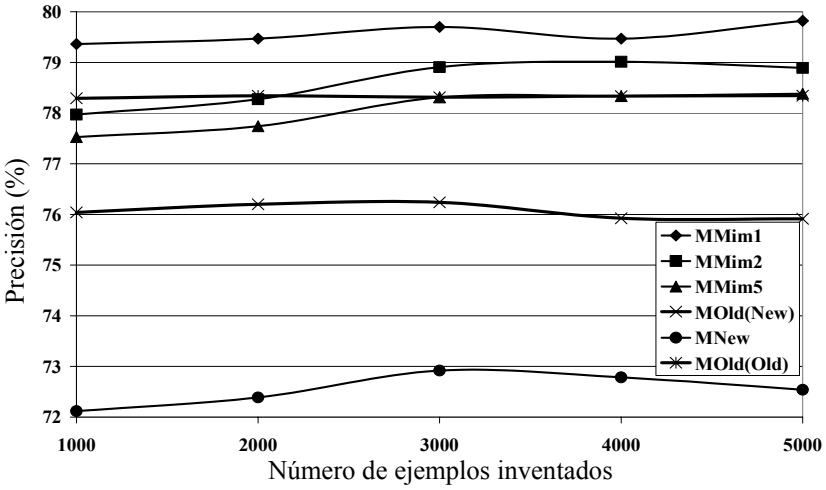


Figura 7.4. Modelo de comportamiento dependiendo de $|T_{Inv}|$ para el conjunto de datos *balance-scale*.

Finalmente, un caso donde no todos los modelos miméticos son capaces de sobrepasar a los otros modelos se muestra en la Figura 7.5.

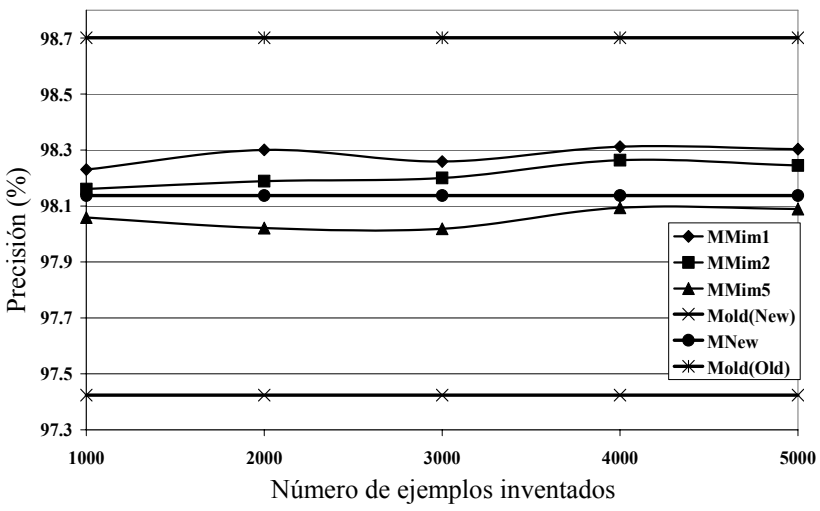


Figura 7.5. Modelo de comportamiento dependiendo de $|T_{Inv}|$ para el conjunto de datos *diabetes*.

Independientemente de estas tres diversas situaciones, y tal como hemos dicho, el número de ejemplos inventados no es muy significativo para los resultados, aunque generalmente es mejor alrededor de los 3000. El máximo no se obtiene en 5000 porque este tamaño también afecta a la proporción del conjunto de datos inventados y el nuevo conjunto de datos.

De aquí en adelante, utilizaremos las medias de los resultados de 1000 a 5000, para conseguir resultados más estables. Cuando comparamos dos aproximaciones utilizando la prueba t de *Student* usamos el caso donde $|T_{Inv}| = 2000$, dado que con este tamaño tenemos aproximadamente el caso promedio.

7.3.3. Relación entre precisión y el factor α

Al final de la sección 7.2 discutimos que, en principio, podemos obtener un factor α, que se define como un mínimo de la proporción óptima entre los tamaños del conjunto de datos inventados y los nuevos datos. En la Tabla 7.4, hemos computado el factor estimado α (α_{est}) aplicando la expresión (7.1) para los conjuntos de datos usados anteriormente, y el factor α real calculado con la expresión (7.2) usado para M_{Mim1}, M_{Mim2} y M_{Mim5} (con $|T_{Inv}| = 2000$). Si analizamos las Tablas 7.4 y 7.5 vemos

que generalmente los mejores resultados de precisión promedio se obtienen con valores del factor α mayores o iguales al estimado. Con lo cual queda demostrado experimentalmente que el valor α estimado es un valor mínimo.

Tabla 7.4. Factores α reales y estimados para los modelos miméticos.

Conjunto de Datos	Clases	Tamaño	α_{Mim1}	α_{Mim2}	α_{Mim5}	α_{est}
anneal	6	898	0.957	0.918	0.817	0.887
audiology	24	226	0.989	0.978	0.947	0.768
balance-scale	3	625	0.970	0.941	0.865	0.945
breast-cancer	2	286	0.986	0.972	0.933	0.747
colic	2	368	0.982	0.965	0.916	0.984
diabetes	2	768	0.963	0.929	0.839	0.898
hepatitis	2	155	0.992	0.985	0.963	0.778
iris	3	150	0.993	0.986	0.966	0.999
vowel	11	990	0.953	0.910	0.802	0.696
zoo	7	101	0.995	0.990	0.976	0.653

La Tabla 7.5 muestra la precisión obtenida con cada modelo mimético con los tres diferentes niveles de sobremuestreo: M_{Mim1}, M_{Mim2} y M_{Mim5} y el modelo mimético usando el valor estimado de α ($M_{Mim\alpha}$). En general, las diferencias entre los resultados obtenidos por los modelos miméticos con grados de sobremuestreo 1, 2, 5 no es demasiado grande, aunque M_{Mim2} y M_{Mim5} son mejores que M_{Mim1}. En promedio la precisión del modelo mimético usando el valor estimado de α es mayor que las precisiones de los demás modelos miméticos.

Tabla 7.5. Precisión promedio de los modelos miméticos. Los dos mejores resultados en negrita.

Conjunto de Datos	M_{Mim1}	M_{Mim2}	M_{Mim5}	$M_{Mim\alpha}$
anneal	94.82	**96.35**	96.27	**96.32**
audiology	59.20	62.19	**63.65**	**64.36**
balance-scale	**79.56**	**78.61**	78.06	78.01
breast-cancer	**68.96**	**68.98**	68.47	67.01
colic	83.77	**83.79**	83.20	**84.05**
diabetes	**70.55**	69.19	68.79	**69.40**
hepatitis	76.95	**77.27**	**77.20**	76.84
iris	92.21	92.54	**92.57**	**93.37**
vowel	55.25	**58.94**	**59.45**	58.76
zoo	74.24	76.16	**78.00**	**83.26**
Media	***75.55***	***76.40***	***76.57***	***77.14***

Si además consideramos los dos mejores resultados de precisión en cada conjunto de datos (ver números en negrita en la Tabla 7.5) se cumple que: M_{Mim1} queda 2 veces en primer lugar y 1 vez en segundo lugar, M_{Mim2} queda 4 veces en primer lugar y 2 veces en segundo lugar, M_{Mim5} queda 1 vez en primer lugar y 4 veces en segundo lugar y, finalmente, $M_{Mim\alpha}$ queda 4 veces en primer lugar y 2 veces en segundo. Entonces, los mejores resultados de precisión se obtienen cuando el valor real de α está más cerca de su valor estimado. Por tales razones en las comparaciones con los demás modelos consideramos al $M_{Mim\alpha}$.

7.3.4. Resultados generales

Una vez establecidos los mejores valores del factor de sobremuestreo para la generación del modelo mimético, pasaremos a analizar éste en relación a los otros modelos disponibles M_{Old} y M_{New}. Fijémonos en estos dos últimos modelos. Hay casos donde M_{Old} es preferible a M_{New} y otros casos donde ocurre lo contario, como se puede ver en la Tabla 7.6 (columnas 2 y 3). Atendiendo a las medias, sin embargo, podemos concluir que M_{Old} es mejor que M_{New}. Esto sucede probablemente, como dijimos, porque el tamaño del conjunto de datos usado para el entrenamiento de M_{New} es demasiado pequeño (que es precisamente el escenario que deseamos examinar aquí).

Tenemos resultados más interesantes cuando comparamos los modelos que toman en cuenta el antiguo modelo y los nuevos datos, tal como los modelos miméticos y el modelo combinado. En la Tabla 7.6 mostramos la precisión media obtenida por el modelo mimético usando el α estimado ($M_{Mim\alpha}$), el modelo combinado con iguales pesos (M_{Comb}) y el modelo combinado con pesos derivados de α ($M_{Comb\alpha}$). Para calcular la precisión de $M_{Comb\alpha}$ empleamos la siguiente fórmula:

$$Acc(M_{Comb\alpha}) = Acc(M_{Old}) \cdot \alpha + Acc(M_{New})(1-\alpha) \qquad (7.3)$$

donde $Acc(M)$ es la precisión del modelo M.

Como podemos ver, hay casos donde M_{Old} gana, pero generalmente M_{Comb}, y especialmente el modelo mimético $M_{mim\alpha}$, (ver Tabla 7.6) son mejores.

Tabla 7.6. Precisión promedio de los diferentes modelos.

Conjunto de Datos	$M_{Old}(New)$	M_{New}	M_{Comb}	M_{Comb_α}	M_{Mim_α}
#1	89.23	90.78	93.22	89.44	96.32
#2	61.31	53.39	61.75	61.31	64.36
#3	75.86	72.43	76.67	76.18	78.01
#4	68.75	67.70	68.59	69.13	67.01
#5	84.60	79.64	82.70	84.49	84.05
#6	71.46	68.97	70.65	71.72	69.40
#7	76.00	76.55	78.17	78.28	76.84
#8	93.38	81.67	91.76	93.41	93.37
#9	57.39	46.27	58.81	58.15	58.76
#10	66.20	59.84	70.56	66.26	83.26
Media	***74.42***	***69.72***	***75.29***	***74.84***	***77.14***

Aunque la precisión media de M_{Mim_α} es sensiblemente mejor, la comparación más ajustada puede darse al oponer la técnica mimética con la combinación de los modelos M_{Old} y M_{New}. La Tabla 7.7 muestra la prueba t de *Student* entre M_{Mim_α} y cada uno de los modelos combinados. M_{Mim_α} gana a M_{Comb_α} en 4 casos, pierde en 3 y empata en los 3 restantes. En el caso de comparar M_{Mim_α} y M_{Comb}, M_{Mim_α} gana a M_{Comb} en 5 casos, pierde en 1 y empata en los 4 restantes.

Aunque la Tabla 7.7 muestra una ventaja leve de M_{Mim_α} sobre M_{Comb_α}, es importante destacar que la técnica combinada no produce un modelo comprensible, mientras que la técnica mimética sí, por lo que existen muchas situaciones donde no será aplicable la técnica combinada.

Tabla 7.7. Resultados de las pruebas t de *Student* entre M_{Mim_α} y los modelos combinados (M_{Comb} y M_{Comb_α}). 'V' denota que M_{Mim_α} gana, 'X' denota que M_{Mim_α} pierde y '–' denota un empate. El resumen se interpreta como (V/–/X).

Conjunto de datos	M_{Mim_α} vs. M_{Comb}	M_{Mim_α} vs. M_{Comb_α}
anneal	V	V
audiology	V	V
balance-scale	–	V
breast-cancer	X	X
colic	V	–
diabetes	–	X
hepatitis	–	X
iris	V	–
vowel	–	–
zoo	V	V
resumen	(5/4/1)	(4/3/3)

Finalmente, la Tabla 7.8 muestra una evaluación de la significancia de M_{Mim_α} con respecto a M_{Old} y M_{New}, la cual, unida con los resultados promedio mostrados en la Tabla 7.6, resume la recomendación de usar M_{Mim_α} en este escenario.

Tabla 7.8. Resultados de las pruebas t de *Student* entre M_{Mim_α} vs. M_{Old} y M_{New}. 'V' denota que M_{Mim_α} gana, 'X' denota que M_{Mim_α} pierde y '–' denota un empate. El resumen se interpreta como (V/–/X).

Conjunto de datos	M_{Mim_α} vs. M_{Old}	M_{Mim_α} vs. M_{New}
anneal	V	V
audiology	V	V
balance-scale	V	V
breast-cancer	X	–
colic	–	V
diabetes	X	–
hepatitis	–	–
iris	–	V
vowel	–	V
zoo	V	V
resumen	(4/4/2)	(7/3/0)

En general, M_{Mim_α} gana a M_{Old} en 4 casos, pierde en 2 y empata en los 4 restantes. Referente a M_{New}, M_{Mim_α} gana a M_{New} en 7 casos, pierde en 0 casos y empata en los 3 restantes. Por lo tanto, podemos concluir que M_{Mim_α} es más adecuado que M_{Old} y M_{New}.

7.4. Discusión

En la sección anterior hemos demostrado que el modelo mimético puede mantener parcialmente el buen funcionamiento en las regiones que no han cambiado, y además puede integrar las reglas que cubren la nueva región. Sobre todo, el modelo mimético es, en general, mejor que el modelo antiguo y el nuevo modelo por separado. El modelo combinado no es generalmente mejor que el modelo mimético, aunque en algunas situaciones específicas puede ser mejor que el modelo antiguo, el nuevo modelo y el modelo mimético. No obstante, en promedio, el modelo mimético es generalmente mejor que la técnica combinada y, además, la técnica mimética genera un único modelo, a diferencia de la técnica combinada.

Una de las ventajas de nuestra aproximación es su flexibilidad ya que nuestro método hace pocas suposiciones sobre el tamaño de los nuevos datos, la disponibilidad de datos antiguos, la precisión del modelo antiguo para la situación antigua, la distribución de datos de los nuevos datos o el modelo antiguo. Los experimentos se han realizado sobre conjuntos de datos con variedad en las características de los mismos: diferentes tamaños, diferente número de clases, con sólo datos numéricos, con sólo datos discretos y combinación de ambos tipos de datos, con y sin datos faltantes.

Si el modelo o los datos originales vienen desde una distribución errónea, nuestro método no resulta afectado particularmente ya que usamos una distribución equivocada para el modelo original (usamos una distribución uniforme). Para los nuevos datos, si proceden de una distribución diferente a la distribución original, debemos presumir que esta nueva distribución es correcta, y por ello, usamos los datos nuevos para evaluar todos los modelos y compararlos (original, nuevo, mimético y combinado).

Desde el punto de vista de la “forma” de los modelos, las diferencias entre el modelo original y el modelo mimético son comúnmente no mínimas, a diferencia de otros métodos de revisión de modelos donde no existen excesivas variaciones entre los modelos original y revisado. Por ejemplo, en algún conjunto de datos, tal como *balance-scale*, incluso el atributo de la partición inicial (la raíz del árbol) es diferente. En este caso, es "*left-distance*" para el modelo original mientras que es "*right-distance*" para el modelo mimético. Esto sugiere que la técnica mimética es un método sencillo que permite realizar revisión de teorías con cambios profundos y no con “parches” sobre la teoría original.

El escenario que hemos analizado a fondo puede ser encontrado en muchas organizaciones, puesto que es frecuente que los cambios provengan de un área que tenía una influencia pequeña en el modelo en el pasado pero que tiende a "aparecer" más adelante. No obstante, hay otro escenario que puede ser también frecuente: el cambio sucede globalmente y no en una región particular. En este caso, parece más difícil ver cómo los datos del modelo antiguo y los datos del nuevo modelo pueden coexistir.

Por ejemplo, la Figura 7.6 muestra el modelo antiguo y un nuevo modelo con un cambio global.

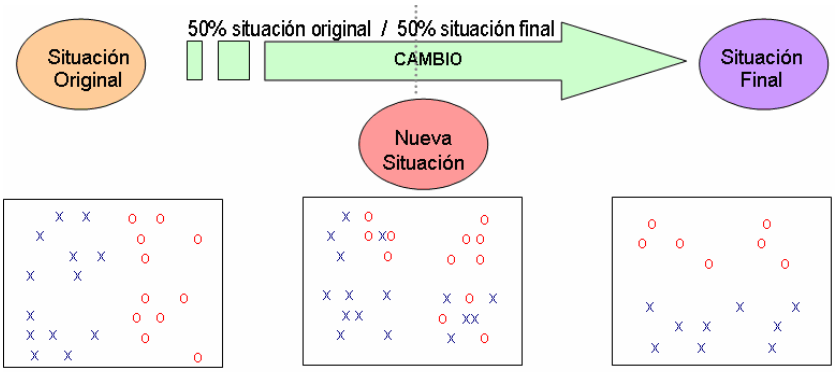


Figura 7.6. Cambio del modelo de forma global y profunda.

En este escenario, si la nueva situación donde detectamos la necesidad de la revisión del modelo está en la mitad del camino entre la situación original y la situación final, tenemos un cambio muy fuerte. Si congelamos el cambio en esta foto a la mitad del camino, el 50% del espacio sigue estando inalterado, pero ahora el resto tiene una mezcla del antiguo y nuevo modelo. En este caso, intentamos representar una situación donde los cambios serán profundos y totales, es decir hay un período de transición donde coexisten la situación original y la situación final. Diferentes escenarios pueden considerarse, por ejemplo si tomamos la foto al final del camino. En ambos casos, la parte del modelo antiguo se preserva (el 50% de las regiones no cambian en este caso).

Con esta representación mostrada en la Figura 7.6, está claro que si tenemos muy pocos datos de la nueva situación es razonable contar con que el clasificador mimético puede capturar bien las regiones comunes (porque utilizamos los ejemplos inventados para esto) y podemos conseguir una aproximación aleatoria en el resto. Esto significaría que la precisión de M_{Mim} para el ejemplo anterior estaría cerca del 75% y con muy pocos datos (puesto que este ejemplo es muy fácil de aprender) es razonable pensar que M_{New} debería presentar una precisión más baja que ese valor. Este resultado se puede obtener en general dependiendo de la complejidad del problema y del tamaño del nuevo conjunto de datos.

Sin embargo, el modelo antiguo en la Figura 7.6 también se comporta cerca del 75%. Por lo tanto, parece que hay situaciones (como ésta) donde puede ser demostrado que, en teoría, el modelo mimético no es mejor que el modelo antiguo.

No obstante, hay otros casos donde el modelo mimético podría ser mejor que el modelo antiguo: por ejemplo, la situación en la Figura 7.7. Dentro de cada patrón las clases son representadas por *X* e *Y*.

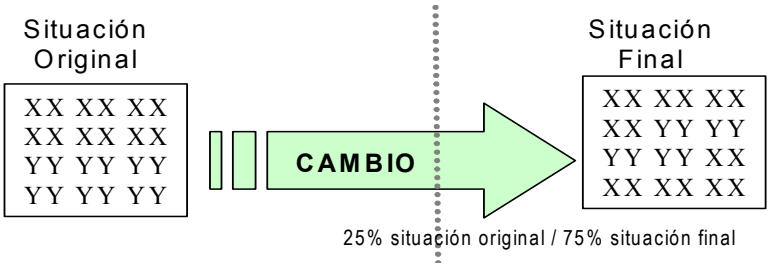


Figura 7.7. Cambio del modelo de forma global y profunda donde la técnica mimética funciona bien.

Una explicación general de por qué la precisión del modelo mimético puede ser mejor que la del modelo antiguo, está en el número de cambios. Un cambio es la sustitución de una pareja de clases, ya sea *XX* o *YY* por su clases contrarias, *YY* o *XX* respectivamente. Observesé en la Tabla 7.9 que a mayor número de cambios (mayor de 4) el modelo mimético gana. Cuantos menos cambios existan más próximos estaremos a la situación original y si existen muchos cambios entonces estaremos casi en la nueva situación, por lo que el modelo mimético es bueno en una situación intermedia.

Con el objetivo de analizar si son más frecuentes los casos como el de la Figura 7.6 que los casos como el de la Figura 7.7, hemos analizado experimentalmente diez escenarios con diferentes patrones originales y finales (Tabla 7.9). Se han obtenido las precisiones de los modelos mimético, antiguo y nuevo aprendidos utilizando estos patrones. En algunos de ellos (las filas grises en la Tabla 7.9, el modelo mimético puede superar en precisión al modelo antiguo y al nuevo modelo (dado un conjunto de datos nuevo muy pequeño). Este es el caso de la Figura 7.7 y que corresponde a la fila 8 de la Tabla 7.9.

Tabla 7.9. Precisiones promedio obtenidas con los experimentos con diferentes patrones de la situación original y final.

Experimento	Patrón Original	Patrón Final	Cambios	M_{Mim}	M_{Old}	M_{New}
1	*XX XX YY* *XX XX YY* *XX YY YY* *XX YY YY*	*XX YY YY* *XX YY YY* *XX XX YY* *XX XX YY*	4	76.72	74.58	78.71
2	*YY XX YY* *XX XX XX* *YY YY YY* *XX YY XX*	*YY YY YY* *XX YY XX* *YY XX YY* *XX XX XX*	4	72.71	74.71	60.58
3	*XX XX YY* *XX XX XX* *YY YY YY* *XX YY YY*	*XX YY YY* *XX YY XX* *YY XX YY* *XX XX YY*	4	73.21	75.01	61.27
4	*XX XX YY* *XX XX YY* *XX YY YY* *XX YY YY*	*YY YY YY* *XX YY YY* *XX XX YY* *XX XX XX*	6	64.29	62.55	68.81
5	*XX XX YY* *XX XX YY* *XX YY YY* *XX YY YY*	*YY YY XX* *XX YY YY* *XX XX YY* *XX XX YY*	6	63.73	63.63	62.25
6	*XX XX YY* *XX XX YY* *XX YY YY* *XX YY YY*	*YY YY XX* *XX YY YY* *YY XX YY* *XX XX XX*	6	51.99	50.41	56.33
7	*XX XX XX* *XX XX XX* *YY YY YY* *YY YY YY*	*XX XX XX* *YY YY YY* *YY YY YY* *XX XX XX*	6	64.28	62.31	64.58
8	*XX XX XX* *XX XX XX* *YY YY YY* *YY YY YY*	*XX XX XX* *XX YY YY* *YY YY XX* *XX XX XX*	6	63.09	62.4	60.95
9	*XX XX XX* *XX XX XX* *YY YY YY* *YY YY YY*	*XX XX XX* *XX YY YY* *YY YY YY* *YY XX XX*	4	51.68	49.77	57.06
10	*YY XX YY* *XX XX XX* *YY YY YY* *XX YY XX*	*YY YY YY* *XX XX XX* *YY XX YY* *XX YY XX*	2	61.6	63.3	57.14
Media				64.33	63.87	62.79

Una taxonomía completa de los casos con los cambios globales donde la técnica mimética podría trabajar es difícil de establecer, puesto que depende de la complejidad del problema (la definición de problema original y final), los cambios de la distribución de la clase, el porcentaje de coincidencias entre la situación original y la situación final, y el punto de la "foto" donde realmente estamos: a mitad del camino, más cerca de la situación original, más cercano a la situación final, etc.

No obstante, la técnica mimética es fácil de utilizar, se puede aplicar fácilmente y absolutamente con cualquier técnica de aprendizaje computacional. Por lo tanto, no hay razón para no

emplearla en la revisión de modelos como una posibilidad razonable. Para cada situación específica podemos evaluar el modelo antiguo, el nuevo modelo y el modelo mimético, y elegir el que mejor se ajuste a la nueva situación.

7.5. Conclusiones

Hemos presentado un nuevo método general para la revisión de modelos basado en un uso balanceado del modelo antiguo (a través de la generación de un conjunto de datos inventados) y los nuevos datos, los cuales sirven para obtener un modelo mimético.

Hemos demostrado que el modelo mimético se comporta mejor que el modelo antiguo, un nuevo modelo y su combinación en escenarios donde los cambios suceden en las regiones que no eran relevantes o inexistentes en la situación original (escenario de novedad). Éste es especialmente el caso cuando tenemos pocos datos nuevos disponibles para conseguir entrenar un buen modelo nuevo. Esta situación es usual en la revisión de modelos ya que en la situación opuesta (una gran cantidad de nuevos datos) la revisión de modelos sería, en general, inútil ya que sería preferible entrenar un nuevo modelo con los datos nuevos.

Un escenario diferente sucede cuando los cambios ocurren de una manera más global y los nuevos datos contradicen, en cierto modo, la situación original. Aquí, los casos donde es útil la técnica mimética no son tan generales como en el escenario anterior. En cualquier caso, el modelo mimético se puede construir muy fácilmente y se puede comparar con los antiguos y nuevos modelos para considerar si el modelo mimético es eventualmente mejor o no.

Resumiendo, en este capítulo hemos demostrado que cualquier técnica de aprendizaje computacional se puede utilizar como método general de revisión de modelos, independientemente de cómo se generó el modelo y sin necesidad de disponer de los datos ni de la distribución original de los mismos.

Capítulo 8

Contextualización por cambios de formato

Como ya decíamos en el capítulo anterior, en la vida de un modelo se presentan cambios en el contexto. Estos cambios pueden deberse a que han aparecido o desaparecido atributos o clases en la nueva situación. Este escenario de cambio en el formato de los datos es muy común en la realidad podemos enunciar múltiples ejemplos:

1. Considere el caso de la aparición de una nueva enfermedad la cual comparte síntomas con otras enfermedades, aquí tenemos un ejemplo de la aparición de un nuevo valor de clase (la nueva enfermedad).
2. Antiguamente en un banco consideraban la autorización de créditos con una respuesta de si o no, pero en su modernización ahora consideran la opción de conceder el crédito condicionando al deudor, entonces ahora se deben considerar tres valores de clase y no dos.
3. En el desarrollo y organización de un supermercado se crea una nueva forma de etiquetar o reconocer a los productos, aquí se crea un nuevo atributo.
4. Un gobierno crea una nueva clave única de identificación (no existía anteriormente) de sus ciudadanos, nace un nuevo atributo.

Así, nuestro objetivo en este capítulo es estudiar la adaptación de los modelos ante cambios en el formato de los datos, ya sean por incremento o disminución de atributos o de valores de la clase. Para realizar este estudio emplearemos los árboles de decisión creados con el algoritmo J48.

Este capítulo está organizado como sigue. En la sección 1 estudiamos el comportamiento del algoritmo J48 ante valores perdidos. En la sección 2 realizamos el análisis del escenario donde aparece un nuevo atributo, estudiando diferentes casos. El caso más destacado es cuando el atributo es muy importante cuya evaluación experimental se realiza en la sección 3. Luego en la sección 4, realizamos la evaluación experimental para cuando aparece un nuevo valor de la clase. Finalmente, presentamos las conclusiones del capítulo y trabajo futuro.

8.1. Los árboles de decisión y los datos perdidos

El objetivo de esta sección es saber qué le pasa a un árbol de decisión construido con el algoritmo J48 cuando se asigna un valor nulo (perdido, faltante, desconocido) en los datos. Esto es importante ya que utilizamos el J48 como modelo de referencia y modelo mimético. Para experimentar usamos el conjunto de datos *weather*. Se realizan validaciones cruzadas de 10×10.

La Tabla 8.1 muestra los valores contenidos en el conjunto de datos *weather*.

Tabla 8.1. Conjunto de datos *weather*.

No.	outlook Nominal	temperature Numeric	humidity Numeric	windy Nominal	**play** Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	72.0	90.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	71.0	91.0	TRUE	no

En primer lugar, construimos un árbol de decisión con los datos anteriores, que no contiene datos faltantes que mostramos en la Figura 8.1.

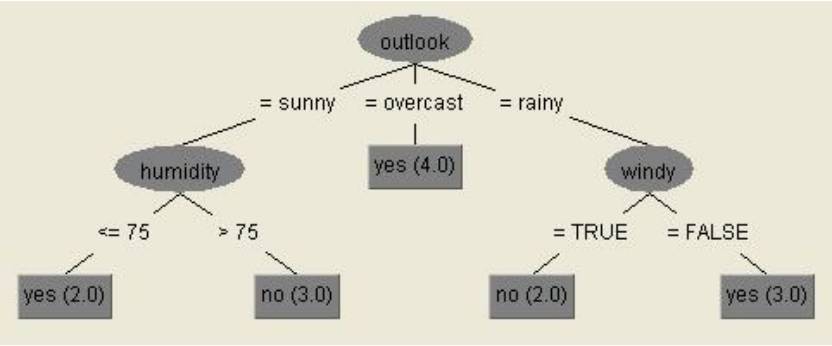


Figura 8.1. Árbol de decisión para el conjunto de datos *weather*.

Ahora, asignaremos un valor nulo al atributo principal (*outlook*) en el primer registro, como se muestra en la Tabla 8.2.

Tabla 8.2. Conjunto de datos *weather*, con un nulo en el atributo *outlook* del primer ejemplar.

No.	outlook Nominal	temperature Numeric	humidity Numeric	windy Nominal	**play** Nominal
1	?	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	72.0	90.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	71.0	91.0	TRUE	no

Analizando el árbol de decisión (ver Figura 8.2) resultante de hacer el cambio anterior a los datos, observamos que se reparte el peso del ejemplo en todas las ramas (todos los posibles valores) del atributo asignado con el nulo. Las hojas que se afectan

con este peso son a las que puede llegarse tomando en cuenta los demás valores de los atributos conocidos. Esto significa que un valor nulo puede tomar cualquier valor del atributo.

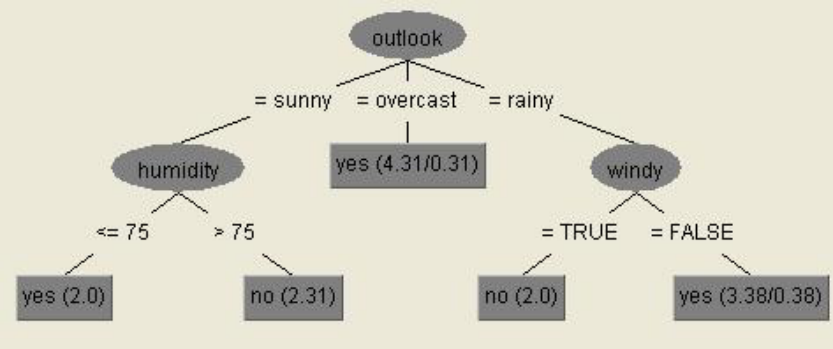


Figura 8.2. Árbol de decisión para el conjunto de datos *weather*, con un nulo en el atributo *outlook* del primer registro

8.2. Escenario con la aparición de un nuevo atributo

Analizaremos cómo son los árboles de decisión construidos con J48 al aparecer un nuevo atributo.

El procedimiento seguido es: partiremos de un árbol completo y luego procesaremos algún atributo. Estudiaremos la eliminación de un atributo.

Se realizaron los siguientes pasos:

a) Se dispone del conjunto de datos *weather*.
b) Se realiza alguna acción sobre un atributo, como por ejemplo eliminarlo, asignar nulos o agregar un atributo auxiliar para relacionarlo con los valores nulos.
c) Se construye el árbol y luego se analiza.

Las opciones a analizar consideran la aparición de un atributo importante (atributo en la raíz del árbol), o uno de baja importancia (antes de las hojas). La Tabla 8.3 muestra un resumen de las opciones a analizar.

Tabla 8.3. Lista de acciones estudiadas sobre un árbol de decisión creado a partir del conjunto de datos *weather*.

Opción	Atributo	Acción
1	(árbol normal)	
2	*outlook*	eliminar
3	*outlook*	poner nulos
4	existe (+ opción 3)	crear
5	*humidity*	eliminar
6	*humidity*	poner nulos
7	existe (+ opción 6)	crear

El atributo "existe" tiene valores auxiliares de tipo nominal que nos dice cuando el valor del atributo procesado (*outlook* o *humidity*) se conoce (T) o no (F). A continuación explicamos cada uno de los casos en detalle.

Opción 1. Árbol normal.

El árbol original se muestra en la Figura 8.1. Teniendo 9 (64.29 %) instancias correctamente clasificadas y 5 (35.71 %) incorrectamente clasificadas. En éste árbol los atributos importantes son: *outlook*, *humidity* y *windy*.

Opción 2. Eliminando Outlook

Si eliminamos el atributo más importante (*outlook*) tenemos como resultado un árbol que lo clasifica todo como yes. Teniendo 9 (64.29 %) instancias incorrectamente clasificadas y 5 (35.71 %) correctamente clasificadas.

Opción 3. Asignando nulos a outlook

Si ponemos aleatoriamente (uno si, uno no) valores nulos a la mitad de los datos de *outlook*. Todos los ejemplares son clasificados como yes, es decir, el árbol degenera en una hoja. Teniendo 7 (50 %) instancias incorrectamente clasificadas y 7 (50 %) correctamente clasificadas.

Por lo tanto, en este caos, el efecto de asignar nulos a la mitad de ejemplares es similar a eliminar el atributo. Del análisis de las opciones 2 y 3 podemos concluir que al eliminar un atributo muy importante, en este caso, el árbol se convierte en una hoja.

Opción 4. Existe + opción 3

Agregamos el atributo auxiliar para la existencia o no del valor (los valores T o F asignados al atributo "existe" respectivamente). Los valores nulos se asocian con el valor de F. La Figura 8.3 muestra el árbol de decisión obtenido con esta opción. Teniendo como resultado a 8 (57.14 %) instancias incorrectamente clasificadas y 6 (42.86 %) correctamente clasificadas.

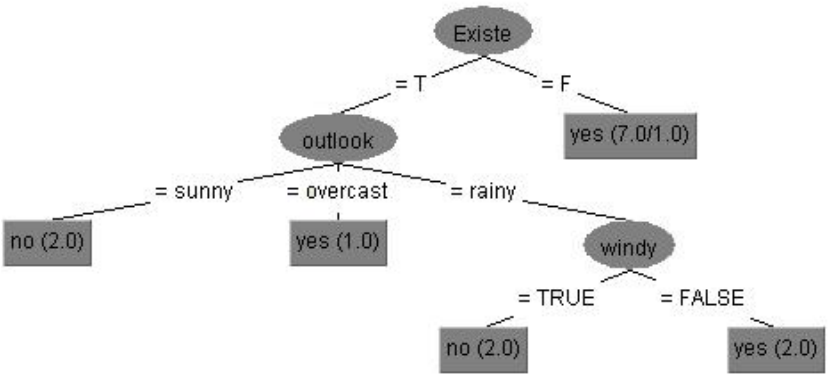


Figura 8.3. Árbol de decisión para el conjunto de datos *weather*, con la mitad de nulos en el atributo *outlook* y la adición del un nuevo atributo existe.

En este caso podemos observar que el atributo auxiliar se convierte en la raíz del árbol, es decir, se generan dos subárboles uno para los valores conocidos y otro para los nulos.

Opción 5. Eliminando humidity.

Ahora trabajamos con un atributo de baja importancia (además de ser numérico), seleccionamos el *humidity*.

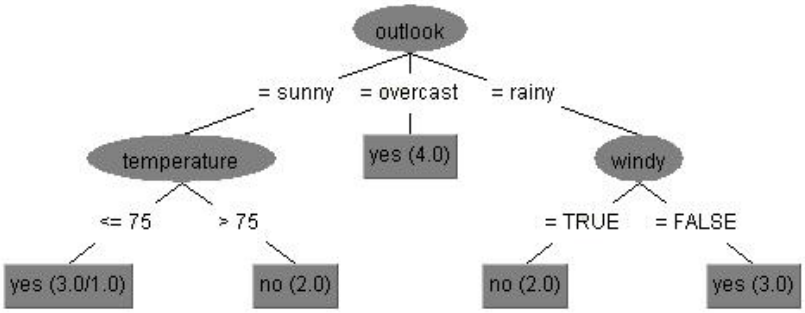


Figura 8.4. Árbol de decisión para el conjunto de datos *weather* habiendo eliminado el atributo *humidity*.

El atributo fue sustituido por *temperature*. Teniendo como resultado a 8 (57.14 %) instancias incorrectamente clasificadas y 6 (42.86 %) correctamente clasificadas. Observamos que al eliminar un atributo de baja importancia, éste es remplazado por otro de importancia similar.

Opción 6. Asignado nulos a humidity.

Si ponemos aleatoriamente (uno si, uno no) valores nulos a la mitad de los datos de *humidity*, tenemos el siguiente árbol.

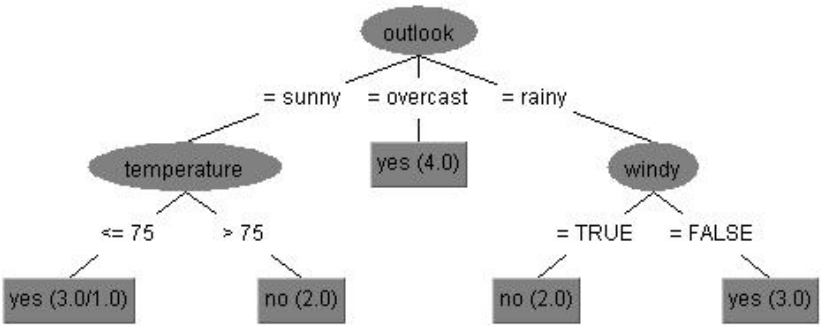


Figura 8.5. Árbol de decisión para el conjunto de datos *weather* habiendo asignado la mitad de valores nulos al atributo *humidity*.

El resultado es igual al caso anterior. Podemos concluir que cuando se ponen nulos en la mitad de los valores de un atributo poco importante otro atributo lo sustituye.

Opción 6. Existe + opción 6.

Si ponemos aleatoriamente valores nulos a la mitad de los datos de *humidity* y agregamos el atributo auxiliar para la existencia o no del valor (similar a lo realizado en la opción 4) tenemos como resultado a 9 (64.29 %) instancias incorrectamente clasificadas y 5 (35.71 %) correctamente clasificadas.

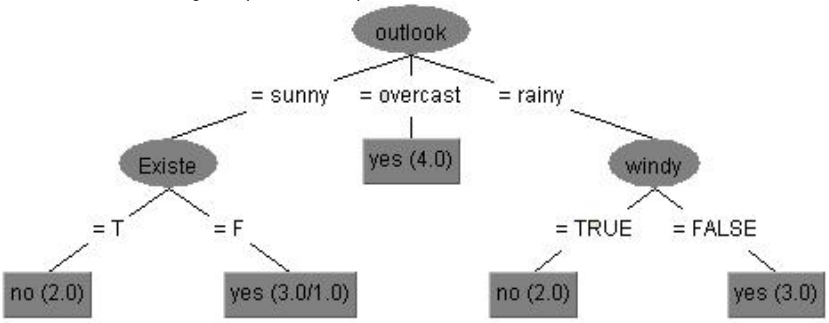


Figura 8.6. Árbol de decisión para el conjunto de datos *weather* habiendo asignado la mitad de valores nulos al atributo *humidity* y la adición del atributo existe.

El atributo artificial existe sustituye al *humidity*. Podemos afirmar que cuando al mismo tiempo de agregar nulos a un atributo de baja importancia se adiciona un atributo auxiliar se observa que el atributo auxiliar sustituye al atributo de baja importancia.

La Tabla 8.4 muestra un resumen de los resultados encontrados en todas las opciones.

Tabla 8.4. Resultados al realizar acciones sobre los atributos de un árbol de decisión creado con el conjunto de datos *weather*.

#	Atributo	Acción	Resultado	Precisión
1			Árbol normal	64.3
2	*outlook*	eliminar	Hoja	35.7
3	*outlook*	poner nulos	Hoja	50.0
4	existe (+ opción 3)	crear	2 subárboles	42.7
5	*humidity*	eliminar	Sustituido por *temperature*	42.7
6	*humidity*	poner nulos	Sustituido por *temperature*	42.7
7	existe (+ opción 6)	crear	Sustituido por existe	35.7

A la vista de este ejemplo, el poner nulos a un atributo es restarle importancia tanto como si lo elimináramos del árbol.

8.3. Experimentos cuando aparece un atributo importante

En esta sección presentamos los experimentos para analizar la técnica de revisión mimética de modelos para cuando aparece un nuevo atributo importante sin cambiar de problema. Asimismo, compararemos el modelo mimético generado con los modelos viejo (M_{Old}), nuevo (M_{New}) y combinado (M_{Comb}).

8.3.1. Configuración experimental

Seguimos la misma nomenclatura y configuración de la evaluación experimental realizada en la sección 7.3. Las únicas diferencias son el procedimiento para generar los conjuntos de datos y los conjuntos de datos del repositorio UCI empleados (ver Tabla 8.5).

Tabla 8.5. Información de los conjuntos de datos utilizados en los experimentos cuando aparece un nuevo atributo importante.

No.	Conjunto de Datos	Atributos		Clases	Tamaño	Datos Faltantes
		Num.	Nom.			
1	anneal	6	32	6	898	No
2	balance-scale	4	0	3	625	No
3	diabetes	8	0	2	768	No
4	iris	4	0	3	150	No
5	vowel	10	3	11	990	No

El procedimiento (ver Figura 8.7) seguido para crear los conjuntos de datos es como sigue.

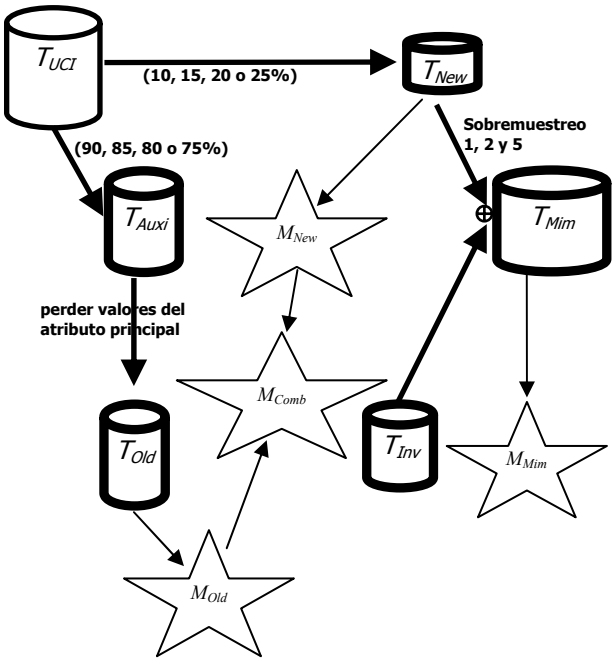


Figura 8.7. Esquema de los conjuntos de datos y modelos empleados en los experimentos cuando aparece un nuevo atributo.

El conjunto de datos original del repositorio UCI lo definimos como T_{UCI}. Creamos aleatoriamente el conjunto de datos nuevo (T_{New}) con diferentes porcentajes (10, 15, 20 o 25) del tamaño de T_{UCI}. Con los datos que no fueron elegidos para el conjunto de datos nuevo se forma el conjunto de datos auxiliar T_{Auxi}. A este conjunto auxiliar se le elimina el atributo principal y así generamos el conjunto de datos antiguo T_{Old}. El conjunto de datos inventados (T_{Inv}) se genera con un muestreo uniforme usando T_{Old} para obtener mínimos y máximos. El valor del atributo más importante (que inicialmente está indefinido) en T_{Inv} lo asignamos con el valor más probable, la media de los valores observados en el T_{New}. Esta asignación la realizamos pues como tratamos con datos numéricos, así si efectuamos una regresión lineal sobre los datos nuevos obtendríamos precisamente el valor medio de los mismos. El conjunto de datos de entrenamiento para los modelos miméticos (T_{Mim}) se crean uniendo T_{Inv} y T_{New} con sobremuestreo de 1, 2 y 5.

La configuración del entrenamiento y validación de los modelos miméticos es similar a la que mostramos en la Tabla 7.3.

Utilizaremos las medias de las precisiones de los modelos miméticos de 1000 a 5000, para conseguir una visión global de los resultados y posteriormente realizaremos las pruebas estadísticas de significancia.

8.3.2. Resultados promedio

A continuación mostramos en tablas los resultados promedio obtenidos con los diferentes porcentajes experimentados. También mostramos, las gráficas de precisión contra los diferentes tamaños del conjunto de datos inventados.

Tabla 8.6. Resultados promedio usando un 10% de los datos para T_{New}.

Modelo	Precisión	MSE	Logloss	AUC
MComb	80.78	0.000	0.000	0.000
MMim1	81.47	0.238	101.565	0.683
MMim2	81.46	0.238	160.320	0.681
MMim5	79.89	0.252	191.474	0.686
MNew	71.01	0.297	183.834	0.648
MOld(New)	83.66	0.242	42.239	0.880
MOld(Old)	83.74	0.241	42.070	0.879

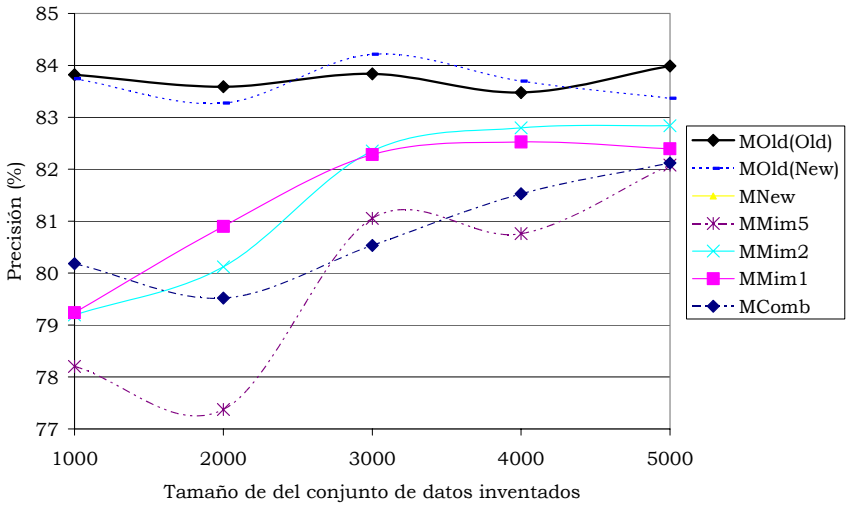


Figura 8.8. Resultados promedio usando un 10% de los datos para T_{New}.

Tabla 8.7. Resultados promedio usando un 15% de los datos para T_{New}.

Modelo	Precisión	MSE	Logloss	AUC
MComb	81.70	0.000	0.000	0.000
MMim1	81.91	0.241	96.506	0.770
MMim2	81.55	0.248	164.277	0.758
MMim5	80.61	0.255	189.030	0.766
MNew	75.02	0.275	150.725	0.735
MOld(New)	83.36	0.247	45.738	0.874
MOld(Old)	83.67	0.241	44.492	0.878

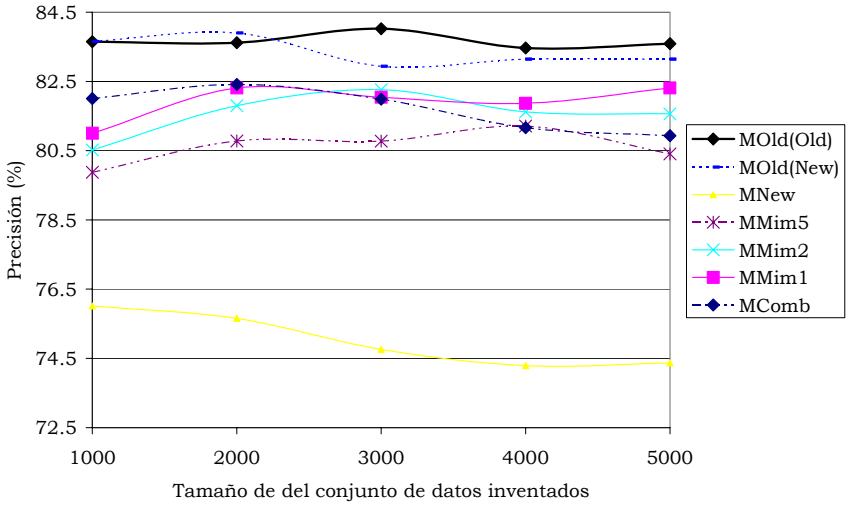


Figura 8.9. Resultados promedio usando un 15% de los datos para T_{New}.

Tabla 8.8. Resultados promedio usando un 20% de los datos para T_{New}.

Modelo	Precisión	MSE	Logloss	AUC
MComb	82.17	0.000	0.000	0.000
MMim1	81.80	0.245	95.538	0.818
MMim2	81.69	0.255	165.750	0.802
MMim5	80.67	0.264	190.543	0.801
MNew	76.84	0.268	131.623	0.790
MOld(New)	82.95	0.252	46.697	0.874
MOld(Old)	83.21	0.243	46.544	0.876

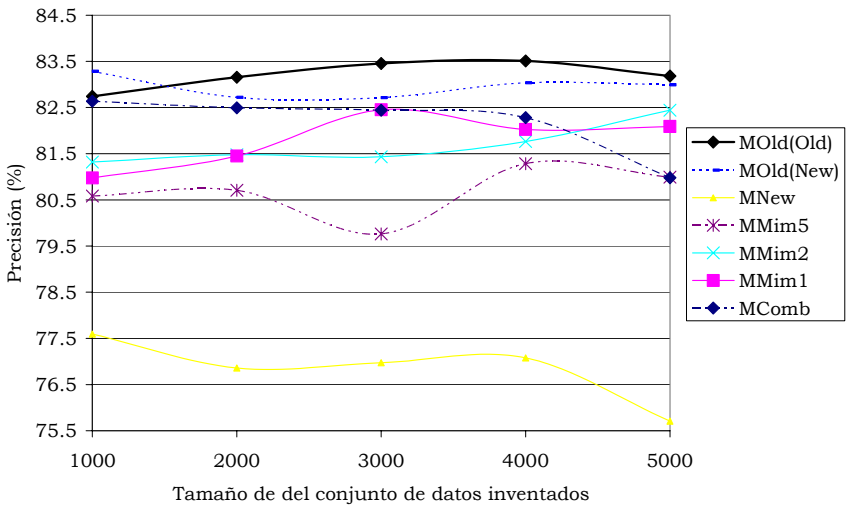


Figura 8.10. Resultados promedio usando un 20% de los datos para T_{New}.

Tabla 8.9. Resultados promedio usando un 25% de los datos para T_{New}.

Modelo	Precisión	MSE	Logloss	AUC
MComb	83.09	0.000	0.000	0.000
MMim1	82.57	0.243	92.921	0.853
MMim2	82.21	0.254	161.828	0.837
MMim5	81.50	0.260	182.713	0.834
MNew	78.66	0.261	115.801	0.833
MOld(New)	82.86	0.253	50.246	0.876
MOld(Old)	82.68	0.245	49.576	0.875

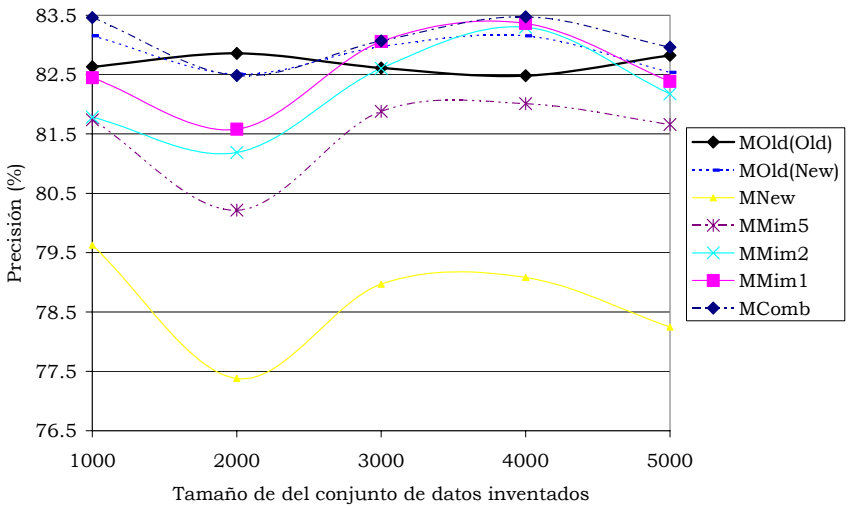


Figura 8.11. Resultados promedio usando un 25% de los datos para T_{New}.

Para estos conjuntos de datos no se aprecia mucho la necesidad de la revisión en los diferentes casos, esto se debe a que al eliminar un atributo importante otro de igual importancia o similar lo sustituye. Por lo que para que se de la necesidad de la revisión es necesario que la importancia del atributo que aparece realmente sea muy importante. Aun así, la precisión promedio del MOld(Old) es regularmente mayor que la precisión de MOld(New) y mucho mayor que la precisión de MNew (obsérvese que siempre es la precisión más baja) y suponemos la necesidad de la revisión. Ahora veamos cual de los métodos miméticos es mejor en precisión, a simple vista MMim1 se ve como el mejor.

8.3.3. Análisis de la significancia para la precisión

Observemos que cuando el conjunto de datos nuevos De acuerdo a las figuras anteriores consideremos el caso promedio el de tamaño 3000, para realizar las pruebas estadísticas de significancia con respecto a la precisión. Ahora realicemos las pruebas t de *Student* a todos los modelos cuando usamos un 25% de los datos para T_{New}.

La interpretación de la notación en la Tabla 8.10 de resultado para las pruebas t de *Student* es: el modelo de la fila (Gana/ Empata/Pierde) al modelo de la columna. La variable medida en las pruebas t es el coste global.

Tabla 8.10 Pruebas t para modelos obtenidos usando un 25% de los datos para T_{New}.

	MOld(Old)	MOld	MNew	MComb	MMim1	MMim2	MMim5
MOld(Old)	-	(0/5/0)	(1/4/0)	(0/5/0)	(0/4/1)	(0/4/1)	(1/3/1)
MOld	(0/5/0)	-	(1/4/0)	(0/5/0)	(0/4/1)	(1/3/1)	(2/2/1)
MNew	(0/4/1)	(0/4/1)	-	(0/4/1)	(0/3/2)	(0/3/2)	(1/2/2)
MComb	(0/5/0)	(0/5/0)	(1/4/0)	-	(0/4/1)	(0/5/0)	(0/4/1)
MMim1	(1/4/0)	(1/4/0)	(2/3/0)	(1/4/0)	-	(0/5/0)	(1/3/1)
MMim2	(1/4/0)	(1/3/1)	(2/3/0)	(0/5/0)	(0/5/0)	-	(0/5/0)
MMim5	(1/3/1)	(1/2/2)	(2/2/1)	(0/4/1)	(1/3/1)	(0/5/0)	-

Efectivamente el MMim1 es el que gana más veces a los demás modelos. Sin embargo, la mayoría tiende al empate.

8.4. Experimentos cuando aparece una nueva clase

En esta sección presentamos los experimentos para analizar la técnica de revisión mimética de modelos para cuando aparece una nueva clase sin cambiar de problema. Asimismo, compararemos el modelo mimético generado con los modelos viejo (M_{Old}), nuevo (M_{New}) y combinado (M_{Comb}).

8.4.1. Configuración experimental

Seguimos la misma nomenclatura y configuración de la evaluación experimental realizada en la sección 7.3. Las únicas diferencias son el procedimiento para generar los conjuntos de datos y los conjuntos de datos del repositorio UCI empleados (ver Tabla 8.11), los cuales solamente tienen tres clases.

Tabla 8.11. Información de los conjuntos de datos empleados en los experimentos cuando aparece una nueva clase.

No.	Conjunto de datos	Atr.	Atr. Num.	Atr. Nom.	Clases	Tamaño	Faltan
1	balance-scale	4	4	0	3	625	No
2	cmc	9	2	7	3	1,473	No
3	hayes-roth	4	0	4	3	132	No
4	iris	4	4	0	3	150	No
5	waveform-5000	40	40	0	3	5,000	No

El procedimiento (ver Figura 8.12) seguido para crear los conjuntos de datos es como sigue.

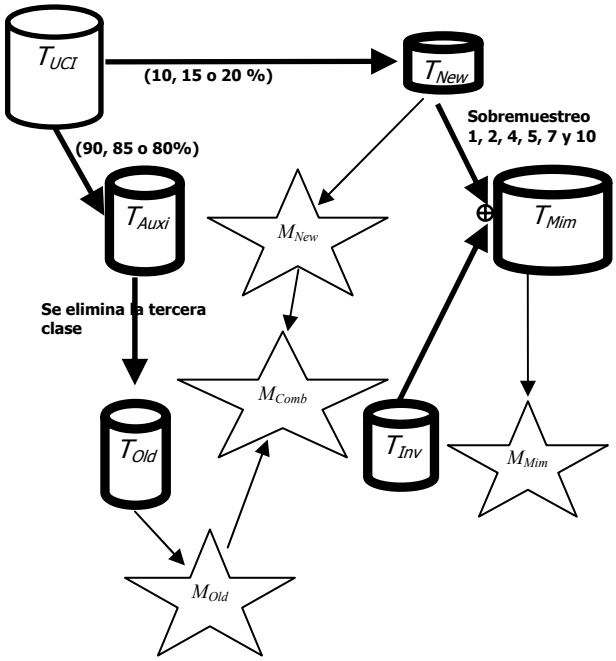


Figura 8.12. Esquema de los conjuntos de datos y modelos empleados en los experimentos cuando aparece una nueva clase.

El conjunto de datos original del repositorio UCI lo definimos como T_{UCI}. Creamos aleatoriamente el conjunto de datos nuevo (T_{New}) con diferentes porcentajes (10, 15 o 20) del tamaño de T_{UCI}. Con los datos que no fueron elegidos para el conjunto de datos nuevo forman el conjunto de datos auxiliar T_{Auxi}. A este conjunto auxiliar se le elimina la tercera clase y así generamos el conjunto de datos antiguo T_{Old}. El conjunto de datos inventados (T_{Inv}) se genera con un muestreo uniforme usando T_{Old} para obtener mínimos y máximos. El conjunto de datos de en-

trenamiento para los modelos miméticos (T_{Mim}) se crean uniendo T_{Inv} y T_{New} con sobremuestreo (1, 2, 4, 5, 7 y 10) de los ejemplares que son de la tercera clase.

La configuración del entrenamiento y validación de los modelos miméticos es la misma que mostramos en la Tabla 7.3.

Utilizaremos las medias de los resultados de 1000 a 5000, para obtener una visión global y así poder definir que pruebas estadísticas son necesarias realizar.

8.4.2. Resultados promedio

A continuación mostramos en tablas los resultados promedio obtenidos con los diferentes porcentajes experimentados. Asicomo, las gráficas de precisión contra los diferentes tamaños del conjunto de datos inventados, para mayor claridad se a eliminado la curva del MOld(Old).

Tabla 8.12. Resultados promedio usando un 10% de los datos para T_{New}.

Modelo	Precisión	MSE	Logloss	AUC
MComb	61.00	0.000	0.000	0.000
MMim1	61.02	0.431	148.288	0.599
MMim10	63.19	0.415	288.795	0.619
MMim2	62.09	0.423	204.238	0.607
MMim4	62.84	0.416	275.430	0.617
MMim5	62.71	0.418	284.442	0.614
MMim7	63.32	0.413	284.977	0.618
MNew	64.05	0.384	197.380	0.659
MOld(New)	54.88	0.502	387.076	0.500
MOld(Old)	83.08	0.253	35.446	0.500

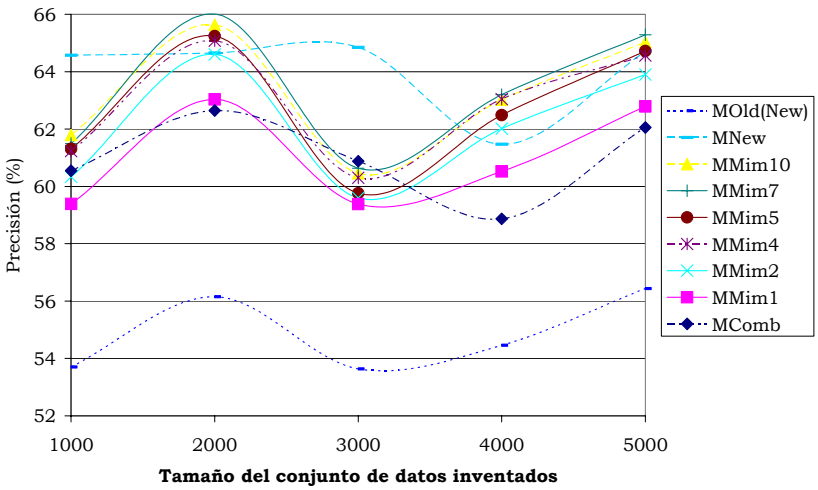


Figura 8.13. Resultados promedio usando un 10% de los datos para T_{New}.

Tabla 8.13. Resultados promedio usando un 15% de los datos para T_{New}.

Modelo	Precisión	MSE	Logloss	AUC
MComb	60.41	0.000	0.000	0.000
MMim1	62.02	0.446	138.228	0.618
MMim10	63.64	0.434	265.327	0.639
MMim2	63.24	0.436	205.831	0.631
MMim4	63.76	0.430	253.239	0.639
MMim5	63.62	0.432	259.472	0.638
MMim7	63.74	0.432	264.077	0.640
MNew	65.78	0.383	179.267	0.685
MOld(New)	54.94	0.504	387.596	0.500
MOld(Old)	83.24	0.253	35.158	0.500

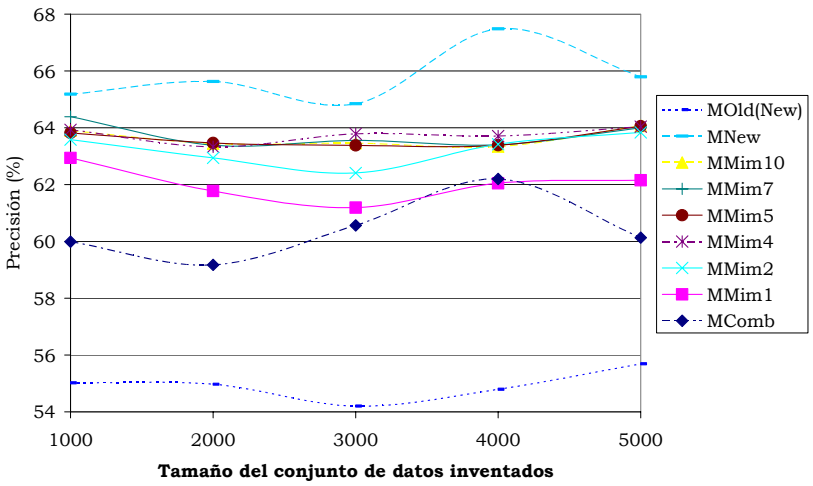


Figura 8.14 Resultados promedio usando un 15% de los datos para T_{New}.

Tabla 8.14. Resultados promedio usando un 20% de los datos para T_{New}.

Modelo	Precisión	MSE	Logloss	AUC
MComb	62.05	0.000	0.000	0.000
MMim1	62.86	0.459	119.058	0.664
MMim10	65.04	0.438	236.816	0.667
MMim2	64.49	0.442	187.086	0.678
MMim4	65.22	0.435	230.684	0.670
MMim5	65.16	0.436	233.562	0.670
MMim7	65.11	0.437	237.159	0.668
MNew	68.93	0.366	149.331	0.737
MOld(New)	55.19	0.503	384.517	0.500
MOld(Old)	83.26	0.254	35.331	0.500

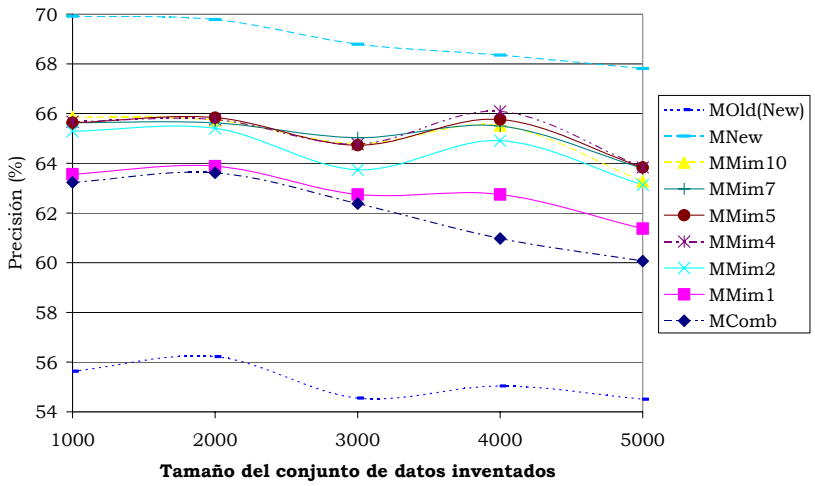


Figura 8.15 Resultados promedio usando un 20% de los datos para T_{New}.

Para estos conjuntos de datos se detecta la necesidad de la revisión en los diferentes casos, es decir, la precisión del MOld(Old) es mucho mayor que la de MOld(New).

El único caso donde el método mimético logra ganarle al los demás modelos es cuando el porcentaje es de 10 con un tamaño de datos inventados de 2000. En específico en los modelos miméticos con sobremuestreo mayor o igual a 4. Esto significa que la técnica mimética es útil en una etapa temprana del cambio de contexto, cuando el conjunto de datos nuevos tiene un tamaño pequeño.

8.4.3. Análisis de la significancia para la precisión

Así pues, emplearemos el caso global donde los miméticos ganan a los demás modelos. Es decir, emplearemos los modelos en el escenario del 10% de los datos para T_{New} con un tamaño de datos inventados de 2000 para realizar las pruebas estadísticas de significancia con respecto a la precisión.

La interpretación de la notación en la Tabla 8.15 de resultado para las pruebas t de *Student* es: el modelo de la fila (Gana/Empata/Pierde) al modelo de la columna. La variable medida en las pruebas t es el coste global.

Tabla 8.15 Pruebas t para modelos obtenidos usando un 25% de los datos para T_{New}.

	MOld(Old)	MOld	MComb	MNew	MMim1	MMim2	MMim4	MMim5	MMim7	MMim10
MOld(Old)	-	(4/1/0)	(4/1/0)	(3/2/0)	(4/1/0)	(4/1/0)	(4/1/0)	(4/1/0)	(4/1/0)	(4/1/0)
MOld	(0/1/4)	-	(0/4/1)	(0/3/2)	(0/3/2)	(0/3/2)	(0/3/2)	(0/3/2)	(0/2/3)	(0/2/3)
MComb	(0/1/4)	(0/5/0)	-	(0/4/1)	(1/3/1)	(1/2/2)	(1/2/2)	(1/2/2)	(1/1/3)	(1/1/3)
MNew	(0/2/3)	(2/3/0)	(1/4/0)	-	(2/2/1)	(0/4/1)	(0/4/1)	(0/4/1)	(1/3/1)	(0/4/1)
MMim1	(0/1/4)	(2/3/0)	(1/3/1)	(2/2/1)	-	(0/4/1)	(0/4/1)	(0/4/1)	(0/4/1)	(0/5/0)
MMim2	(0/1/4)	(2/3/0)	(2/2/1)	(1/4/0)	(1/4/0)	-	(0/5/0)	(0/5/0)	(0/4/1	(0/5/0)
MMim4	(0/1/4)	(2/3/0)	(2/2/1)	(1/4/0)	(1/4/0)	(0/5/0)	-	(0/5/0)	(0/4/1	(0/5/0)
MMim5	(0/1/4)	(2/3/0)	(2/2/1)	(1/4/0)	(1/4/0)	(0/5/0)	(0/5/0)	-	(0/4/1	(0/5/0)
MMim7	(0/2/3)	(2/3/0)	(3/1/1)	(1/3/1)	(1/4/0)	(1/4/0)	(1/4/0)	(1/4/0)	-	(0/5/0)
MMim10	(0/2/3)	(2/3/0)	(3/1/1)	(0/4/1)	(2/3/0)	(0/5/0)	(0/5/0)	(0/5/0)	(0/5/0)	-

De acuerdo a estas pruebas los miméticos mejores son a partir de que exista sobremuestreo. El conjunto de datos donde siempre ganan los miméticos al MNew es *hayes-roth*. Aquí queda demostrado que si existen casos donde los modelos miméticos utilizando sobremuestreo pueden ser útiles.

8.5. Conclusiones

Como hemos visto en este capítulo otro caso de revisión de modelos se da cuando el formato de los datos cambia. En específico cuando aparecen o se eliminan atributos o valores de clase.

Si en un árbol de decisión creado con el algoritmo J48 se eliminar un atributo muy importante, el árbol se convierte en una hoja. Y si se eliminar un atributo de baja importancia, este es sustituido por otro de similar importancia. Cuando se ponen nulos en la mitad de los valores de un atributo muy importante el árbol degenera en una hoja. El poner nulos a un atributo es restarle importancia tanto como si lo eliminámos del árbol. Cuando se ponen nulos en la mitad de los valores de un atributo poco importante otro atributo lo sustituye. Cuando al mismo tiempo de agregar nulos a un atributo importante y se adiciona un atributo auxiliar se observa que el atributo auxiliar se convierte en la raíz del árbol, es decir, se generan dos subárboles

uno para los valores conocidos y otro para los nulos. En el caso de agregar nulos a un atributo de baja importancia y adicionar un atributo auxiliar se observa que el atributo auxiliar sustituye al atributo de baja importancia.

En el caso de la revisión de modelos cuando aparece un nuevo atributo importante, es necesario que la importancia del atributo que aparece realmente sea muy importante. El mejor modelo mimético para este escenario es el MMim1, es decir basta con aplicar el esquema de modelo mimético sin emplear el sobremuestreo.

Para el caso de la revisión de modelo cuando aparece un nuevo valor de clase si se detecta la necesidad de la revisión. El único caso donde el método mimético logra ganarle al los demás modelos es cuando el porcentaje es de 10. En específico en los modelos miméticos con sobremuestreo mayor o igual a 2. Esto significa que la técnica mimética es útil en una etapa temprana del cambio de contexto, cuando el conjunto de datos nuevos tiene un tamaño pequeño.

Como trabajo futuro sería interesante estudiar la asignación de los valores del atributo principal con otros métodos diferentes al de la media.

Capítulo 9

Contextualización al coste

La mayoría de los modelos existentes (ya hayan sido aprendidos usando técnicas de minería de datos u otras técnicas semiautomáticas o bien elaborados por un experto) se desarrollan pensando en un contexto inicial de costes. Sin embargo, en el momento de su aplicación, o, transcurrido un cierto tiempo, el contexto de costes puede haber variado, por lo que incluso si el modelo sigue siendo válido en cuanto a aciertos, puede dejar de ser adecuado con respecto al coste.

Adicionalmente, tal y como mencionamos en el capítulo 6, en muchas aplicaciones reales se utilizan modelos de los que no se dispone de los datos de entrenamiento a partir de los cuales se derivó el modelo. Esto puede deberse a muchas razones: el modelo pudo haber sido construido por un experto humano sin aplicar ninguna técnica de aprendizaje computacional, o también puede ser que los datos usados para construir el modelo simplemente ya no existan por pérdida o daño de los mismos. Generalmente, porque ha pasado bastante tiempo. Ésta es una situación real en muchas áreas tales como la ingeniería, la diagnosis, los procesos de fabricación, los negocios, etc.

Por tanto, las técnicas de aprendizaje automático de minería de datos, aunque cada vez más útiles y frecuentes en la adquisición de conocimiento a partir de datos, no se pueden aplicar si tenemos ya un modelo que queremos adaptar o transformar, pero no disponemos de los datos originales.

En este capítulo, proponemos y evaluamos una familia de métodos para transformar un modelo previamente generado para que se adapte a una nueva situación de costes sin tener que volver a entrenarlo (en el caso de modelos de minería de datos) o volver a obtenerlo (en el caso de modelos obtenidos por expertos). Esta trasformación se realiza, por lo tanto, sin contar con los datos originales. Básicamente, la idea consiste en usar la técnica mimética para generar un modelo similar al inicial (oráculo) pero contextualizado al coste. Para ello, proponemos al menos seis formas de decidir la clasificación para disminuir el coste global del modelo mimético. Se proponen tres criterios de decisión de la clasificación (umbrales) y dos esquemas diferentes del mimético (sin contar con los datos originales) para obtener una disminución del coste global del modelo. Nos centramos en el estudio de problemas de clasificación binaria (sólo dos clases).

En un modelo predictivo el criterio de decisión consiste, generalmente, en predecir la clase con la probabilidad a posteriori más alta o, en problemas de dos clases, predecir la clase cuya probabilidad a posteriori supere el valor 0.5. Sin embargo, dado que las estimaciones de probabilidad son inexactas, no hay justificación real para tal criterio de decisión ni siquiera en el caso de que las clases estén balanceadas. Por lo tanto, el criterio de decisión debe ser un parámetro adicional al modelo y tiene que ser aprendido de los datos (Lachiche y Flach, 2003). En el caso de dos clases, el criterio de decisión consiste simplemente en establecer un umbral, que se puede aprender de diferentes formas. De esta forma, si la probabilidad a posteriori de alguna clase en un ejemplo excede al valor de un umbral entonces el ejemplo es de esa clase. En este capítulo estudiamos tres métodos de obtener este umbral en los modelos miméticos: directo, por ordenamiento y por análisis ROC, siendo estas dos últimas propuestas originales del autor de esta tesis. Estos tres métodos los usamos en dos nuevos esquemas propuestos del método mimético (uso de los costes antes o después de la mimetización).

El capítulo está organizado como sigue. En la sección 1 revisamos la literatura relacionada y relativa a la clasificación sensible al coste para el caso de dos clases. En la sección 2 presentamos la contextualización mimética, posibles aplicaciones y el esquema de aprendizaje. En la sección 3 describimos los algoritmos para determinar los diferentes umbrales a estudiar,

ejemplos explicativos de su aplicación, así como un análisis de sus propiedades y de las relaciones existentes entre ellos. En la sección 4 describimos los cuatro estilos propuestos de generación de datos inventados, los esquemas utilizados en este trabajo para el aprendizaje de los modelos miméticos así como una descripción de los modelos miméticos a estudiar. En la sección 5 describimos las diferentes configuraciones e incluimos la evaluación experimental efectuada y los resultados generales obtenidos, los cuales demuestran la bondad del método mimético para sensibilizar al coste modelos de caja negra, analizando qué configuración es mejor. En la sección 6 se realiza una discusión de la investigación realizada. Finalmente, la última sección se dedica a mostrar las conclusiones sobre los resultados obtenidos, dando un procedimiento para utilizar el método mimético en la contextualización a costes.

9.1. Trabajos relacionados

La solución tradicional al problema de sensibilizar clasificadores al coste es el análisis ROC. El problema del análisis ROC es que exceptuando algunos casos (árboles de decisión, por ejemplo), el cambio del criterio de decisión de la clasificación (umbral) puede dar como resultado un modelo no comprensible (Ferri et al., 2004). Además, el análisis ROC requiere de un conjunto de validación (o validación cruzada) extra para dibujar la curva. En la situación propuesta en este capítulo no tenemos datos de entrenamiento y por tanto, no podemos tener datos de validación.

Incluyendo los trabajos propios del análisis ROC, no existen demasiados trabajos similares al propuesto, aunque sí relacionados.

La primer forma de conseguir la sensibilización al coste de un modelo es el aprendizaje sensible al coste por ejemplos ponderados (*cost-sensitive learning by example weighting*) (Abe et. al., 2004). Los métodos que siguen esta técnica modifican la distribución de los ejemplos de entrenamiento antes de aplicar el método de aprendizaje, de modo que el clasificador aprendido a partir de la distribución modificada sea sensible al coste. El trabajo en esta área se remonta a tiempo atrás e incluye el método de la estratificación.

La estratificación (Breiman et. al., 1984; Chan y Stolfo, 1998) es una aproximación sencilla que se realiza por sobremuestreo (*oversampling*) o submuestreo (*undersampling*). Este método permite adaptar un algoritmo de aprendizaje a un contexto de coste sin modificar el propio algoritmo. Para ello se altera la frecuencia de las clases, repitiendo ejemplos de las clases que más afectan al coste (sobremuestreo), o eliminando ejemplos de las clases que menos afectan al coste (submuestreo).

Otro trabajo relacionado es el de (Elkan, 2001). Este autor prueba un teorema (que ya había sido usado por Domingos (1999) puesto que es trivial) que demuestra cómo cambiar la proporción de ejemplos en un conjunto de entrenamiento para tomar decisiones óptimas en la clasificación sensible al coste usando un clasificador aprendido por un método de aprendizaje estándar no sensible al coste. Además, recomienda usar las probabilidades estimadas del clasificador para estimar las decisiones óptimas. La predicción óptima para un ejemplo x es la clase i que minimiza

$$L(x,i) = \sum_j P(j \mid x) C(i,j) \tag{9.1}$$

donde $P(j|x)$ es la probabilidad condicional de cada clase j dado el ejemplo x, $C(i,j)$ es la entrada en la matriz de costes C que define el coste de predecir la clase i cuando la clase verdadera es j, tal que si $i=j$ entonces la predicción es correcta, mientras que si $i \neq j$ la predicción es incorrecta.

En problemas binarios (donde las dos clases se denominan 0 y 1) la predicción óptima será la clase 1 si y sólo si el coste esperado de esta predicción es menor o igual al coste esperado de predecir la clase 0, es decir si y solamente si

$$\begin{aligned} & P(0 \mid x) \cdot C(1,0) + P(1 \mid x) \cdot C(1,1) \\ & \leq P(0 \mid x) \cdot C(0,0) + P(1 \mid x) \cdot C(0,1) \end{aligned}$$

Lo cual es equivalente a

$$(1-p) \cdot C(1,0) + p \cdot C(1,1) \leq (1-p) \cdot C(0,0) + p \cdot C(0,1)$$

siendo $p = P(1 \mid x)$. Si esta desigualdad se convierte en igualdad, entonces la predicción de cada clase es óptima.

Entonces, el umbral para realizar decisiones óptimas es una probabilidad p^* tal que satisface la expresión

$$(1-p^*)\cdot C(1,0)+p^*\cdot C(1,1)=(1-p^*)\cdot C(0,0)+p^*\cdot C(0,1)$$

Asumiendo condiciones razonables (es decir, $C(1,0)>C(0,0)$ y $C(0,1)>C(1,1)$) la predicción óptima es la clase 1 si y solo si $p \geq p^*$. Si despejamos p* de la ecuación anterior obtenemos el umbral

$$p^*=\frac{C(1,0)-C(0,0)}{C(1,0)-C(0,0)+C(0,1)-C(1,1)} \qquad (9.2)$$

Un tercer trabajo relacionado es el algoritmo *MetaCost* (Domingos, 1999) el cual emplea una combinación de clasificadores (basada en *bagging*) para reetiquetar los ejemplos de entrenamiento. Las estimaciones de probabilidad de las clases se realizan aprendiendo clasificadores múltiples y, por cada ejemplo, usa la fracción del voto total para cada clase como estimación de su probabilidad. Después usa la fórmula (9.1) para reetiquetar cada ejemplo de entrenamiento con la clase que minimice el coste. Este conjunto reetiquetado se emplea para aprender un nuevo clasificador sensible al coste.

Por su parte, el algoritmo *costing* (Zadrozny et. al, 2003) es un algoritmo de meta-aprendizaje para sensibilizar al coste un algoritmo de aprendizaje, parecido a *MetaCost*. El algoritmo emplea también una combinación de clasificadores (pero basada en *boosting*). Este algoritmo altera la distribución original de los ejemplos de la siguiente forma: el número de veces que se repite un ejemplo viene dado por un factor proporcional al coste relativo de cada ejemplo y luego se aplica a cada ejemplo un criterio de pertenencia para formar parte del conjunto de ejemplos de entrenamiento. Esta aproximación tuvo buenos resultados en dos conjuntos de datos. El problema de *costing* es que no es determinista la elección del criterio de aceptación de ejemplares de entrenamiento.

Finalmente, el trabajo más relacionado es el de (Lachiche y Flach, 2003). Su idea es modificar el umbral del criterio de clasificación una vez aprendido el modelo para maximizar la precisión o minimizar el coste. Los autores proponen y evalúan un algoritmo para encontrar ese umbral en una curva ROC dada. La curva ROC se construye en base a las probabilidades esti-

madas del conjunto de instancias. Con su aproximación Lachiche y Flach consiguen mejorar los resultados de los clasificadores (por ejemplo, Naive Bayes) incluso cuando la distribución de clases y los costes son los mismos que en el entrenamiento. Esto muestra que la mayoría de las técnicas de aprendizaje y de la minería de datos hacen la elección del umbral mal, de una manera local y no global. Un primer problema con la aproximación de Lachiche y Flach es que utiliza un conjunto de validación. Un segundo problema de este algoritmo es su tiempo de ejecución pues busca el mejor umbral de entre todas las probabilidades estimadas, es decir, tiene que probar todos los posibles umbrales obtenidos de cada probabilidad estimada. El método que proponemos solamente calcula un umbral que logra minimizar el coste.

En resumen, aunque existen muchas aproximaciones, se pueden dividir entre aquellas que se basan en la modificación de un umbral tras el entrenamiento y aquellas que se basan en ponderar (repetir o eliminar) los ejemplos antes del entrenamiento.

La aproximación por ejemplos ponderados es muy general puesto que reutiliza clasificadores arbitrarios y no requiere estimaciones exactas de la probabilidad de las clases. El método de sobremuestreo lo incluimos como referencia y nos servirá para comparar con el resto de métodos basados en umbral.

Como veremos, una diferencia general de todos los métodos anteriores y nuestros métodos es que los primeros requieren conocer los datos de entrenamiento; mientras que en nuestro caso los inventamos.

9.2. Contextualización mimética

En general, como ya dijimos al principio del capítulo, disponemos de un modelo inicial que fue aprendido usando técnicas de minería de datos u otras técnicas semiautomáticas o bien fue elaborado por un experto. No hacemos ninguna suposición sobre si el contexto inicial del aprendizaje se utilizó o no una matriz de costes. Sin embargo, ahora supondremos que el contexto ha cambiado y que se requiere considerar diferentes costes de los errores en la clasificación. En este escenario o contexto actual no tenemos los datos de entreamiento originales.

Por tal motivo inventaremos datos aleatorios que permitan capturar las fronteras del dominio del problema representado por el modelo inicial. La contextualización mimética (ver Figura 9.1) que proponemos nos permite cambiar el contexto del modelo inicial (oráculo) para que sea sensible al coste, adaptándose al nuevo contexto y expresándolo en forma de reglas.

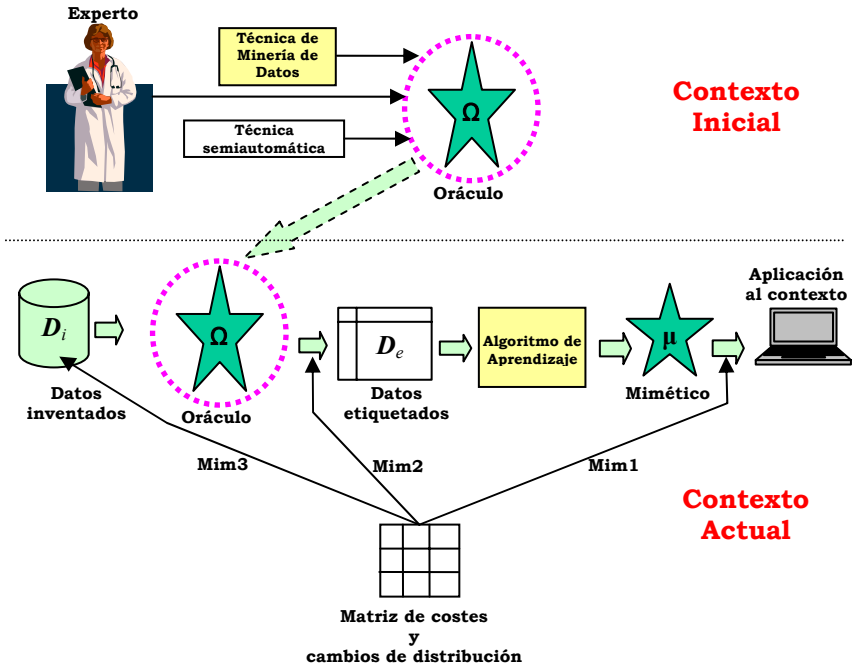


Figura 9.1. La contextualización mimética.

Las principales ventajas de nuestra propuesta son no requerir el reentrenamiento del modelo inicial y no es necesario conocer los datos originales (si los hubo en su caso). De los datos originales tan solo necesitamos disponer de los valores máximos y mínimos de sus atributos.

A partir de estos valores máximos y mínimos y aplicando la distribución uniforme podemos obtener un conjunto de datos inventados, que luego etiquetamos con el oráculo. Los datos inventados o los datos etiquetados o el umbral del mimético se pueden transformar usando la información de una matriz de costes. En la siguiente sección describimos diferentes métodos miméticos para realizar esta transformación, ya sea a través de umbrales o con sobremuestreo, que dependen fundamental-

mente del momento en el que se usa la información de costes (en la Figura 9.1, se han denotado como Mim1, Mim2 y Mim3).

En este nuevo contexto, a partir de los costes para las predicciones correctas e incorrectas, nuestro objetivo es predecir la clase de un ejemplo obteniendo el coste esperado más bajo.

En nuestro estudio se puede utilizar cualquier matriz de costes *C*. Una matriz de costes genérica como la siguiente:

		Real 1	Real 0
Predicho	1	$C(1,1)$	$C(1,0)$
	0	$C(0,1)$	$C(0,0)$

se puede transformar en:

		Real 1	Real 0
Predicho	1	0	1
	0	*skew*	0

(9.3)

donde

$$skew = \frac{C(0,1) - C(1,1)}{C(1,0) - C(0,0)}$$

Como hemos asumido una condición razonable (es decir, $C(1,0)>C(0,0)$ y $C(0,1)>C(1,1)$) el *skew* es un número real positivo.

Obsérvese que aunque típicamente el *skew* no solo depende de la matriz de costes sino también de la distribución de clases (ver en el capítulo 2 la subsección 4.2), en nuestro caso solo consideraremos los costes. De esta forma, las predicciones correctas no tienen coste y en las predicciones incorrectas el coste de un positivo falso es 1 mientras el coste de un negativo falso es *skew*. Por lo tanto, *skew* nos indica la importancia relativa en coste de los ejemplos negativos frente a los positivos. De aquí en adelante emplearemos la clase 1 como sinónimo de la clase positiva y la clase 0 como sinónimo de la clase negativa.

9.3. Determinación del umbral

Como hemos visto en las diferentes aproximaciones basadas en umbral mencionadas en la sección 9.1, el problema crucial es determinar un umbral que consiga minimizar el coste global. A diferencia de la propuesta de Flach y Lachiche, queremos un método que no requiera un conjunto de validación. En esta situación, si no tenemos un conjunto de validación presentamos tres opciones para determinar el umbral: 1) no suponer nada sobre la estimación de probabilidades y actuar como si estuvieran distribuidas uniformemente, 2) asumir que las probabilidades estimadas están bien ordenadas (es decir, a mayor probabilidad estimada, mayor probabilidad real) y 3) suponer que las probabilidades estimadas son equivalentes a las frecuencias relativas de las clases en el conjunto de datos de prueba (probabilidades calibradas). Estas tres visiones dan lugar a tres métodos diferentes de determinar el umbral: método directo (Dir), método por ordenamiento (Ord) y método basado en el análisis ROC (ROC). A continuación pasaremos a describir cada uno de ellos.

9.3.1. Umbral directo

El primer método, como hemos dicho, no ha de asumir nada, en principio, sobre las probabilidades. En este caso podemos derivar el umbral que minimice la ecuación (9.1) siguiendo a (Elkan, 2001). La forma más directa de definir este umbral es sustituir los valores de la matriz de costes (9.3) en la ecuación (9.2), con lo que tenemos:

$$p^* = \frac{1-0}{1-0+skew-0}$$

Simplificando

$$p^* = \frac{1}{1+skew}$$

Definimos el $Threshold_{Dir}$ como el umbral para que la decisión óptima sea la clase 0. Así, se debe cumplir

$$Threshold_{Dir} = 1 - p^*$$

Sustituimos el valor de p*

$$Threshold_{Dir} = 1 - \frac{1}{1 + skew}$$

Y operando, finalmente queda:

$$Threshold_{Dir} = \frac{skew}{1 + skew}$$

De esta forma las instancias se clasifican de clase 0 si y sólo si la probabilidad estimada de la clase 0 es mayor o igual que $Threshold_{Dir}$.

Obsérvese que aunque hemos partido de las probabilidades para deducir la ecuación del umbral, éste depende únicamente del valor *skew* y no de las probabilidades. Esto se debe a que la fórmula de Elkan saca el máximo local para cada ejemplo, pero no saca el valor óptimo global. En realidad, desde el punto de vista global, esta técnica actúa como si las probabilidades estuvieran uniformemente distribuidas (situación que no es frecuente ni conveniente).

Los siguientes ejemplos ilustran el cálculo del umbral directo para diferentes *skews*.

skew	$Threshold_{Dir}$
1	0.5
2	0.66
3	0.75
5	0.83
10	0.9091

Como puede observarse cuando el *skew* es 1, es decir, cuando el coste de los ejemplos de clase 0 es igual al de los de la clase 1, el umbral es 0.5, que coincide con el criterio de decisión utilizado en los modelos predictivos.

9.3.2. Umbral por ordenamiento

El umbral directo no toma en cuenta las probabilidades estimadas, tan sólo el *skew*. Para tomar en cuenta el orden de estas probabilidades y el valor del *skew* proponemos el cálculo de un nuevo umbral más completo, el umbral por ordenamiento.

El umbral por ordenamiento ($Threshold_{Ord}$) es la probabilidad estimada tal que permita obtener la proporción $1/(1+skew)$ de clase 0 en el conjunto de datos inventados.

Los datos inventados son generados por medio de una distribución uniforme, la idea es, a través de ellos poder imitar y capturar el dominio de un oráculo. Cada ejemplo inventado tiene una probabilidad estimada de pertenencia a alguna clase, esta es asignada por el oráculo. A continuación, ordenamos decrecientemente de acuerdo a la probabilidad de pertenencia a la clase 0. Luego, obtenemos el umbral por ordenamiento encontrando la probabilidad que permite dividir este vector de probabilidades de acuerdo a la proporción deseada.

En la Figura 9.2 hemos representado un vector de probabilidades estimadas desde la máxima probabilidad hasta la mínima probabilidad. Los índices del vector van de 0 hasta n-1, donde n es el número de ejemplares del conjunto de datos inventados. El vector se divide de acuerdo a los costes en dos segmentos, uno que tenga la proporción de $1/(skew+1)$ ejemplos de clase 0 y el otro con la proporción $(skew/(skew+1))$ de ejemplos. En el punto de división de los dos segmentos se encuentra el valor del umbral buscado, en la posición *Pos*, posiblemente un número real, debido a que este punto no tiene por qué coincidir necesariamente con un índice de algún elemento del vector. Así pues, para obtener el valor del umbral hacemos uso de los índices más cercanos a *Pos*, los índices a y b y de sus probabilidades p_a y p_b.

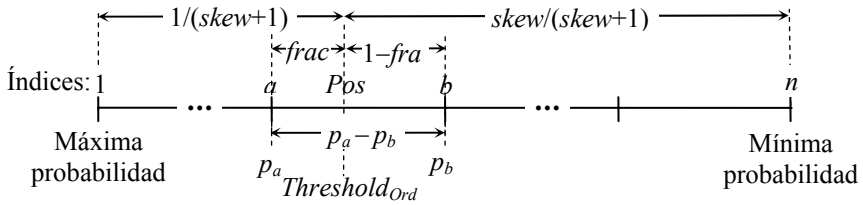


Figura 9.2 Esquema de un vector de probabilidades ordenado para explicar el cálculo del $Threshold_{Ord}$.

De la Figura 9.2 podemos obtener el umbral como p_a menos la fracción de probabilidad existente entre p_a y el $Threshold_{Ord}$ (*frac*), matemáticamente:

$$Threshold_{Ord} = p_a - (p_a - p_b) \cdot frac$$

O análogamente, tomando como base la probabilidad p_b, el umbral viene determinado por, p_b más la fracción de probabilidad existente entre p_b y el $Threshold_{Ord}$ (1-*frac*). Matemáticamente:

$$Threshold_{Ord} = p_b + (p_a - p_b) \cdot (1 - frac) \qquad (9.4)$$

Formalmente, sea $P = p_1, p_2, \ldots, p_n$ una secuencia de probabilidades ordenadas decrecientemente, es decir, $p_i \in P$ e $i \in \{1..n\}$, tal que $p_i > p_{i+1}$. Entonces, el umbral $Threshold_{Ord}$ se encuentra en una posición *Pos*, entre dos probabilidades consecutivas p_a y p_b tales que:

- La posición del umbral es:

$$Pos = \frac{n-1}{skew + 1} + 1 \qquad (9.5)$$

- la posición de la probabilidad p_a es:

$$a = Lower(Pos)$$

  donde *Lower*(*t*) es una función que devuelve la parte entera de *t*. Por ejemplo, *Lower*(6.2) = 6.

- la posición de la probabilidad p_b es:

$$b = Upper(Pos)$$

  donde *Upper*(*t*) es una función que devuelve el entero más próximo mayor o igual a *t*. Por ejemplo, *Upper*(6.2) = 7.

- La distancia entre la posición del umbral (*Pos*) y la posición de p_a viene dada por:

$$frac = Pos - a \tag{9.6}$$

siguiendo con el ejemplo, en este caso *frac* vale 0.2.

Para un caso concreto, reemplazamos todos estos valores en la ecuación (9.4) podemos calcular el valor del umbral. De esta forma, las instancias se clasifican de clase 0 si y sólo si la probabilidad estimada de la clase 0 es mayor que $Threshold_{Ord}$.

Para que el umbral anterior ajuste bien el porcentaje de ejemplos de una clase y otra, hay que tener en cuenta que es posible que existan probabilidades estimadas que sean iguales al valor del umbral. En este caso deberemos repartir también de forma proporcional los ejemplares, es decir, cuántos ejemplos con probabilidad igual al $Threshold_{Ord}$ deben etiquetarse de clase 0 y cuántos de clase 1. Para realizar este reparto en el momento de aplicar el umbral emplearemos un índice de retención por repetición (*Ro*). Nótese que de acuerdo a la fórmula (9.5) esta situación no afecta al cálculo de la posición del umbral, ya que ésta sólo depende del *skew* y del número de ejemplos.

Dada $P = p_1, p_2, \ldots, p_n$ una secuencia de probabilidades ordenadas decrecientemente, tales que p_a, es la probabilidad inmediatamente anterior al umbral ($Threshold_{Ord}$). Sea $P' = p_{a-r+1}, p_{a-r+2}, \ldots, p_{a-1}, p_a, p_{a+1}, \ldots, p_{a+s}$ la subsecuencia de probabilidades repetidas del umbral, es decir, $p_j \in P'$ y $j \in \{a-r+1..a+s\}$, tal que $p_j = Threshold_{Ord}$. Entonces, definimos el índice de retención como

$$Ro = \frac{r}{r+s} \tag{9.7}$$

donde r es el número de probabilidades anteriores a p_a incluida esta y s es el número de probabilidades posteriores a p_a. Nótese que *Ro* es un número comprendido entre 0 y 1. Como se deduce de la definición anterior el índice *Ro* indica la proporción de probabilidades iguales al $Threshold_{Ord}$ que ocupan una posición anterior al mismo. Dado que estamos considerando que las probabilidades están ordenadas decrecientemente, en esta aproximación por ordenamiento una vez determinada la posición del umbral, cualquier ejemplo cuya probabilidad esté situada en una posición anterior a la del umbral es clasificada como de clase 0 (ya que esto significa que dicha probabilidad es mayor que el umbral). La idea de aplicar este mismo criterio

cuando la probabilidad del ejemplo es igual al umbral, pero sacando como posición el índice *Ro*. Para ello procedemos de la siguiente forma: para cada ejemplo con probabilidad igual al $Threshold_{Ord}$ generamos un número aleatorio comprendido entre 0 y 1, que representa la posición relativa de dicho ejemplo con respecto a todos los ejemplos de igual probabilidad, entonces, si dicho número es ≥ *Ro* se predice la clase 0 y, en caso contrario, la clase 1.

Véase el algoritmo de la función en la Figura 9.3 para una explicación determinista de la forma de proceder en la aplicación del umbral y *Ro* a una probabilidad p_i.

```
FUNCIÓN PredecirClase (Threshold_Ord, Ro, p_i)
  SI p_i = Threshold_Ord ENTONCES
      na = generar un número aleatorio entre 0 y 1
      SI na ≥ Ro ENTONCES
          Clase=0
      SI NO
          Clase=1
  SI NO
      SI p_i > Threshold_Ord ENTONCES
          Clase=0
      SI NO
          Clase=1
  REGRESAR Clase
FIN-FUNCIÓN
```

Figura 9.3. Función para predecir la clase de un ejemplo dados el umbral por ordenamiento ($Threshold_{Ord}$), el índice de retención (*Ro*) y su probabilidad estimada p_i.

El siguiente ejemplo ilustra la aproximación basada en ordenamiento que acabamos de describir

Ejemplo. Consideremos las probabilidades de la clase 0 de 8 ejemplos (n=8), con los siguientes valores ya ordenados descendentemente:

1	1	1	1	0.9	0.6	0.4	0.1

Vamos a calcular los umbrales para los valores del *skew* 1, 2, 3, 5 y 10. Para cada uno de estos valores determinamos el valor *Pos* con la fórmula (9.5). Luego calculamos los valores *a*, *b*, y *frac*, y finalmente, aplicamos la fórmula del $Threshold_{Ord}$ (9.4). Los resultados obtenidos se muestran en la siguiente tabla.

skew	*Pos*	*a*	*frac*	p_a	p_b	$Threshold_{Ord}$
1	4.50	4	0.50	1	0.9	0.95
2	3.33	3	0.33	1	1	1
3	2.75	2	0.75	1	1	1
5	2.17	2	0.17	1	1	1
10	1.64	1	0.64	1	1	1

Consideremos ahora que las probabilidades de la clase 0 de 8 ejemplares (n=8), son los siguientes valores ya ordenados descendentemente:

0.8	0.6	0.6	0.6	0.6	0.6	0.4	0.3

Como puede observarse, hay cinco ejemplos con la misma probabilidad (0.6).

Calculamos el umbral, únicamente para el valor 2 del *skew*.

skew	*Pos*	*a*	*frac*	p_a	p_b	$Threshold_{Ord}$
2	4.33	4	0.33	0.6	0.6	0.6

En este caso, el umbral coincide con la probabilidad que se repite, por lo que necesitamos calcular el índice de retención *Ro*. Aplicando la ecuación (9.8) y teniendo en cuenta que r=2 y s=3, obtenemos

$$Ro = \frac{2}{2+3} = 0.4$$

Si aplicamos el umbral y el índice *Ro* obtendremos las siguientes clases predichas para los 8 ejemplos:

i	1	2	3	4	5	6	7	8
p_i	0.8	0.6	0.6	0.6	0.6	0.6	0.4	0.3
clase	0	0	1	1	1	0	1	1

En el caso de los índices 1, 7 y 8 se aplica el criterio del umbral: si $p_i > 0.6$ entonces clase=0 sino clase=1. Para las probabilidades iguales a 0.6 (es decir, en los índices del 2 al 6) hemos generado números aleatorios entre 0 y 1 tales que 2 ejemplo se han clasificado de clase 0 y 3 de clase 1, Con esto hemos logrado obtener un porcentaje del 37.5% (3/8·100) de

ejemplos de clase 0, siendo éste el porcentaje más cercano al deseado que es del 33.3% (1/(*skew*+1)·100=1/(2+1)·100).

A diferencia del método directo, con este método hemos considerado el vector de probabilidades estimadas y sus valores en el cálculo del umbral. Sin embargo, observemos que el intervalo de los valores de probabilidad estimada limitan el umbral. Por ejemplo, si tenemos un modelo que sólo estima probabilidades en el intervalo [0.5,0.6], el umbral sólo tomará valores en ese intervalo para cualquier *skew*. El tercer método propuesto pasamos a describir a continuación no presenta esta limitación.

9.3.3. Umbral ROC

Como ya dijimos, el cálculo del umbral directo depende tan sólo del *skew*, ignorando completamente cómo son las probabilidades estimadas por el modelo. En cambio el método por ordenamiento, aunque sí que contempla las probabilidades estimadas, está demasiado limitado por ellas, ya que estas probabilidades marcan los límites entre los cuales está el umbral.

Por tanto consideraremos otra forma alternativa de calcular el umbral que permita contemplar los valores de las probabilidades. En este caso, supondremos que el oráculo está calibrado y, por lo tanto, que genera probabilidades calibradas. Por ejemplo, si el oráculo asigna a un ejemplo una probabilidad estimada de ser de la clase 1 de 0.8, esto significa que si tuviésemos 100 ejemplos, el 80% serían de la clase 1 y el 20% serían de la clase 0. En el espacio ROC lo representaremos como un segmento que iría del punto (0,0) al (20,80) con una pendiente de 80/20=4.

Bajo esta suposición, para calcular el nuevo umbral presentamos una versión modificada de la curva ROC, que llamamos NROC. Esta nueva curva se basa en probabilidades calibradas, lo que permite utilizar una probabilidad como un porcentaje de las instancias que son de esa clase. Si tenemos dibujada la curva NROC y luego hacemos el corte en la inclinación adecuada de acuerdo al *skew*, obtendremos un punto que definirá el umbral ROC ($Threshold_{ROC}$).

Veamos primero cómo se construye una curva NROC.

Sea P^c una secuencia de probabilidades estimadas de la clase *c* ordenadas descendentemente, donde *c* puede tomar el valor de 0 ó 1. Sea p_i^c el elemento del P^c, que corresponde a la probabilidad estimada de la clase *c*, en el ejemplo *i*. El índice i puede ser un número entre 1 y el total de ejemplos (n). En un ejemplo i las probabilidades de sus clases son complementarias y están normalizadas, por lo tanto, $p_i^1+p_i^0=1$.

Definición 1. *La suma de las primeras j-ésimas probabilidades estimadas de la clase c es:*

$$Sum_j^{\ c} = \sum_{i=1}^{j} p_i^c$$

donde $1 \le i \le n$ *y* $1 \le j \le n$.

Sea

$$Sum_0^{\ c} = 0$$

Por simplicidad decimos que

$$Sum_n^{\ c} = Sum^c$$

Proposición 1. $j = Sum_j^1+Sum_j^0$.

Demostración. Trivial, ya que

$$Sum_j^{\ 1} + Sum_j^{\ 0} = \sum_{i=1}^{j} p_i^1 + \sum_{i=1}^{j} p_i^0$$
$$= \sum_{i=1}^{j} (p_i^1 + p_i^0)$$

Dado que $p_i^1+p_i^0=1$, si sustituimos este valor en la ecuación anterior, tenemos

$$Sum_j^{\ 1} + Sum_j^{\ 0} = \sum_{i=1}^{j} 1$$

Luego entonces

$$Sum_j^{\ 1} + Sum_j^{\ 0} = j$$

□

Definición 2. *Consideramos el espacio NROC definido por el eje de las abscisas desde* 0 *a* Sum^0*, y el eje de ordenadas de* 0 *a* Sum^1*. Cada punto de la curva NROC está definido por sus coordenadas* (Sum_i^0, Sum_i^1)*. El primer punto de la curva NROC es el punto* (*0,0*)*. Luego, se normalizar el espacio NROC, para que esté entre los puntos* (0,0) *y* (1,1)*. Los puntos se normalizan dividendo los valores de las abscisas entre* Sum^0 *y las ordenadas entre* Sum^1*. Finalmente, el punto i de la curva NROC normalizada se define como:*

$$\left(\frac{Sum_i^0}{Sum^0}, \frac{Sum_i^1}{Sum^1}\right)$$

donde $1 \leq j \leq n$*. Finalmente, para dibujar la curva NROC se conectan en secuencia los puntos usando segmentos de recta.*

Por cuestión de cálculo conviene expresar los puntos en función de solamente las probabilidades estimadas de los ejemplos de ser de la clase 0. Así, por la Proposición 1 cambiamos el numerador de la ordenada (Sum_i^1) por $i–Sum_i^0$, de igual forma el denominador de la ordenada (Sum_i^1) se cambia por $n–Sum^0$ resultando:

$$\left(\frac{Sum_i^0}{Sum^0}, \frac{i - Sum_i^0}{n - Sum^0}\right)$$

Ejemplo. Veamos un ejemplo para ilustrar la Definición 2, supongamos que tenemos 4 instancias (n=4) con sus probabilidades estimadas de ser de la clase 0, ya ordenadas:

1	0.9	0.7	0.4

De aquí Sum^1=1 y Sum^0=3.

Calculamos los puntos de la curva NROC sin normalizar:

(0,0)	(0,1)	(0.1,1.9)	(0.4,2.6)	(1,3)

Normalizando estos puntos obtendríamos:

(0,0)	(0,0.333)	(0.1,0.633)	(0.4,0.866)	(1,1)

Obtenemos la curva NROC dibujando los segmentos entre estos puntos y el resultado es la figura siguiente:

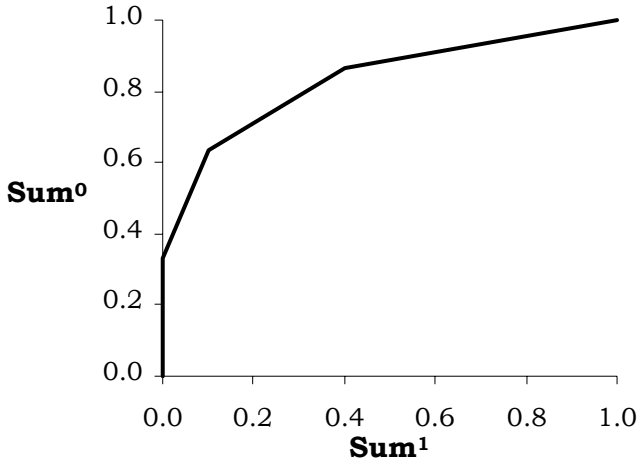


Figura 9.4. Ejemplo de curva NROC.

Una vez definida la curva NROC, veamos cómo podemos estimar umbrales a partir de ella.

Hay que normalizar el *skew* porque hemos normalizado el espacio ROC. En la Figura 9.5 mostramos gráficamente esta normalización, observamos que el espacio ROC rectangular original se convierte a un espacio ROC cuadrado:

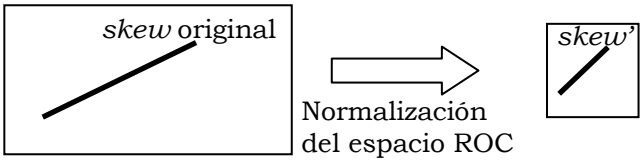


Figura 9.5. Normalización del espacio ROC.

Finalmente, *skew'* se define como:

$$skew' = skew \cdot \frac{Sum^0}{Sum^1}$$

Si el *skew'* es exactamente paralelo a un segmento de la curva, el umbral debe corresponder al ejemplo que "ha generado ese segmento". Es decir, si $skew' = p_i/(1-p_i)$ el umbral debe ser p_i.

Operando con estas dos últimas condiciones obtenemos:

$$Threshold_{ROC} = \frac{skew'}{1 + skew'}$$

Y ateniendo a la relación entre *skew'* y *skew*, se cumple

$$Threshold_{ROC} = \frac{skew \cdot \frac{Sum^0}{Sum^1}}{1 + skew \cdot \frac{Sum^0}{Sum^1}}$$

dividiendo numerador y denominador por $skew \cdot Sum^0 / Sum^1$ tenemos finalmente:

$$Threshold_{ROC} = \frac{1}{1 + \frac{1}{skew \cdot \frac{Sum^0}{Sum^1}}}$$

$$Threshold_{ROC} = \frac{1}{1 + \frac{1}{skew} \cdot \frac{Sum^1}{Sum^0}} \qquad (9.8)$$

De esta forma las instancias se clasifican de clase 0 si y sólo si la probabilidad estimada de la clase 0 es mayor o igual a $Threshold_{ROC}$.

El umbral ROC no tiene en cuenta el orden de las probabilidades, sino que sólo tiene en cuenta la suma de las probabilidades.

En la fórmula (9.7) se observa que:

1) si $skew=0$ entonces $Threshold_{ROC} \rightarrow 0$.

2) si $skew \rightarrow \infty$ entonces $Threshold_{ROC} \rightarrow 1$

Ejemplo. Las probabilidades de la clase 0 de 8 ejemplos (n=8) con los siguientes valores ordenados descendentemente:

1	1	1	1	0.9	0.6	0.4	0.1

Calculemos los umbrales para valores del *skew* de 1, 2, 3, 5 y 10.

Para el cálculo procedemos siguiendo la fórmula (9.8). Primero realizamos el cálculo de las sumas: Sum^1=2 y Sum^0=6, y después utilizamos la fórmula del $Threshold_{ROC}$. Los resultados obtenidos para los diferentes *skews* se muestran en la siguiente tabla:

skew	$Threshold_{ROC}$
1	0.7500
2	0.8571
3	0.9000
5	0.9375
10	0.9677

Ahora pasemos a estudiar las propiedades de los umbrales estudiados.

9.3.4. Propiedades de los tres métodos para calcular el umbral.

Una vez definidos estos tres umbrales diferentes, vamos a analizar cómo son sus límites máximos y mínimos, así como la relación entre sus valores.

Primero, analicemos sus límites máximo y mínimo. El valor máximo del umbral se presenta con un *skew* muy alto tendiendo a infinito.

Máximo ($skew=\infty$)

$$Threshold_{Dir} \rightarrow 1$$

$$Threshold_{ROC} \rightarrow 1$$

Para el $Threshold_{Ord}$ el límite superior no es 1 necesariamente, sino que viene marcado por el valor del ejemplo con probabilidad más alta.

El valor mínimo del umbral se encuentra cuando el *skew* toma un valor de 0.

Mínimo (*skew*=0)

$$Threshold_{Dir} = 0$$

$$Threshold_{ROC} = 0$$

Para $Threshold_{Ord}$ el límite inferior no es 0 necesariamente, sino que viene marcado por el valor del ejemplo con probabilidad más baja.

A continuación, comparemos los valores que se obtienen con los tres métodos. Está claro que siempre podemos encontrar casos donde $Threshold_{Dir} > Threshold_{Ord}$, y a la inversa, ya que $Threshold_{Ord}$ está limitado por los valores de las probabilidades de la clase 0 de los ejemplos. Lo mismo sucede entre $Threshold_{ROC}$ y $Threshold_{Ord}$.

Ejemplo. Veamos estos casos con un ejemplo. Consideremos el siguiente vector de probabilidades:

1	0.85	0.7	0.5	0.3	0.1

Y los valores del *skew* de 1, 2, 3, 5 y 10.

Los umbrales definidos por cada método son:

skew	$Threshold_{Dir}$	$Threshold_{Ord}$	$Threshold_{ROC}$
1	0.5000	0.6000	0.5750
2	0.6667	0.7500	0.7302
3	0.7500	0.8125	0.8023
5	0.8333	0.8750	0.8712
10	0.9091	0.9318	0.9312

Como puede verse, se cumple que $Threshold_{Dir} < Threshold_{Ord}$, $Threshold_{ROC} < Threshold_{Ord}$ y $Threshold_{Dir} < Threshold_{ROC}$.

Ejemplo. Sea ahora el siguiente vector de probabilidades:

0.6	0.35	0.2	0.13	0.1	0.05

Y los valores del *skew* de 1, 2, 3, 5 y 10.

En este caso los umbrales son:

skew	$Threshold_{Dir}$	$Threshold_{Ord}$	$Threshold_{ROC}$
1	0.5000	0.1650	0.2383
2	0.6667	0.2500	0.3849
3	0.7500	0.3125	0.4842
5	0.8333	0.3917	0.6101
10	0.9091	0.4864	0.7578

Como se observa, para todos los valores del skew considerados, se cumple que $Threshold_{Dir} > Threshold_{Ord}$, $Threshold_{ROC} > Threshold_{Ord}$ y $Threshold_{Dir} > Threshold_{ROC}$.

Sin embargo, la relación entre $Threshold_{ROC}$ *y* $Threshold_{Dir}$ depende de la relación entre Sum^1 *y* Sum^0, como muestra la siguiente proposición:

Proposición 2. *Si* $Sum^0/Sum^1 > 1$ *entonces* $Threshold_{ROC} > Threshold_{Dir}$, *si* $Sum^0/Sum^1 < 1$ *entonces* $Threshold_{ROC} < Threshold_{Dir}$, *y finalmente si* $Sum^0/Sum^1 = 1$ *entonces* $Threshold_{ROC} = Threshold_{Dir}$.

Demostración. La demostración es trivial y la haremos por contradicción. Llamemos x a Sum^0/Sum^1. Sea $Sum^0/Sum^1>1$ (es decir, x>1) y supondremos que $Threshold_{ROC} \leq Threshold_{Dir}$. Remplazando los umbrales por sus definiciones:

$$\frac{skew \cdot x}{1 + skew \cdot x} \leq \frac{skew}{skew + 1}$$

operando

$$skew \cdot x \cdot (skew + 1) \leq skew \cdot (1 + skew \cdot x)$$

$$\Downarrow$$

$$skew^2 \cdot x + skew \cdot x \leq skew + skew^2 \cdot x$$

$$\Downarrow$$

$$skew \cdot x \leq skew$$

$$\Downarrow$$

$$x \leq 1$$

Lo que contradice la hipótesis. Las demostraciones para los otros dos casos son similares, por lo que no se incluyen.

□

Hasta este punto hemos visto que, en general, los umbrales obtenidos son diferentes, aunque hemos demostrado las equivalencias entre los límites de los tres umbrales. Ahora aboquémonos a la tarea de demostrar en qué casos los valores de los tres umbrales son iguales. El siguiente teorema muestra que esto es así cuando se parte de un conjunto de datos con un vector de probabilidades uniforme.

Teorema 1. Si *tenemos un conjunto de datos de n ejemplos con probabilidades uniformemente distribuidas y ordenadas de forma descendente:*

$$P^0 = \{1, \frac{m-1}{m}, \ldots, \frac{2}{m}, \frac{1}{m}, 0\}$$

es decir

$$p_i^0 = \frac{n-i}{n-1} \quad \text{y}$$

$$p_i^1 = \frac{i-1}{n-1}$$

por simplicidad usamos m, donde m=n–1

$$p_i^0 = \frac{m-i+1}{m} \quad \text{y} \qquad (9.9)$$

$$p_i^1 = \frac{i-1}{m}$$

entonces $Threshold_{ROC} = Threshold_{Dir} = Threshold_{Ord}$.

Demostración. Demostremos primero que $Threshold_{ROC} = Threshold_{Dir}$. Como las probabilidades están uniformemente distribuidas entonces:

$$Sum^1 = \sum_{i=0}^{m} \frac{i-1}{m} = \sum_{i=0}^{m} \frac{m-i+1}{m} = Sum^0$$

Por lo que

$$\frac{Sum^1}{Sum^0} = 1$$

Ahora, por la Proposición 2 se cumple que $Threshold_{ROC}$ = $Threshold_{Dir}$.

Ahora demostremos que $Threshold_{Ord}$ = $Threshold_{Dir}$. En una distribución uniforme la diferencia entre probabilidades contiguas es siempre igual a 1/m, por lo que

$$p_a - p_b = \frac{1}{m}$$

si despejamos la probabilidad p_a, tenemos

$$p_a = p_b + \frac{1}{m} \tag{9.10}$$

También sabemos por la ecuación (9.5) que la posición del umbral es

$$Pos = \frac{n-1}{skew + 1} + 1 = \frac{m}{skew + 1} + 1 \tag{9.11}$$

Y por la ecuación (9.6) esta posición es también igual a

$$Pos = a + frac \tag{9.12}$$

Como las probabilidades son uniformes, entonces, para encontrar el valor del índice a que corresponde a una determinada probabilidad p_a, sustituimos en la ecuación (9.9) el índice i por a y el valor de la probabilidad p_i^0 por p_a:

$$p_a = \frac{m - a + 1}{m}$$

Despejando a

$$a = m \cdot (1 - p_a) + 1$$

Si sustituimos este valor en la ecuación (9.12)

$$Pos = m \cdot (1 - p_a) + 1 + frac$$

Ahora sustituimos el valor de p_a dado por la ecuación (9.10)

$$Pos = m \cdot (1 - p_b - \frac{1}{m}) + 1 + frac$$

Operando

$$-2 + frac = Pos - m + p_b \cdot m$$

Sustituyendo el valor de *Pos* dado por la ecuación (9.11)

$$-2 + frac = \frac{m}{skew+1} - m + p_b \cdot m$$

Sumamos en el segundo miembro

$$-2 + frac = \frac{m - m \cdot skew - m + p_b \cdot m \cdot skew + p_b \cdot m}{skew+1}$$

Simplificando y factorizando en m

$$-1 + frac = \frac{m \cdot (-skew + p_b \cdot skew + p_b)}{skew+1}$$

Multiplicando toda la ecuación por -1

$$1 - frac = \frac{m \cdot (skew - p_b \cdot skew - p_b)}{skew+1}$$

Sustituyendo en la fórmula del $Threshold_{Ord}$ (9.4)

$$Threshold_{Ord} = p_b + \frac{1}{m} \cdot (\frac{m \cdot (skew - p_b \cdot skew - p_b}{skew+1})$$

Realizamos la división por m y realizamos la suma con p_b, resulta

$$Threshold_{Ord} = \frac{p_b \cdot skew + p_b + skew - p_b \cdot skew - p_b}{skew+1}$$

Simplificando, finalmente obtenemos la misma fórmula del $Threshold_{Dir}$.

$$Threshold_{Ord} = \frac{skew}{skew+1} = Threshold_{Dir}$$

□

Realizadas las demostraciones de que los valores de los umbrales coinciden cuando las probabilidades están distribuidas uniformemente. Hagamos un ejemplo para mostrar empíricamente estos resultados.

Consideremos un ejemplo con probabilidades de la clase 0 uniformemente distribuidas:

1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0

Además de los valores de *skew* de 1, 2, 3, 5 y 10.

Los resultados del cálculo de los tres umbrales se muestran en la siguiente tabla:

skew	*a*	*frac*	p_a	p_b	umbral		
					Dir	*Ord*	*ROC*
1	6	0.00	0.5	0.4	0.50	0.50	0.50
2	4	0.33	0.7	0.6	0.66	0.66	0.66
3	3	0.50	0.8	0.7	0.75	0.75	0.75
5	2	0.66	0.9	0.8	0.83	0.83	0.83
10	1	0.91	1.0	0.9	0.91	0.91	0.91

Como se observa todos los tipos de umbrales son iguales para cada *skew* específico.

9.4. Aprendizaje de los modelos miméticos

Una vez determinados los diferentes métodos de calcular el umbral, pasaremos a definir los modelos miméticos a estudiar. Para ello presentaremos varias propuestas sobre cómo inventar los conjuntos de datos de entrenamiento, así como diferentes esquemas de aprendizaje a aplicar. De esta forma cada modelo mimético se generará usando un esquema de aprendizaje concreto, sobre un conjunto de datos inventados de una cierta manera y empleando un tipo de umbral.

9.4.1. Generación del conjunto de entrenamiento del modelo mimético

Como dijimos en la introducción, en este capítulo contemplamos el caso en el que no tenemos los datos originales. Nótese que si pudiéramos disponer de ellos los resultados del modelo mimético serían mejores, puesto que la técnica mimética trabaja generalmente mejor cuando el conjunto de datos inventados se genera según la distribución real, como vimos en el capítulo 4.

En nuestro caso usaremos sólo datos inventados, que se generarán con las mínimas suposiciones posibles. En concreto, para aplicar una distribución uniforme basta con conocer el rango de cada atributo.

El conjunto de datos inventados D puede ser generado generalmente aplicando uno de los métodos siguientes:

a) El tipo a priori (o simplemente tipo a) genera el mismo porcentaje de clases que en los datos originales, es decir, la distribución de frecuencias relativas de las clases en los datos inventados son iguales que en los datos originales. En este caso, para generar los datos inventados necesitamos conocer los porcentajes de las clases, además de los valores máximos y mínimos (rango) de los atributos.
b) El tipo balanceado (tipo b) genera el mismo número de ejemplares inventados para cada clase, es decir, genera un 50% de instancias que tengan la clase 0 y un 50% instancias cuya clase sea 1.
c) El tipo aleatorio (tipo c) no establece ninguna condición sobre las frecuencias de las clases en los datos inventados, obteniéndolas tal como las genera aleatoriamente la distribución uniforme.
d) El tipo de sobremuestreo (tipo d) condiciona a que las frecuencias de las clases estén definidas de acuerdo al *skew*, es decir, se obtiene una proporción de $1/(skew + 1)$ de instancias con clase 0 y el resto $(skew/(skew + 1))$ con etiqueta de clase 1.

Para obtener los cuatro tipos, generamos ejemplos al azar y entonces los etiquetamos usando el oráculo. Se acaba este proceso cuando obtenemos el porcentaje correcto de ejemplares según el tipo de generación seleccionado. Este conjunto de datos etiquetado D_l entonces se utiliza para entrenar al modelo mimético.

Ahora ya sólo nos falta describir los esquemas y las configuraciones del aprendizaje que proponemos para crear los diferentes modelos miméticos a estudiar.

9.4.2. Esquemas de aprendizaje de los modelos miméticos

Como dijimos en la sección 2, la contextualización puede realizarse en diferentes puntos (véase la Figura 9.1). Estos diferentes puntos (junto con los métodos de determinación del umbral vistos en la sección 3 y los tipos de generación del conjunto inventado vistos en el subapartado anterior, 9.4.1) constituyen diferentes alternativas de contextualización. Todo esto amplía las fronteras de aplicación del método mimético en la contextualización al coste. Veamos los esquemas principales Un primer esquema simple (cuyo modelo producido denominamos Mim0) se corresponde con la técnica mimética original sin ninguna contextualización. Los otros dos esquemas (que dan lugar a los modelos Mim1 y Mim2) se obtienen aplicando los umbrales antes de (a priori, Mim2) o después de (a posteriori, Mim1) generar el modelo mimético. Finalmente, el último estilo (Mim3) corresponde a la técnica mimética estándar pero incluyendo sobremuestreo. Mim1 y Mim2 son contribuciones del autor, mientras que Mim0 y Mim3 los usaremos como referencia. Veamos en detalle estos esquemas.

Esquema 0: Mim0. Este esquema consiste en obtener el modelo mimético utilizando un clasificador J48 con poda, como se ha estudiado en detalle en los capítulos 4 y 6 (en especial en el 6, en el que tampoco se usan datos de entrenamiento). A continuación, se aplica el modelo usando un umbral de 0.5. La explicación gráfica de este proceso la encontramos en la Figura 9.6.

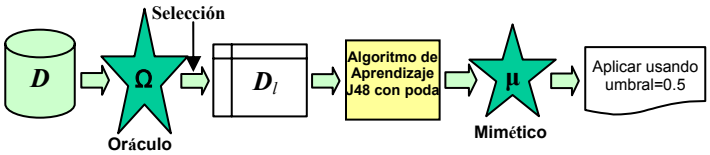


Figura 9.6. Proceso para la obtención y aplicación del modelo mimético simple (esquema 0).

Esquema 1: Mim1. Este esquema a posteriori consiste en generar un modelo mimético usando un clasificador J48 sin poda. Luego calculamos los valores del umbral y de *Ro* usando los datos inventados y el clasificador mimético obtenido. Finalmente se aplica el modelo mimético tomando en consideración el umbral y, en su caso, el valor *Ro* calculados anteriormente. El

esquema que explica gráficamente este proceso lo encontramos en la Figura 9.7.

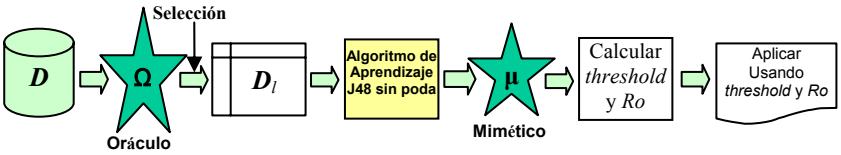


Figura 9.7. Proceso para la obtención y aplicación del modelo mimético a posteriori (esquema 1).

Esquema 2: Mim2. El esquema a priori consiste en calcular primero el umbral usando los datos inventados y el oráculo. A continuación se reetiquetan los datos inventados empleando el umbral y, en su caso, el índice *Ro.* Luego se entrena el modelo mimético usando un clasificador J48 con poda. Finalmente, se aplica este modelo sin considerar el umbral calculado anteriormente. La explicación gráfica del proceso se muestra en la Figura 9.8.

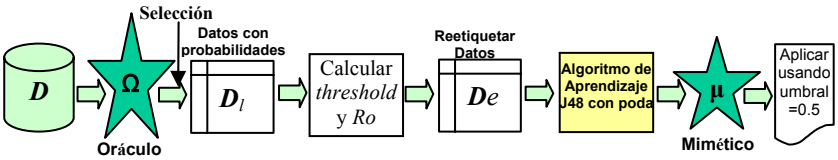


Figura 9.8. Proceso para obtención y aplicación del modelo mimético a priori (esquema 2).

Esquema 3: Mim3. El esquema por sobremuestreo consiste en obtener el modelo mimético con un clasificador J48 con poda empleando sobremuestreo, es decir, el conjunto de datos $\boldsymbol{D}_i$ se obtiene siguiendo el tipo d de generación de datos del subapartado 4.1. A continuación se evalúa el modelo usando un umbral de 0.5. La explicación gráfica de este proceso la encontramos en la Figura 9.9.

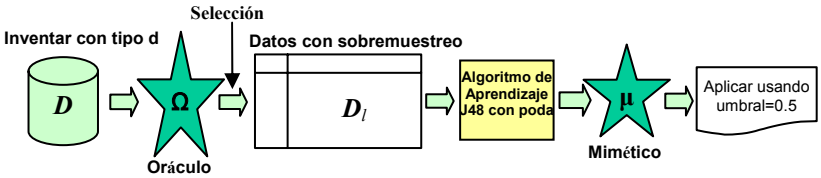


Figura 9.9. Proceso para obtención y aplicación del modelo mimético usando sobremuestreo (esquema 3).

Como se puede observar, en Mim0, Mim2 y Mim3 hemos usado poda, pero en Mim1 no. La razón es que en el Mim1 calculamos el umbral a partir del mimético y, para ello, es fundamental tener una buena estimación de las probabilidades. Diversos trabajos, como vimos en el capítulo 2, por ejemplo (Provost y Domingos, 2003; Ferri et. al, 2004), muestran que las probabilidades se estiman mejor si no hay poda y se realiza un suavizado de las probabilidades de las hojas.

9.4.3. Configuraciones

Como tenemos tres formas diferentes de calcular el umbral, combinamos los estilos Mim1 y Mim2 con todos los métodos de umbral, obteniendo seis tipos de configuraciones. Junto a éstos tenemos el Mim0 y el Mim3 (que son los estilos de aprendizaje que no emplean un umbral calculado). El esquema Mim3 es en realidad un caso especial del Mim0 en donde empleamos la técnica del sobremuestreo (tipo d de invención de datos).

Finalmente, si combinamos todos los tipos de esquemas (excepto Mim3) con cada uno de las formas de inventar datos (tipos a, b y c descritos en la sección 4.1) entonces tenemos 21 modelos a estudiar más el Mim3. A continuación se presentan en la Tabla 9.1 los veintidós casos posibles de modelos miméticos que estudiaremos experimentalmente.

Tabla 9.1 Enumeración de todos los posibles modelos miméticos estudiados.

Configuración	Método de generación del conjunto de datos inventado			
	a	b	c	d
Mim0	√	√	√	
Mim1Ord	√	√	√	
Mim1Dir	√	√	√	
Mim1ROC	√	√	√	
Mim2Ord	√	√	√	
Mim2Dir	√	√	√	
Mim2ROC	√	√	√	
Mim3				√

A partir de ahora la nomenclatura empleada para definir cada modelo mimético será *ConfiguraciónTipo*, donde *Configuración* es cualquiera de las configuraciones listados en la primera columna la Tabla 9.1 y *Tipo* es la letra que define la forma de inventar el conjunto de datos de entrenamiento. Por ejemplo, Mim0b significa que es el modelo mimético simple (esquema 0) entrenado con el conjunto de datos balanceados (tipo b). Mientras que Mim2Dira significa que es el modelo mimético a priori (esquema 2) usando el umbral directo (Dir) y entrenando con el conjunto de datos inventados con la misma distribución de frecuencias relativas de las clases que los datos originales (tipo a).

9.5. Experimentos

En esta sección presentamos los experimentos realizados para analizar los 22 modelos miméticos empleando 5 diferentes contextos de costes, además de realizarlos con dos oráculos diferentes: una red neuronal (RN) y un clasificador Naives Bayes (NB). La idea de utilizar la red neuronal es por su alta expresividad, dado que puede capturar patrones complejos, además de ser considerada un clasificador calibrado debido a que optimiza el error cuadrático medio y por lo tanto es un buen estimador de probabilidades. El caso de usar el clasificador Naives Bayes es para tener buena precisión (pero en general inferiores a la RN) y estimaciones de probabilidad que no necesariamente están calibradas (Cohen y Goldszmidt, 2004). Con esto tenemos oráculos que generan de muy buena a buena precisión y con diferencias en las estimaciones de las probabilidades. Esto nos permitirá estudiar el comportamiento de los modelos miméticos considerando los casos en los que la clase 1 es mayoritaria y minoritaria.

9.5.1. Configuración Experimental

Los experimentos están diseñados para evaluar los modelos miméticos ante diferentes escenarios. El primer escenario consiste en partir de conjuntos de datos balanceados o casi-balanceados. Para ver la influencia de la clase mayoritaria en conjuntos de datos desbalanceados se propone un segundo escenario que considera a la clase 1 como la clase mayoritaria

y, finalmente, un tercer escenario que estudia los métodos miméticos cuando la clase 1 es la clase minoritaria. Esta última variante no es redundante con la anterior porque sólo usaremos *skews* ≥ 1, con lo que hay que contemplar el desbalanceo en los dos sentidos.
Para los experimentos hemos empleado 20 conjuntos de datos (véase la Tabla 9.2) del repositorio de la UCI (Black y Merz, 1998).

Tabla 9.2. Información de los conjuntos de datos utilizados en los experimentos. Los conjuntos de datos con la observación de balanceados fueron balanceados aleatoriamente. Los conjuntos de datos con observación casi balanceado y desbalanceado son copia fiel del repositorio de la UCI.

No.	Conjunto de datos	Observación	Atributos		Tamaño	Porcentajes	
			Num.	Nom.		Clase A	Clase B
1	credit-a	Casi-Balanceado	6	9	690	307	383
2	heart-statlog	Casi-Balanceado	13	0	270	150	120
3	monks1	Balanceado	0	6	556	278	278
4	monks3	Casi-Balanceado	0	6	554	266	288
5	monks2	Balanceado	0	6	412	206	206
6	tic-tac	Balanceado	0	8	664	332	332
7	breast-cancer	Balanceado	0	9	170	85	85
8	labor	Balanceado	8	8	40	20	20
9	vote	Balanceado	0	16	336	168	168
10	diabetes	Balanceado	8	0	536	268	268
11	haberman-breast	No Balanceado	3	0	306	81	225
12	monks2	No Balanceado	0	6	601	206	395
13	abalone-morethan	No Balanceado	7	1	4177	1447	2730
14	tic-tac	No Balanceado	0	8	958	332	626
15	breast-cancer	No Balanceado	0	9	286	85	201
16	labor	No Balanceado	8	8	57	20	37
17	vote	No Balanceado	0	16	435	168	267
18	credit-g	No Balanceado	7	13	1000	300	700
19	diabetes	No Balanceado	8	0	768	268	500
20	liver	No Balanceado	6	0	345	145	200

Los primeros diez conjuntos de datos servirán para las pruebas en el escenario de datos balanceados o casi-balanceados y los diez restantes para los dos escenarios de datos desbalanceados. Los conjuntos de datos del 11 al 20 se procesaron de tal manera que la clase 1 fuera la mayoritaria o la minoritaria dependiendo del tipo de experimentos.

Se realizan las siguientes baterías de experimentos:

1) Se emplean los primeros 10 conjuntos de datos casi-balanceados o balanceados (Tabla 9.2) empleando la matriz de costes (9.3, página 151). Los calores del *skew* considerados son: 1, 2, 3, 5, y 10.

2) Se emplean los últimos 10 conjuntos de datos (Tabla 9.2) no balanceados, con las mismas matrices de costes y valores de *skew* que en el punto anterior. La clase mayoritaria es la clase 1.

3) Igual que 2 pero con la clase 1 como minoritaria.

Todos los modelos miméticos son clasificadores construidos con el algoritmo J48. En el caso de los oráculos se usan los algoritmos del Weka *MultilayerPerceptron* para la red neuronal y *NaiveBayes* para el clasificador Naives Bayes. La cantidad de datos inventados es de 10,000. También normalizamos las estimaciones de probabilidad. En todos los clasificadores empleamos el ajuste de Laplace (por ejemplo, Elkan, 2001). Para todos los experimentos usamos una validación cruzada de 10 pliegues con 10 repeticiones. Los *rankings* y las pruebas t de *Student* se realizan en el nivel de confianza del 5% usando Weka. El *ranking* de Weka es un proceso para posicionar un modelo en una escala ordinal en relación al total de ganados y perdidos en las pruebas t frente a los otros modelos. Finalmente, cuando mostramos los resultados promedios, utilizaremos la media aritmética.

Para hacerse una idea de la magnitud de estos experimentos, hemos de decir que el total de aprendizajes realizados son: 30 (conjuntos de datos) x 2 (oráculos) x 10 (iteraciones) x 10 (pliegues) x 5 (contextos de coste) x 22 (modelos miméticos) = 660,000.

9.5.2. Resultados generales

A continuación presentamos los resultados de los experimentos realizados.

Empecemos con una visión global. En la Tabla 9.3 se muestran los costes promedio obtenidos para todos los modelos miméticos por cada uno de los *skew*. Los mismos resultados se pue-

den ver de forma gráfica en la Figura 9.10. En general se observa que para un *skew*=1 los costes de los diferentes métodos son muy similares con un valor aproximado de 20. A medida que aumenta el *skew* las diferencias de los costes para cada modelo van siendo más visibles. En general, para *skew*s mayores de 2, el Mim2Ordb es el que da un mejor resultado global; de hecho, los Mim2 suelen ser mejores que los Mim1. En la Figura 9.10 se observan dos mínimos importantes para valores del *skew* mayores que 2, definidos por las parejas de modelos Mim1Orda-Mim1Ordb y Mim2Orda-Mim2Ordb.

Tabla 9.3 Costes promedio de todos los modelos miméticos agrupados por *skew*. En negrita los de menor coste global.

Modelo	*skew*					Media
	1	2	3	5	10	
Mim0a	**19.74**	29.58	39.68	59.53	109.81	51.67
Mim0b	20.24	30.30	40.27	60.91	114.45	53.24
Mim0c	20.56	31.35	41.95	63.39	116.87	54.82
Mim1Dira	20.00	29.41	38.03	51.34	75.88	42.93
Mim1Dirb	20.37	30.05	38.37	51.19	73.11	42.62
Mim1Dirc	20.72	30.82	39.29	51.74	73.16	43.15
Mim1Orda	25.20	32.00	34.32	36.86	39.07	33.49
Mim1Ordb	20.67	29.36	33.03	36.43	38.76	31.65
Mim1Ordc	26.17	34.85	39.49	43.84	47.22	38.32
Mim1ROCa	20.25	29.65	37.80	51.03	71.92	42.13
Mim1ROCb	20.37	29.84	37.66	50.60	72.49	42.19
Mim1ROCc	20.73	29.31	36.54	48.25	68.92	40.75
Mim2Dira	**19.74**	**27.00**	31.86	39.25	51.74	33.92
Mim2Dirb	20.24	27.58	32.24	39.67	51.91	34.33
Mim2Dirc	20.56	29.02	34.69	42.98	58.87	37.22
Mim2Orda	23.80	29.98	32.75	35.77	37.42	31.94
Mim2Ordb	20.61	28.51	**31.69**	**34.84**	**37.03**	**30.54**
Mim2Ordc	24.75	33.26	38.11	41.83	45.74	36.74
Mim2ROCa	20.43	27.77	32.65	39.10	50.22	34.03
Mim2ROCb	20.58	28.13	33.75	40.83	53.47	35.35
Mim2ROCc	20.83	29.30	34.92	44.46	60.12	37.93
Mim3	20.33	28.51	34.85	44.69	61.65	38.01

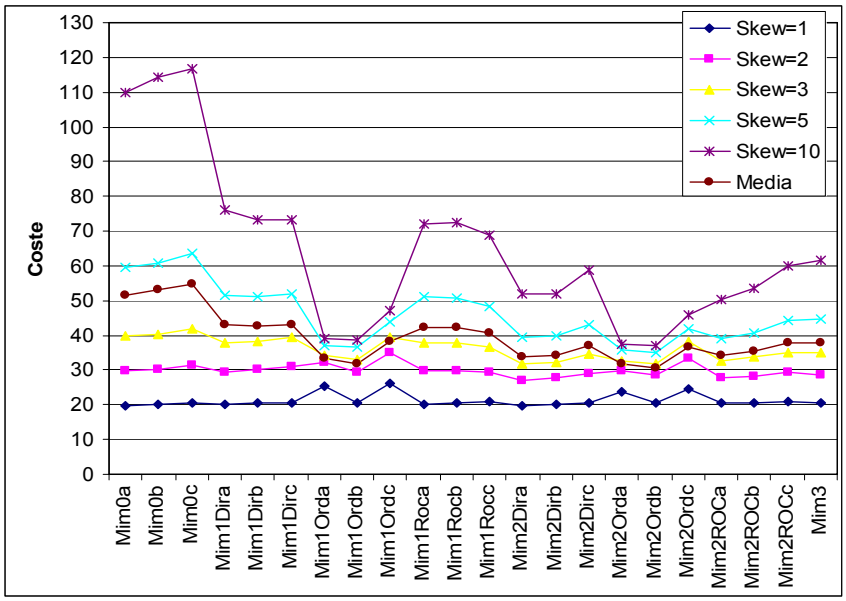


Figura 9.10 Resultados promedio de costes agrupados por *skew*.

Cambiemos del escenario general al caso específico del oráculo NB. La Tabla 9.4 muestra los promedios de los aciertos y los costes para cada modelo mimético separado en cada tipo de batería de experimentos. Aquí también se observan los mínimos ya indicados por la visión global. Las mayores ganancias en coste respecto al oráculo NB son debidas a los conjuntos de datos donde el *skew* actúa positivamente sobre la clase mayoritaria, con una reducción promedio aproximada del 34% respecto al coste original y con una pérdida en precisión del 6%. La mejora en coste es también muy significativa respecto al Mim3, que representa la solución por sobremuestreo. En el caso de los conjuntos donde el *skew* afecta positivamente sobre la clase minoritaria, la reducción de coste es también muy significativa para algunos métodos (Mim2Ordb) pero no para todos (algunos empeoran, como Mim1Dirc). En el caso de la clase minoritaria como era de esperar, el porcentaje de aciertos se ve afectado de manera importante. Finalmente, el caso balanceado tiene un comportamiento intermedio.

Tabla 9.4 Precisiones y costes promedio de todos los modelos miméticos de acuerdo al tipo de batería de experimento usando Naive Bayes como oráculo.

Modelo	Balanceado		Mayoritaria		Minoritaria	
	Precisión	Coste	Precisión	Coste	Precisión	Coste
Mim0a	71.89	31.11	70.40	49.25	**70.39**	91.39
Mim0b	72.03	31.38	70.25	55.83	70.24	84.94
Mim0c	69.53	39.85	69.48	70.64	69.51	75.17
Mim1Dira	72.67	28.28	70.28	42.51	69.71	79.99
Mim1Dirb	72.71	28.39	70.27	47.10	69.22	71.49
Mim1Dirc	70.25	33.65	70.12	56.32	68.49	63.41
Mim1Orda	63.83	20.15	68.37	32.72	51.81	49.32
Mim1Ordb	63.97	20.19	69.04	**30.36**	55.50	47.93
Mim1Ordc	62.39	25.33	66.15	41.63	57.23	55.34
Mim1ROCa	72.96	28.38	70.61	46.19	68.37	69.58
Mim1ROCb	**73.01**	28.44	70.56	46.05	68.57	71.37
Mim1ROCc	72.28	27.12	70.42	44.95	67.91	66.63
Mim2Dira	67.87	**17.06**	70.84	36.16	62.81	53.56
Mim2Dirb	67.86	17.23	**70.91**	37.76	62.54	52.87
Mim2Dirc	64.88	26.43	70.37	45.15	60.97	49.91
Mim2Orda	65.92	19.25	69.51	32.38	56.00	47.39
Mim2Ordb	65.90	19.24	69.87	**30.39**	58.56	**46.35**
Mim2Ordc	64.09	25.31	67.06	40.62	60.21	54.35
Mim2ROCa	68.89	18.76	70.56	37.31	60.29	49.83
Mim2ROCb	68.83	19.09	70.80	38.12	62.86	54.22
Mim2ROCc	66.27	25.68	69.12	41.01	62.70	57.35
Mim3	72.45	27.84	70.43	40.34	68.92	64.17
NB	**76.84**	28.16	**74.26**	64.89	**74.26**	67.93

Pasemos al segundo caso específico con el oráculo RN. La Tabla 9.5 muestra los promedios de las precisiones y los costes para cada modelo mimético separado en cada tipo de batería de experimentos. Aquí se observa una coincidencia con el mínimo de la visión global en dos de los tres casos; esto sucede en los dos casos con conjuntos de datos no balanceados. En concreto, los mínimos se encuentran en Mim2Ordb. En el otro caso (balanceado) el mejor método es Mim3. Este caso excepcional lo estudiaremos más adelante. Las mayores ganancias en coste son más modestas y corresponden a los conjuntos de datos con clase minoritaria, teniendo una reducción promedio aproximada del 5.5% respecto al coste original y con una pérdida en precisión del 17.8%. Pese a todo esto, Mim2Ordb da los mejores resultados en dos de las tres situaciones.

Tabla 9.5 Precisiones y costes promedio de todos los modelos miméticos de acuerdo al tipo de batería de experimento usando la red neuronal como oráculo.

Modelo	Balanceado		Mayoritaria		Minoritaria	
	Precisión	Coste	Precisión	Coste	Precisión	Coste
Mim0a	82.91	15.53	76.83	56.26	**76.91**	66.46
Mim0b	83.03	15.54	73.93	73.16	74.40	58.57
Mim0c	82.94	19.21	76.36	56.82	76.51	67.26
Mim1Dira	79.78	12.32	76.44	40.95	71.97	53.54
Mim1Dirb	79.94	12.28	75.85	48.06	69.31	48.39
Mim1Dirc	81.00	12.30	75.85	39.23	72.20	53.97
Mim1Orda	66.92	14.64	70.69	34.65	49.58	49.45
Mim1Ordb	67.00	14.69	71.76	30.66	54.59	46.05
Mim1Ordc	65.38	16.42	69.91	41.07	51.33	50.10
Mim1ROCa	79.65	12.18	76.23	47.38	68.17	49.09
Mim1ROCb	79.73	12.13	76.03	46.90	68.85	48.26
Mim1ROCc	79.33	11.92	76.29	45.65	67.56	48.23
Mim2Dira	82.24	12.13	**77.38**	35.10	73.04	49.50
Mim2Dirb	82.25	11.96	75.83	40.73	70.39	45.41
Mim2Dirc	82.62	15.48	76.29	36.22	72.57	50.15
Mim2Orda	69.25	13.81	73.13	32.80	53.53	46.03
Mim2Ordb	69.38	13.76	73.25	**29.97**	57.50	**43.53**
Mim2Ordc	67.86	15.47	72.57	37.99	54.13	46.68
Mim2ROCa	82.40	12.25	77.17	40.21	70.77	45.83
Mim2ROCb	82.31	12.21	75.63	42.89	69.99	45.58
Mim2ROCc	**83.19**	14.39	76.50	43.51	71.45	45.61
Mim3	79.71	**11.03**	76.32	39.62	67.13	45.03
RN	**86.02**	13.07	**80.88**	44.51	**80.83**	53.02

Hasta aquí hemos realizado el estudio global de resultados. Ahora nos abocaremos a los análisis específicos. En concreto realizaremos el análisis de la significancia para el coste con un *skew* de 10, comparando las pruebas t de los mejores modelos. El segundo análisis estudiaremos la relación entre el coste y el porcentaje de aciertos. Y finalmente induciremos un análisis de los árboles de decisión generados por el mejor método mimético, viendo como es la inteligibilidad de los modelos obtenidos en la contextualización.

9.5.3. Análisis específicos

9.5.3.1. Análisis de la significancia para el coste

Ahora pasaremos a ver cuál de las técnicas es mejor con respecto al coste. Para ello, como hemos dicho, usaremos un ranking basado en los casos en lo que gana cada técnica.

Los tres mejores costes obtenidos por *ranking* en los experimentos con un *skew* de 10 se muestran en la Tabla 9.6.

Tabla 9.6. Ranking de los modelo en base al coste.

Oráculo	Batería	*Ranking* de modelos		
		1º	2º	3º
NB	Balanceados	Mim2Orda	Mim2Ordb	Mim2Dira
RN	Balanceados	Mim3	Mim2Ordb	Mim1Dirc
NB	Mayoritaria	Mim2Ordb	Mim1Ordb	Mim2Orda
RN	Mayoritaria	Mim2Ordb	Mim2Orda	Mim2Ordc
NB	Minoritaria	Mim2Orda	Mim2Ordb	Mim2Dirb
RN	Minoritaria	Mim2Ordb	Mim2Orda	Mim2ROCb

Como se puede ver, Mim2Ordb ocupa el primero o segundo lugar en todos los casos.

Para comprobar si realmente en el caso RN-Balanceado el mejor modelo es Mim3, vamos a realizar un análisis más detallado a los mejores modelos miméticos (para cada uno de los casos incluidos en la Tabla 9.6), incluyendo el oráculo, el Mim3 y el mejor Mim0 (en la Tabla 9.7 mostramos un resumen de estos modelos). Note que, para el caso RN-balanceados se han tomado como los tres primeros modelos miméticos los que ocupan en el ranking las posiciones 2º, 3º y 4º, ya que el 1º es el Mim3 que se ha tomado ya como modelo a analizar. Se les realizaron las pruebas t de uno contra todos, para ver la significancia de los modelos. Dichos resultados se muestran en las tablas 9.8 hasta la 9.13, según cada uno de los seis casos (oráculo y tipo de conjunto de datos).

Tabla 9.7. Modelos seleccionados para realizar las pruebas t de *Student*. Bala se refiere a los conjuntos de datos balanceados, Mayo significa que son los conjuntos de datos donde la clase 1 es la mayoritaria y Mino representa los conjuntos de datos donde la clase 1 es la minoritaria.

Batería	Modelo Oráculo	Modelos miméticos				
		1	2	3	Mim0	Mim3
Bala	NB	Mim2Orda	Mim2Ordb	Mim2Dira	Mim0a	Mim3
Bala	RN	Mim2Ordb	Mim1Dirc	Mim2Orda	Mim0b	Mim3
Mayo	NB	Mim2Ordb	Mim1Ordb	Mim2Orda	Mim0c	Mim3
Mayo	RN	Mim2Ordb	Mim2Orda	Mim2Ordc	Mim0c	Mim3
Mino	NB	Mim2Orda	Mim2Ordb	Mim2Dirb	Mim0b	Mim3
Mino	RN	Mim2Ordb	Mim2Orda	Mim2ROCb	Mim0b	Mim3

La interpretación de la notación en las tablas de resultados para las pruebas t de *Student* es: el modelo de la fila (Gana/ Empata/Pierde) al modelo de la columna. La variable medida en las pruebas t es el coste global.

Para conjuntos de datos balanceados empleando NB (ver Tabla 9.8) si bien Mim2Dira parece competitivo uno contra uno con Mim2Orda y Mim2Ordb, la visión global es que los mejores modelos son Mim2Orda y Mim2Ordb, no existiendo diferencias significativas entre ellos.

Tabla 9.8 Pruebas t para modelos obtenidos con conjuntos de datos balanceados y NB como oráculo.

	NB	Mim0a	Mim2Orda	Mim2Dira	Mim2Ordb	Mim3
NB	-	(3/4/3)	(1/1/8)	(1/2/7)	(1/1/8)	(2/4/4)
Mim0a	(3/4/3)	-	(1/0/9)	(0/2/8)	(1/0/9)	(0/3/7)
Mim2Orda	(8/1/1)	(9/0/1)	-	(4/1/5)	(1/9/0)	(9/0/1)
Mim2Dira	(7/2/1)	(8/2/0)	(5/1/4)	-	(5/1/4)	(6/2/2)
Mim2Ordb	(8/1/1)	(9/0/1)	(0/9/1)	(4/1/5)	-	(9/0/1)
Mim3	(4/4/2)	(7/3/0)	(1/0/9)	(2/2/6)	(1/0/9)	-

Los mejores modelos para conjuntos de datos balanceados empleando la RN (ver Tabla 9.9) son Mim3, Mim1Dirc, Mim2Orda y Mim2Ordb, sin embargo, los Mim2Ord le ganan por uno al Mim1Dirc. Aunque el Mim3 le gana a la RN y al Mim0b, empata con los demás modelos y está equilibrado en cuanto a conjuntos de datos ganados y perdidos con los demás modelos. Por lo tanto, en base a este análisis más detallado, no se sustenta que Mim3 sea mejor que Mim2Orda y Mim2Ordb, sino más bien al contrario.

Tabla 9.9 Pruebas t para modelos obtenidos con conjuntos de datos balanceados y como oráculo RN.

	RN	Mim0b	Mim1Dirc	Mim2Orda	Mim2Ordb	Mim3
RN	-	(4/5/1)	(2/2/6)	(4/1/5)	(4/1/5)	(2/3/5)
Mim0b	(1/5/4)	-	(1/3/6)	(4/1/5)	(4/0/6)	(2/2/6)
Mim1Dirc	(6/2/2)	(6/3/1)	-	(4/1/5)	(4/1/5)	(2/6/2)
Mim2Orda	(5/1/4)	(5/1/4)	(5/1/4)	-	(1/8/1)	(5/0/5)
Mim2Ordb	(5/1/4)	(6/0/4)	(5/1/4)	(1/8/1)	-	(5/0/5)
Mim3	(5/3/2)	(6/2/2)	(2/6/2)	(5/0/5)	(5/0/5)	-

Como consecuencia de las dos tablas anteriores, en general para conjuntos de datos balanceados se recomienda utilizar Mim2 con el umbral por ordenamiento y un conjunto de entrenamiento balanceado (Mim2Ordb).

El mejor modelo para conjuntos de datos no balanceados y clase mayoritaria empleando NB (ver Tabla 9.10) es claramente Mim2Ordb.

Tabla 9.10 Pruebas t para modelos obtenidos con conjuntos de datos no balanceados y clase mayoritaria usando el oráculo NB.

	NB	Mim0c	Mim1Ordb	Mim2Orda	Mim2Ordb	Mim3
NB	-	(3/4/3)	(0/1/9)	(0/1/9)	(0/1/9)	(0/3/7)
Mim0c	(3/4/3)	-	(0/0/10)	(0/0/10)	(0/0/10)	(0/0/10)
Mim1Ordb	(9/1/0)	(10/0/0)	-	(6/0/4)	(2/4/4)	(9/1/0)
Mim2Orda	(9/1/0)	(10/0/0)	(4/0/6)	-	(0/5/5)	(10/0/0)
Mim2Ordb	(9/1/0)	(10/0/0)	(4/4/2)	(5/5/0)	-	(10/0/0)
Mim3	(7/3/0)	(6/3/1)	(0/1/9)	(0/0/10)	(0/0/10)	-

El mejor modelo para conjuntos de datos no balanceados y clase mayoritaria empleando RN (ver Tabla 9.11) es de nuevo claramente Mim2Ordb.

Tabla 9.11 Pruebas t para modelos obtenidos con conjuntos de datos no balanceados y clase mayoritaria usando el oráculo RN.

	RN	Mim0c	Mim2Orda	Mim2Ordb	Mim2Ordc	Mim3
RN	-	(7/1/2)	(1/0/9)	(1/0/9)	(1/0/9)	(1/1/8)
Mim0c	(2/1/7)	-	(1/1/8)	(1/1/8)	(1/1/8)	(1/1/8)
Mim2Orda	(9/0/1)	(8/1/1)	-	(1/6/3)	(4/4/2)	(6/3/1)
Mim2Ordb	(9/0/1)	(8/1/1)	(3/6/1)	-	(5/3/2)	(6/3/1)
Mim2Ordc	(9/0/1)	(8/1/1)	(2/4/4)	(2/3/5)	-	(5/4/1)
Mim3	(8/1/1)	(8/1/1)	(1/3/6)	(1/3/6)	(1/4/5)	-

Por lo tanto, a la vista de las dos tablas anteriores, en general para los conjuntos de datos no balanceados con clase mayoritaria se recomienda utilizar Mim2 con el umbral por ordenamiento y un conjunto de entrenamiento balanceado (Mim2Ordb).

El mejor modelo para conjuntos de datos no balanceados y clase minoritaria empleando NB (ver Tabla 9.12) es Mim2Orda, aunque Mim2Ordb es el segundo mejor.

Tabla 9.12 Pruebas t para modelos obtenidos con conjuntos de datos no balanceados y clase minoritaria usando el oráculo NB.

	NB	Mim0b	Mim2Orda	Mim2Ordb	Mim2Dirb	Mim3
NB	-	(4/2/4)	(1/0/9)	(1/0/9)	(1/1/8)	(1/2/7)
Mim0b	(4/2/4)	-	(0/0/10)	(0/0/10)	(0/1/9)	(0/2/8)
Mim2Orda	(9/0/1)	(10/0/0)	-	(3/7/0)	(5/2/3)	(8/1/1)
Mim2Ordb	(9/0/1)	(10/0/0)	(0/7/3)	-	(4/3/3)	(8/1/1)
Mim2Dirb	(8/1/1)	(9/1/0)	(3/2/5)	(3/3/4)	-	(7/1/2)
Mim3	(7/2/1)	(8/2/0)	(1/1/8)	(1/1/8)	(2/1/7)	-

El mejor modelo para conjuntos de datos no balanceados y clase minoritaria empleando RN (ver Tabla 9.13) es Mim2Ordb.

Tabla 9.13 Pruebas t para modelos obtenidos con conjuntos de datos no balanceados y clase minoritaria usando el oráculo RN.

	RN	Mim0b	Mim2Orda	Mim2Ordb	Mim2ROCb	Mim3
RN	-	(5/1/4)	(3/1/6)	(3/1/6)	(2/0/8)	(2/0/8
Mim0b	(4/1/5)	-	(2/2/6)	(2/2/6)	(0/2/8)	(2/0/8)
Mim2Orda	(6/1/3)	(6/2/2)	-	(2/5/3)	(5/1/4)	(7/0/3)
Mim2Ordb	(6/1/3)	(6/2/2)	(3/5/2)	-	(5/1/4)	(7/0/3)
Mim2ROCb	(8/0/2)	(8/2/0)	(4/1/5)	(4/1/5)	-	(3/3/4)
Mim3	(8/0/2)	(8/0/2)	(3/0/7)	(3/0/7)	(4/3/3)	-

A la vista de las dos tablas anteriores, y aunque en la Tabla 9.12 gana ligeramente Mim2Orda sobre Mim2Ordb, por simplicidad, en general para los conjuntos de datos no balanceados con clase minoritaria se recomienda utilizar Mim2 con el umbral por ordenamiento y un conjunto de entrenamiento balanceado (Mim2Ordb).

En resumen, en un análisis global de los resultados vemos que el mejor modelo desde el punto de vista de coste es Mim2Ordb.

9.5.3.2. Análisis de la relación entre el coste y el porcentaje de aciertos

Veamos ahora cuál es el comportamiento si tenemos en cuenta tanto la precisión como el coste. En las Tablas 9.4 y 9.5 mostramos los datos de acierto pero no los analizamos en detalle. Para realizar este estudio proponemos un índice de pérdida de precisión por unidad de coste ganado. El cálculo de este índice lo realizamos con la siguiente fórmula:

$$Índice = \frac{\dfrac{Prec(Oráculo) - Prec(Mimético)}{Prec(Oráculo)}}{\dfrac{Coste(Oráculo) - Coste(Mimético)}{Coste(Oráculo)}}$$

donde la función *Prec* define la precisión y la función *Coste* es el coste global del modelo. Nos interesan solamente los modelos miméticos que logren ganar en coste al oráculo, es decir, aquellos cuyo índice es positivo.

La Tabla 9.14 muestra los índices de pérdida obtenidos con los valores promedios de la precisión y el coste. Se observa que los mejores modelos son Mim2Dira y Mim2Ordb en los conjuntos balanceados. En los conjuntos donde la primera clase es la mayoritaria, el mejor modelo es Mim2Dira, mientras que en los conjuntos donde la primera clase es la minoritaria, el mejor modelo es Mim2ROCa. En el caso de considerar un promedio general, el mejor modelo es Mim2Dira. Sin embargo, el que siempre ocupa el primer o segundo lugar en todos los tipos de conjuntos es el modelo Mim2Dira.

En resumen, en un análisis de los resultados vemos que el mejor modelo desde el punto de vista de coste y porcentaje de aciertos es Mim2Dira.

9.5.3.3. Análisis de la inteligibilidad del modelo mimético en la contextualización

Una de las ventajas de que los modelos Mim2 sean mejores que los Mim1 (y sobre todo que el Mim0, el Mim3 y el oráculo) es que, dado un nuevo contexto, el modelo Mim2 representa una adaptación al contexto que, al estar en forma de reglas y no

necesitar de umbral (pues se aplica antes, ver Figura 9.8), supone una "explicación" de la adaptación. Para tener una visión desde el punto de vista de la comprensibilidad de los modelos obtenidos hemos realizado un análisis de las reglas para el modelo Mim2Ordb con el conjunto de datos "*vote*" empleando los dos oráculos. Se eligió aleatoriamente un experimento de cada *skew.*

Tabla 9.14 Índice de pérdida en los promedios de precisión y coste para los modelos miméticos. En negrita el menor índice. Los valores negativos representan malos modelos, debido a que no lograron ganarle en coste al oráculo.

Modelo	Balanceado	Mayoritaria	Minoritaria	Todos
Mim0a	-0.3772	1.4360	-0.1645	-0.3549
Mim0b	-0.3466	-0.3949	-0.3614	-0.3506
Mim0c	-0.1475	-0.3632	-0.3292	-0.2877
Mim1Dira	4.2222	0.2290	-0.8309	1.3231
Mim1Dirb	4.6568	0.4469	12.1272	1.2950
Mim1Dirc	-0.6224	0.4672	3.1457	1.5909
Mim1Orda	1.2621	0.2699	1.8885	0.8279
Mim1Ordb	1.2712	0.2090	1.3016	0.6410
Mim1Ordc	-17.0051	0.5043	2.3401	1.3871
Mim1Roca	3.8863	0.3696	6.3537	1.1339
Mim1Rocb	3.9400	0.3672	10.4019	1.1325
Mim1Rocc	1.3027	0.3162	2.5128	0.8331
Mim2Dira	**0.2682**	**0.1282**	0.8388	**0.3283**
Mim2Dirb	**0.2681**	0.1918	0.7628	0.3791
Mim2Dirc	-5.6728	0.2136	0.8049	0.5404
Mim2Orda	0.8583	0.1994	1.2910	0.6160
Mim2Ordb	0.8481	0.1728	0.9796	0.5109
Mim2Ordc	17.7063	0.3552	1.5955	0.9782
Mim2ROCa	0.2866	0.1641	**0.7414**	0.3665
Mim2ROCb	0.2989	0.2165	0.8205	0.4121
Mim2ROCc	2.9313	0.2700	0.9080	0.5719
Mim3	1.1505	0.2009	1.2641	0.5027

Se analizaron las reglas de cada modelo cuya cobertura es igual o mayor al 5% de las instancias (ver Tabla 9.15). Como resultado general, indica que aunque el atributo *physician-fee-freeze* siempre está en el primer nivel del árbol, el segundo y tercer nivel del árbol presentan variaciones en sus atributos. No obstante, estas diferencias no permiten concluir si realmente los árboles de decisión son diferentes o no, ya que globalmente (sin tener en cuenta el nivel), los atributos se repiten. Sin embargo, analizando la cobertura de las reglas, si se puede afirmar que los árboles son diferentes (ver Tabla 9.15).

Tabla 9.15 Resumen de las reglas de los árboles de decisión de los modelos miméticos Mim2Ordb generados con el conjunto de datos balanceado *vote*. Entre paréntesis se muestran la cantidad de instancias que cubre la regla y los errores encontrados (total instancias cubiertas/errores).

skew	Reglas principales
10	physician-fee-freeze = n | crime = y | | education-spending = y | | | aid-to-nicaraguan-contras = n: republican (702/3) physician-fee-freeze = y | crime = n | | education-spending = y: republican (1414/7) | crime = y: republican (3606/10)
5	physician-fee-freeze = y | el-salvador-aid = n | | crime = y | | | education-spending = y: republican (838/4) | el-salvador-aid = y | | adoption-of-the-budget-resolution = n: republican (2088/2) | | adoption-of-the-budget-resolution = y | | | crime = y: republican (895/4)
3	physician-fee-freeze = y | crime = y | | el-salvador-aid = n | | | adoption-of-the-budget-resolution = n: republican (859/8) | | el-salvador-aid = y: republican (1986/9)
2	physician-fee-freeze = y | crime = y | | education-spending = y | | | aid-to-nicaraguan-contras = n: republican (1008/2)
1	physician-fee-freeze = n | el-salvador-aid = n | | adoption-of-the-budget-resolution = y: democrat (923/8) physician-fee-freeze = y | crime = y | | adoption-of-the-budget-resolution = n | | | education-spending = y | | | | aid-to-nicaraguan-contras = n: republican (546)

Para confirmar esta aseveración cualitativa, se extrajeron los siguientes parámetros del árbol generado por el modelo: el número de reglas (*R*), el número total de condiciones (*C*), el máximo número de reglas por clase (*Max_R*) y el máximo número de condiciones por clase (*Max_C*). Con estos parámetros y empleando las fórmulas 2.1 y 2.2 (capítulo 2) se calcularon la complejidad y la comprensibilidad del modelo. Obtuvimos que en general la complejidad de las reglas disminuye conforme el *skew* aumenta a partir de un *skew* mayor de 2 (ver Tabla 9.16). Como consecuencia la comprensibilidad aumenta cuando el *skew* aumenta, ésta es otra valoración cuantitativa que confirma que los árboles son diferentes para diferentes valores del *skew*.

Tabla 9.16 Resumen de los parámetros de los árboles de decisión de los modelos miméticos Mim2Ordb generados con el conjunto de datos vote. Donde *R* = número de reglas, *C* = número de condiciones, *Max_R* = máximo número de reglas por clase, *Max_C* = Máximo número de condiciones por clase, *Compl* = complejidad y *Compre* = comprensibilidad.

Oráculo	Batería	*skew*	*R*	*C*	*Max_R*	*Max_C*	*Compl*	*Compre*
NB	Balanceados	10	167	333	99	12	667	-9.48
		5	263	525	145	12	1051	-14.85
		3	322	643	166	12	1287	-18.17
		2	387	793	203	13	1587	-20.65
		1	448	895	241	13	1791	-23.24
NB	Mayoritaria	10	200	399	121	14	799	-9.72
		5	316	631	180	14	1263	-15.28
		3	403	805	230	15	1611	-18.14
		2	496	991	280	14	1983	-23.85
		1	455	909	241	13	1819	-23.6
NB	Minoritaria	10	185	369	106	12	739	-10.5
		5	265	529	142	12	1059	-14.98
		3	288	575	151	13	1151	-15.05
		2	361	721	190	13	1443	-18.79
		1	456	911	250	13	1823	-23.63
RN	Balanceados	10	125	249	70	13	499	-6.65
		5	193	385	103	14	771	-9.46
		3	263	525	133	13	1051	-13.79
		2	253	505	127	13	1011	-13.28
		1	250	499	125	14	999	-12.2
RN	Mayoritaria	10	176	351	98	14	703	-8.62
		5	227	453	124	14	907	-11.06
		3	388	775	212	13	1551	-20.15
		2	311	621	156	13	1243	-16.25
		1	222	443	113	13	887	-11.68
RN	Minoritaria	10	135	269	75	13	539	-7.16
		5	167	333	87	13	667	-8.84
		3	212	423	106	12	847	-12.08
		2	292	583	149	13	1167	-15.27
		1	196	391	99	14	783	-9.64

En definitiva, el método mimético y, en especial, las variantes Mim2 permiten obtener modelos específicos y diferentes para cada contexto, que permiten dar una explicación al usuario de qué reglas cambian y son más importantes en cada contexto de costes.

Esto hace del método mimético una herramienta doblemente útil en la contextualización a costes.

9.6. Discusión

A continuación realizamos una discusión sobre los métodos analizados.

En primer lugar, de las dos configuraciones propuestas (Figuras 9.6 y 9.7), la Mim2 (a priori) es preferible, y la razón ha de buscarse en el hecho de que el oráculo es casi siempre mejor que su imitación (el mimético) con lo que la búsqueda del umbral se puede realizar mejor en el propio oráculo. En segundo lugar, entre los tres tipos principales de generación del conjunto de datos inventados, los resultados muestran que los tipos a)(a priori) y b) (balanceado) son claramente mejores que c) (aleatorio). Por tanto, es importante controlar la proporción de clases etiquetadas por el oráculo. Si bien las diferencias entre a) y b) no son muy grandes, dependen de la configuración y del tipo de conjunto de datos (balanceado o no), siendo iguales en el caso de los conjuntos de datos perfectamente balanceados. En tercer lugar, respecto a los métodos de determinación del umbral, es conveniente centrar la discusión de por qué el método de ordenamiento funciona bien según la calidad (calibración y separabilidad) de las probabilidades estimadas. El método del umbral directo daría muy buenos resultados si las probabilidades estuvieran calibradas. Debido a que regularmente éste no es el caso, es necesario tener en cuenta únicamente el orden de las probabilidades estimadas y hacer una partición de las mismas acorde con los costes, siendo éste el método del umbral por ordenamiento. Esta opción parece dar los mejores resultados respecto a costes. No obstante, dado que este umbral se ve afectado por el rango de valores de las probabilidades estimadas, vimos necesario considerar el análisis ROC para mitigar este problema, proponiendo derivar el umbral mediante las curvas NROC que, aun dando peores resultados en coste, obtienen un compromiso entre coste y porcentaje de acierto. De nuevo, el problema del método basado en ROC es que asume que las probabilidades están bien calibradas.

Para comprender mejor estos resultados, pasamos ahora a mostrar diferentes medidas que analizan la calidad de las probabilidades (calibración y separabilidad/ordenación, medidas por MSE y Logloss, y sólo separabilidad/ordenación, medida por AUC). Desde el punto de vista de las métricas MSE, Logloss y AUC en general (ver Tabla 9.17 y 9.18) se confirman las expectativas que se tenían al elegir los oráculos, para estos conjuntos de datos la RN es un mejor clasificador que un NB, es

decir como la RN tiene un menor MSE, menor Logloss y mayor AUC que un NB en todas las baterías de experimentos, por lo que la RN es un clasificador más calibrado y refinado que un NB.

Tabla 9.17 Resumen de las métricas del error cuadrático medio (MSE), Logloss y área debajo de la curva ROC (AUC) usando NB como oráculo.

Modelo	Balanceado			Mayoritaria			Minoritaria		
	MSE	Logloss	AUC	MSE	Logloss	AUC	MSE	Logloss	AUC
Mim0a	0.468	1.549	0.751	0.498	1.720	0.681	0.498	1.718	0.681
Mim0b	0.468	1.546	0.752	0.497	**1.682**	0.689	0.497	**1.685**	0.688
Mim0c	0.496	1.817	0.742	0.505	1.739	0.692	0.505	1.737	0.693
Mim1Dira	0.467	1.565	**0.756**	0.495	1.744	0.695	0.495	1.740	0.695
Mim1Dirb	0.467	1.562	**0.756**	**0.493**	1.701	**0.697**	**0.494**	1.703	**0.697**
Mim1Dirc	0.489	1.744	0.752	0.501	1.729	0.696	0.501	1.729	**0.697**
Mim1Orda	0.467	1.565	**0.756**	0.495	1.744	0.695	0.495	1.740	0.695
Mim1Ordb	0.467	1.562	**0.756**	**0.493**	1.701	**0.697**	**0.494**	1.703	**0.697**
Mim1Ordc	0.489	1.744	0.752	0.501	1.729	0.696	0.501	1.729	**0.697**
Mim1ROCa	0.467	1.565	**0.756**	0.495	1.744	0.695	0.495	1.740	0.695
Mim1ROCb	0.467	1.562	**0.756**	**0.493**	1.701	**0.697**	**0.494**	1.703	**0.697**
Mim1ROCc	0.489	1.744	0.752	0.501	1.729	0.696	0.501	1.729	**0.697**
Mim2Dira	0.508	2.256	0.721	0.502	2.066	0.665	0.552	2.347	0.692
Mim2Dirb	0.508	2.258	0.722	0.500	2.008	0.669	0.555	2.373	0.692
Mim2Dirc	0.541	2.584	0.706	0.508	2.072	0.668	0.574	2.475	0.691
Mim2Orda	0.535	1.984	0.712	0.512	2.158	0.657	0.608	2.809	0.667
Mim2Ordb	0.534	1.987	0.712	0.513	2.222	0.645	0.588	2.576	0.669
Mim2Ordc	0.547	2.132	0.689	0.535	2.399	0.625	0.580	2.523	0.657
Mim2ROCa	0.500	2.138	0.729	0.500	1.938	0.671	0.572	2.696	0.685
Mim2ROCb	0.500	2.143	0.729	0.500	1.957	0.670	0.553	2.388	0.691
Mim2ROCc	0.522	2.393	0.712	0.517	1.975	0.667	0.560	2.547	0.684
Mim3	**0.465**	**1.488**	0.748	0.498	1.734	0.677	0.506	1.703	0.690
NB	0.379	0.958	0.821	0.410	1.009	0.761	0.410	1.009	0.761

Tabla 9.18 Resumen de las métricas del error cuadrático medio (MSE), Logloss y área debajo de la curva ROC (AUC) usando RN como oráculo.

Modelo	Balanceado			Mayoritaria			Minoritaria		
	MSE	Logloss	AUC	MSE	Logloss	AUC	MSE	Logloss	AUC
Mim0a	0.305	0.639	**0.871**	0.403	1.006	0.764	0.402	1.001	0.767
Mim0b	0.304	**0.633**	**0.871**	0.421	1.059	0.766	0.419	1.047	0.768
Mim0c	0.310	0.678	0.867	0.405	**1.000**	0.760	0.405	0.995	0.762
Mim1Dira	0.304	0.642	**0.871**	0.402	1.019	0.768	**0.401**	1.013	0.769
Mim1Dirb	**0.303**	0.636	**0.871**	0.416	1.058	**0.769**	0.414	1.046	**0.771**
Mim1Dirc	0.304	0.649	0.870	**0.401**	0.996	0.766	**0.401**	**0.994**	0.767
Mim1Orda	0.304	0.642	**0.871**	0.402	1.019	0.768	**0.401**	1.013	0.769
Mim1Ordb	**0.303**	0.636	**0.871**	0.416	1.058	**0.769**	0.414	1.046	**0.771**
Mim1Ordc	0.304	0.649	0.870	**0.401**	0.996	0.766	**0.401**	**0.994**	0.767
Mim1ROCa	0.304	0.642	**0.871**	0.402	1.019	0.768	**0.401**	1.013	0.769
Mim1ROCb	**0.303**	0.636	**0.871**	0.416	1.058	**0.769**	0.414	1.046	**0.771**
Mim1ROCc	0.304	0.649	0.870	**0.401**	0.996	0.766	**0.401**	**0.994**	0.767
Mim2Dira	0.309	0.671	0.870	0.404	1.160	0.750	0.431	1.265	0.765
Mim2Dirb	0.308	0.664	**0.871**	0.412	1.164	0.752	0.451	1.346	0.767
Mim2Dirc	0.313	0.721	0.866	0.412	1.194	0.743	0.433	1.249	0.761
Mim2Orda	0.484	1.536	0.805	0.460	1.367	0.720	0.617	2.438	0.711
Mim2Ordb	0.483	1.538	0.804	0.464	1.453	0.700	0.581	2.184	0.726
Mim2Ordc	0.503	1.598	0.788	0.464	1.423	0.711	0.613	2.418	0.703
Mim2ROCa	0.308	0.660	0.870	0.403	1.087	0.757	0.448	1.451	0.763
Mim2ROCb	0.308	0.658	0.870	0.412	1.137	0.754	0.454	1.388	0.767
Mim2ROCc	0.307	0.658	0.868	0.407	1.138	0.749	0.443	1.282	0.761
Mim3	0.339	0.760	0.867	0.413	1.047	0.755	0.486	1.362	0.758
RN	0.271	0.826	0.903	0.342	0.877	0.835	0.343	0.885	0.834

En las tablas 9.17 y 9.18 parece observarse que los Mim1 funcionan mejor que los Mim2 en términos de calibración. Esto es esperable, porque los Mim1 no modifican el oráculo y lo aprovechan sin variación del umbral. El umbral se cambia a posteriori (sobre el mimético) y, los cambios de umbral no afectan al AUC y afectan menos al MSE y al Logloss que al porcentaje de aciertos (y muchísimo menos que al coste, lógicamente).

Lo importante de las tablas 9.17 y 9.18 es que no hay una pérdida significativa en calibración (y por supuesto, ninguna en separabilidad) respecto a Mim0, el caso que no tiene en cuenta

el coste, exceptuando los Mim2Ord donde esta pérdida sí que es significativa.

En resumen, desde un punto de vista del coste, el método del umbral por ordenamiento ha resultado ganador. A pesar de que el método ROC es el más especializado, no es muy bueno en coste, esto es debido a que las probabilidades estimadas no están calibradas. Sin embargo, al considerar tanto coste como precisión hemos visto que el método que tiende a ser mejor es el método directo.

9.7. Conclusiones

En este capítulo hemos planteado cómo adaptar un modelo existente a un nuevo contexto de coste, con las mínimas suposiciones posibles: no hay datos de entrenamiento ni de validación, no hay conocimiento de la distribución de los datos (sólo máximos y mínimos de los atributos), no suponemos (en general) que el modelo existente calibre bien las probabilidades, etc. Hemos presentado diferentes métodos para abordar el problema y los hemos analizado teórica y experimentalmente. Como resultado podemos afirmar que las técnicas introducidas son útiles para reducir los costes del modelo, siendo superiores a las aproximaciones clásicas de sobremuestreo. La recomendación ante estos resultados generales es que cuando el objetivo sea solamente minimizar el coste global se recomienda utilizar el método a priori (es decir, Mim2, véase Figura 9.8) con el umbral por ordenamiento y un conjunto de datos inventados de forma balanceada empleando la distribución uniforme (Mim2Ordb). Demostramos que la comprensibilidad de este modelo aumenta cuando el *skew* aumenta y los árboles generados son diferentes de acuerdo al contexto del *skew*, obteniéndose una explicación del contexto en forma de reglas.

Si se desea obtener menos pérdida de precisión por unidad de ganancia de coste el mejor modelo es también un Mim2 pero con un umbral directo y empleando un conjunto de datos inventados a priori (Mim2Dira).

Una desventaja de los modelos miméticos es que bajan su precisión al ganar en coste al oráculo y dado que existe una garantía teórica de que la precisión no disminuye cuando los clasificadores se calibran (Cohen y Goldszmidt, 2004), además de

que existe la posibilidad de incrementar su precisión, sería entonces interesante estudiar en un futuro los métodos miméticos realizando su calibración, pensamos que así el método ROC podría ser mejor que los otros dos. El problema es que en la situación que contemplamos en este capítulo, no disponemos de datos de entrenamiento o de validación y, por tanto, no podemos calibrar.

Finalmente, otro aspecto a considerar en el futuro es que en el *skew* usado para los experimentos no hemos incluido la distribución de clases (como sí es habitual en el análisis ROC). Los ejemplos inventados los hemos generado para "forzar" distintas distribuciones. Si la distribución de aplicación se supiera, lo más lógico sería usar esta información y derivar un *skew*' a partir del *skew* y la distribución de clases.

Capítulo 10

Conclusiones y trabajo futuro

Si recordamos, el objetivo general de la tesis es estudiar y mejorar el método mimético para que permita la extracción de conocimiento, la contextualización, la revisión de conocimiento y la fusión de modelos, sin necesidad de analizar internamente la fuente de datos o el modelo a adaptar o integrar, y proporcionando modelos finales en forma de reglas.

Este objetivo general se ha cumplido a través de los capítulos de esta tesis.

El objetivo de estudiar el método mimético lo hemos realizado con: el análisis teórico del método mimético efectuado en el capítulo 4, el estudio de diferentes factores que afectan el método mimético incluyendo los datos de entrenamiento originales en el capítulo 5, en el capítulo 6 estudiamos el método mimético sin incluir los datos de entrenamiento.

Para cumplir con el objetivo de mejorar el método mimético en concreto se realizo: la propuesta de una métrica de calidad en el capítulo 5 que permite negociar precisión y número de reglas, también, el empleo del nivel de confianza para obtener el conjunto de datos inventados en los capítulos 5 y 9,

Extendimos la aplicación del método mimético al área de la extracción de conocimiento realizamos una modificación del principio MML la cual nos sirvió de base para proponer un pro-

cedimiento práctico para la obtención de conocimiento del experto o expertos minimizando el número de preguntas a realizarles.

Una de las principales áreas de aplicación que descubrimos fue el área de la contextualización de modelos. En el capítulo 7 tratamos la contextualización de los modelos debido a la aparición de nuevos datos, esta aplicación esta dentro de la revisión de modelos. Continuando con la apliación de las fronteras de la aplicación del método mimético en el capítulo 8 descubrimos la utilidad del método mimético para la contextualización de los modelos cuando existen cambios dentro del formato de los datos, es decir cuando los atributos o valores de clases presentan nuevos valores o su desaparición. Un de los últimos descubrimientos de aplicación del método mimético ha sido la contextualización de modelos a diferentes contextos de costes en el capítulo 9. A continuación describimos en más detalle las contribuciones de esta tesis.

10.1. Conclusiones

La primera parte de las aportaciones ha consistido en analizar teóricamente el marco mimético. Dicho análisis teórico no se había realizado hasta la fecha. A partir de este análisis, realizado en el capítulo 4, hemos corroborado algunos resultados que se conocían anteriormente, pero sólo por vía experimental, y hemos aportado algunos resultados novedosos sobre el método mimético. Bajo algunas suposiciones razonables se ha mostrado que es posible aproximar un clasificador a un oráculo tanto como se quiera. Las suposiciones razonables son: el oráculo no ha de ser fractal y el mimetizador puede ajustarse al detalle que se quiera. Por ejemplo, un árbol de decisión no podado (con un conjunto de datos inventados arbitrariamente grande) puede alcanzar el 100% de fidelidad respecto del oráculo. Además, hemos mostrado teóricamente que es útil agregar el conjunto de entrenamiento al conjunto de datos aleatorios para la segunda etapa.

Los casos donde el método mimético no es bueno dependen del modelo, pero la única situación donde el mimético puede ser desestimado es cuando la precisión del oráculo es peor que la

precisión del clasificador mimético, generado usando únicamente el conjunto de datos de entrenamiento original.

En un escenario mimético útil, por contra, el primer algoritmo de aprendizaje debe ser rápido. El segundo algoritmo de aprendizaje debe ser altamente preciso para conjuntos de datos grandes.

Hemos cumplido con el objetivo de analizar los límites teóricos y caracterizar el comportamiento de la técnica mimética en base a curvas de aprendizaje para configuraciones del método con y sin considerar los datos de entrenamiento. Esto es especialmente oportuno para la revisión de modelos, extracción de conocimiento para la construcción de sistemas expertos y la contextualización de los modelos a costes, todas ellas aplicaciones del método mimético no estudiadas con anterioridad.

Tanto los resultados teóricos como los empíricos ligan necesariamente el número de ejemplos con la precisión y la fidelidad del modelo. Esto, como se ha visto, está relacionado con el tamaño del modelo mimético y, por lo tanto, el número de reglas y su comprensibilidad.

La segunda parte de la tesis (capítulos 5 y 6) ha abordado los estudios experimentales con dos configuraciones principales (con y sin datos de entrenamiento originales). Se propuso una métrica de calidad para considerar precisión y comprensibilidad de un modelo de clasificación. Esta métrica se evaluó primeramente para los modelos miméticos usando un conjunto de datos que incluía los datos originales de entrenamiento de manera similar a cómo la técnica mimética fue introducida por Domingos. La reducción del número de reglas exigió sistemáticamente una reducción de la precisión y viceversa. Sin embargo, el estudio de factores tales como el tamaño del conjunto de datos inventados y la modificación de la distribución de ejemplos (por selección y repetición usando la confianza del oráculo) se ha mostrado herramientas útiles para mantener mejoras significativas en precisión que perceptiblemente reducen el número de reglas, mejorando por lo tanto, la métrica de calidad.

Adicionalmente, el análisis empírico del método mimético utilizando los datos originales ha dado otros resultados interesantes. El uso de la confianza del oráculo para modificar la distribución de los ejemplos es una de las nuevas contribuciones

principales de esta tesis y sugiere que el aumento en el número de reglas puede ser parcialmente debido al sobreajuste a los ejemplos de baja confianza etiquetados por el oráculo.

En el capítulo 6 hemos analizado un escenario sin datos de entrenamiento originales, situación no abordada anteriormente (al menos desde este punto de vista). Si bien esta situación es más sencilla que la vista en el capítulo 5, este escenario corresponde a muchas situaciones donde la adquisición del conocimiento se hace con preguntas simples, sobre el valor de la clase de los ejemplos a uno o más expertos. Hemos obtenido una aproximación práctica, fácil de implementar y general que minimiza las preguntas al experto. Es posible aplicarla especialmente en problemas donde los casos están bien estructurados (definidos por medio de una tabla de atributos/clase) y además el modelo del experto es generado con muchas reglas que lo hacen incomprensible, como por ejemplo en diagnosis.

Con los resultados generales obtenidos en el capítulo 6 para el estudio empírico del método mimético sin contar con los datos originales, hemos ideado una metodología para estimar el número de casos necesarios para obtener el modelo "óptimo", en términos del compromiso entre la precisión, el número de preguntas a realizar al experto y el tamaño del modelo. Hemos visto que, generalmente, podemos aproximar este valor de forma razonable con sólo tres iteraciones, lo que significa que pediremos algunos casos al experto en una primera entrevista, y en la segunda o tercera entrevista sabremos casi exactamente el número de casos requeridos. Estos resultados y la metodología propuesta son un paso adelante en la adquisición de conocimiento a través del aprendizaje computacional de forma mucho más práctica y fácil. Este paso puede ayudar a solucionar el cuello de botella de la adquisición de conocimiento.

La tercera y última parte de la tesis ha extendido la técnica mimética a problemas de adaptación, contextualización y revisión de modelos.

Hasta la fecha, el modelo mimético no se había utilizado de una manera genérica para adaptarse a diferentes contextos. Hemos extendido el uso del método mimético para su aplicación al área de contextualización; En concreto a la revisión de modelos accionada por novedad (aparece una nueva zona de datos no existentes originalmente) o por cambios de formato

(aparecen o desaparecen atributos o clases) y a la clasificación sensible al coste.

En concreto, en el capítulo 7 hemos presentado un nuevo método general para la revisión de modelos basado en un uso balanceado del modelo antiguo (a través de la generación de un conjunto de datos inventados) y los nuevos datos. Todo junto sirve para obtener un modelo mimético. Hemos demostrado que el modelo mimético se comporta mejor que el modelo antiguo, un nuevo modelo y su combinación en escenarios donde los cambios suceden en las regiones que no eran relevantes o eran inexistentes en la situación original (escenario de novedad). Éste es especialmente el caso cuando tenemos pocos datos nuevos disponibles para conseguir entrenar un buen modelo nuevo.

Un escenario diferente sucede cuando los cambios ocurren de una manera más global y los nuevos datos contradicen, en cierto modo, la situación original. Aquí, los casos donde es útil la técnica mimética no son tan generales como en el escenario anterior. En cualquier caso, el modelo mimético se puede construir muy fácilmente y se puede comparar con los antiguos y nuevos modelos para considerar si el modelo mimético es eventualmente mejor o no. De esta forma, hemos demostrado que cualquier técnica de aprendizaje computacional se puede utilizar como método general de revisión de modelos, independientemente de cómo se generó el modelo y sin necesidad de disponer de los datos ni de la distribución original de los mismos.

En el caso de la revisión de modelos cuando aparece un nuevo atributo importante, es necesario que la importancia del atributo que aparece realmente sea muy importante. El mejor modelo mimético para este escenario el esquema simple de aplicarlo sin utilizar sobremuestreo.

Para el caso de la revisión de modelos cuando aparece un nuevo valor de clase, también hemos demostrado que el método mimético es útil cuando nos encontramos en una etapa temprana del cambio de contexto, es decir, nuevamente cuando el conjunto de datos nuevos tiene un tamaño pequeño.

También hemos planteado cómo adaptar un modelo existente a un nuevo contexto de coste, con las mínimas suposiciones posibles: no hay datos de entrenamiento ni de validación, no hay conocimiento de la distribución de los datos (sólo máximos y

mínimos de los atributos), no suponemos (en general) que el modelo existente calibre bien las probabilidades, etc. Hemos presentado diferentes métodos para abordar el problema y los hemos analizado teórica y experimentalmente. Como resultado podemos afirmar que las técnicas introducidas son útiles para reducir los costes del modelo, siendo superiores a las aproximaciones clásicas de sobremuestreo. Se realizaron recomendaciones de configuraciones del método mimético para lograr minimizar el coste global del modelo u obtener menos pérdida de precisión por unidad de ganancia de coste.

10.2. Contribuciones de la tesis

En esta tesis hemos conseguido que la técnica mimética permita agilizar y mejorar los procesos de adquisición, adaptación y mantenimiento del conocimiento en áreas en que este proceso no se realiza de forma automática o se trata de procesos muy caros en términos de tiempo o dinero. También facilitamos la reutilización de los modelos al permitir su adaptación, contextualización y revisión. La gran variedad de aplicaciones que encontramos nos induce a pensar que la técnica mimética es general y muy útil en el mantenimiento, conservación y tranformación del conocimiento.

En resumen se lograron las siguientes contribuciones de la tesis:

1. Demostramos teóricamente que se puede aproximar totalmente un clasificador a un buen oráculo usando la técnica mimética.
2. Hemos mostrado que la técnica mimética permite extraer conjuntos de reglas cortos a partir de cualquier modelo de caja negra, y usar este conjunto de reglas como la explicación de la caja negra. Una buena combinación de factores del método genera buenos resultados en el compromiso de precisión y comprensibilidad.
3. Estudiamos el compromiso entre precisión y comprensibilidad de los modelos miméticos, que permita maximizarlos de forma conjunta. Se aportaron dos marcos de trabajo para evaluar la calidad del compromiso entre comprensibilidad y precisión de los modelos miméticos.
4. Demostramos que cualquier técnica de aprendizaje computacional se puede utilizar como método general

de revisión de modelos si se emplea el esquema mimético, independientemente de cómo se generó el modelo y sin necesidad de disponer de los datos ni de la distribución original de los mismos.

5. Encontramos que la técnica mimética se puede utilizar en una etapa temprana del cambio de contexto de un modelo, cuando el conjunto de datos nuevos tiene un tamaño pequeño. Esto es útil en los problemas de cambio de contexto de los modelos al aparecen nuevos datos o al cambio del formato de los datos.
6. Estudiamos teórica y experimentalmente configuraciones del método mimético que permiten adaptar un modelo existente a un nuevo contexto de coste, con las mínimas suposiciones posibles. Pudiéndose maximizar el coste global u obtener menos pérdida de precisión por unidad de ganancia de coste.
7. Ayudamos a resolver el cuello de botella de la extracción del conocimiento para la construcción de sistemas expertos de una manera más eficiente. Para ello, propusimos una metodología para estimar el número de los casos necesarios para obtener el modelo "óptimo", en términos del compromiso entre la precisión, el número de preguntas a realizar al experto y el tamaño del modelo.
8. Programamos diferentes implementaciones del método mimético utilizando la paquetería Weka.
9. Hemos estudiado otras formas de generar los datos inventados las cuales permitieron mejorar el comportamiento del método mimético y descubir nuevas aplicaciones del mismo.

En esta tesis hemos consolidado a la técnica mimética como una técnica general para la extracción de reglas comprensibles a partir de cualquier oráculo. Donde el oráculo no sólo es una combinación de clasificadores (como anteriormente existía), sino que ahora se ha demostrado que se puede también aplicarse a las redes neuronales, a expertos o conjuntos de expertos, o cualquier clasificador considerado como caja negra, en general usar cualquier modelo que tenga muy buena precisión pero con problemas de comprensibilidad. Hemos contribuido a que el método mimético sea el primer método general para convertir cualquier modelo de caja negra en un modelo simple. Consecuentemente, con el conocimiento aprendido de la técnica mimética hemos descubierto aplicaciones que antes no se habían detectado. Estas aplicaciones facilitan la extracción de

conocimiento, la adaptación, contextualización y revisión de modelos.

Nuestras contribuciones principales han sido contrastadas y difundidas a través de las siguientes publicaciones:

- Blanco-Vega R., Hernández-Orallo J., Ramírez-Quintana Mª J. (2004a) Analysing the Trade-off between Comprehensibility and Accuracy in Mimetic Models. The 7th International Conference on Discovery Science, DS 2004, LNAI 3245, pp. 338-346.

- Blanco-Vega R., Hernández-Orallo J., Ramírez-Quintana Mª J. (2004b). El Modelo Mimético, una Alternativa para la Comprensibilidad de Modelos de "Caja Negra". Red Española de Minería de Datos y Aprendizaje.

- Blanco-Vega R., Hernández-Orallo J., Ramírez-Quintana Mª J. (2005a). La Técnica Mimética en Ausencia de Datos Originales: Aprendizaje y Revisión de Modelos. III Taller Nacional de Minería de Datos y Aprendizaje, TAMIDA'2005. Taller integrado en el I Congreso Español De Informática (CEDI'2005).

- Blanco-Vega R., Hernández-Orallo J., Ramírez-Quintana Mª J. (2005b) Knowledge Acquisition through Machine Learning: Minimising Expert's Effort. IEEE, The Fourth International Conference on Machine Learning and Applications (ICMLA'05). Los Angeles, CA, USA.

- Hernández Orallo J., Blanco Vega R., Cèsar Ferri Ramírez, Ramírez Quintana Mª J. (2005). Modelos en minería de datos: interpretación, combinación y adaptación. XV Escuela de Verano de Informática "Jornadas Técnicas de Minería de Datos".

- Blanco-Vega R., Hernández-Orallo J., Ramírez-Quintana Mª J. (2006). La Técnica Mimética en Ausencia de Datos Originales: Aprendizaje y Revisión de Modelos. Revista Iberoamericana de Inteligencia Artificial.

- Blanco Vega R., Ferri Ramírez Cèsar, Hernández Orallo J., Ramírez Quintana Mª J. (2006). Estimating the Class Probability Threshold without Training Data. ROC

Analysis in ML, 3rd International Workshop, ROCML-2006, Workshop associated to ICML'2006, the 23rd International Conference on Machine Learning.

10.3. Trabajo futuro

El método mimético tiene muchas otras áreas donde poderse aplicar y más cuestiones que estudiar. Por ejemplo, existen muchas preguntas abiertas relacionadas con la metodología teórica. En seguida mostramos algunas de las cuestiones que estudiaremos en un futuro.

En el cálculo del tamaño óptimo del número de pregunts a realizar a un experto para extraerle su conocimiento quedan varias cuestiones abiertas. Primero, estudiar la posibilidad de agrupar casos similares por técnicas de agrupamiento y después pedir que el experto etiquete los grupos y no los casos individuales. También estudiar otros métodos el aprendizaje computacional y otras de sus tareas.

En revisión de modelos, primero, quisiéramos analizar más profundamente la influencia de los cambios en la distribución de la clase.

Estamos en la busqueda del descubrimiento de más nichos de aplicación de la técnica mimética. La técnica es especialmente beneficiosa cuando se utiliza en escenarios donde la representación del problema cambia (nuevos atributos, nuevos valores, nuevas clases, etc.).

En la contextualización por aparición de un nuevo atributo sería interesante estudiar la asignación de los valores indeterminados con otros métodos diferentes al de la media.

Otro aspecto a considerar en el futuro es que en el *skew* usado para los experimentos de contextualización al coste no hemos incluido la distribución de clases (como sí es habitual en el análisis ROC). Los ejemplos inventados los hemos generado para "forzar" distintas distribuciones. Si la distribución de aplicación se supiera, lo más lógico sería usar esta información y derivar un *skew*' a partir del *skew* y la distribución de clases.

Una desventaja de los modelos miméticos es que bajan su precisión al ganar en coste al oráculo y dado que existe una garantía teórica de que la precisión no disminuye cuando los clasificadores se calibran (Cohen y Goldszmidt, 2004), además de que existe la posibilidad de incrementar su precisión, sería entonces interesante estudiar en un futuro los métodos miméticos realizando su calibración, pensamos que así el método ROC podría ser mejor que los otros dos. El problema es que en la situación que contemplamos en este capítulo, no disponemos de datos de entrenamiento o de validación y, por tanto, no podemos calibrar.

Finalmente, otra cuestión a estudiar es la selección de instancias. Los métodos de selección de instancias podían ser útiles para reducir el tamaño del conjunto de datos inventados. Estos métodos que disminuyen la cantidad de datos se denominan selección de instancias (Brighton y Mellish, 2002; Kibbler y Aha, 1987; Liu y Motoda; 2001). El uso de la selección de instancias junto con la técnica mimética no se ha explorado hasta la fecha, aunque tienen bastante similitud. Esto también es necesario debido a que en muchas técnicas de aprendizaje y, en especial, en los árboles de decisión, existe una relación lineal entre el tamaño del conjunto de datos de entrenamiento y el número de reglas (Oates y Jensen, 1997; Blanco et al, 2005b), así reduciendo el número de datos reduciremos el número de reglas, manteniendo una precisión adecuada.

Referencias

Andrews, R.; Diederich, J., Tickle, A.B. (1995). Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge-Based Systems, 8(6), 373-389.

Abe, Naoki; Zadrozny, Bianca; Langford, John. (2004). An Iterative Method for Multi-class Cost-sensitive Learning. KDD'04, August 22–25, Seattle, Washington, USA.

Angluin. D. (1987) Queries and concept learning. Machine Learning, 2:319-342.

Baeza-Yates, R.; Ribeiro-Neto, B. (1999). Modern Information Retrieval. Addison-Wesley.

Black C. L.; Merz C. J. (1998) UCI repository of machine learning databases

Blanco Vega R., Ferri Ramírez Cèsar, Hernández Orallo J., Ramírez Quintana Ma J. (2006). Estimating the Class Probability Threshold without Training Data. ROC Analysis in ML, within ICML'2006, the 23rd International Conference on Machine Learning.

Blanco-Vega R., Hernández-Orallo J., Ramírez-Quintana Ma J. (2004a) Analysing the Trade-off between Comprehensibility and Accuracy in Mimetic Models. The 7th International Conference on Discovery Science, DS 2004, LNAI 3245, pp. 338-346.

Blanco-Vega R., Hernández-Orallo J., Ramírez-Quintana Ma J. (2004b). El Modelo Mimético, una Alternativa para la Comprensibilidad de Modelos de "Caja Negra". Red Española de Minería de Datos y Aprendizaje.

Blanco-Vega R., Hernández-Orallo J., Ramírez-Quintana Ma J. (2005a). La Técnica Mimética en Ausencia de Datos Originales: Aprendizaje y Revisión de Modelos. III Taller Nacional de Minería de Datos y Aprendizaje, TAMIDA'2005. Taller integrado en el I Congreso Español De Informática (CEDI'2005).

Blanco-Vega R., Hernández-Orallo J., Ramírez-Quintana Ma J. (2005b) Knowledge Acquisition through Machine Learning: Minimising Expert's Effort. IEEE, The Fourth International Conference on Machine Learning and Applications (ICMLA'05). Los Angeles, CA, USA.

Blanco-Vega R., Hernández-Orallo J., Ramírez-Quintana Ma J. (2006). La Técnica Mimética en Ausencia de Datos Originales: Aprendizaje y Revisión de Modelos. Revista Iberoamericana de Inteligencia Artificial.

Blockeel, H.; Struyf, J. (2001). Frankenstein classifiers: Some experiments, Proceedings of the Eleventh Belgian-Dutch Conference on Machine Learning (Hoste, V. and De Pauw, G., eds.), pp. 5-12.

Bouckaert, R. R. (2004). Estimating replicability of classifier learning experiments. In Proceedings of the Twenty-First international Conference on Machine Learning (Banff, Alberta, Canada, July 04 - 08). ICML '04, vol. 69. ACM Press, New York, NY, 15.

Bouckaert, Remco R. (2003). Choosing Learning Algorithms Using Sign Tests with High Replicability, Lecture Notes in Computer Science, Volume 2903, Pages 710 – 722.

Bouckaert, Remco R. (2005). Low Replicability of Machine Learning Experiments is not a Small Data Set Phenomenon. ICML Meta Learning workshop.

Bouckaert, Remco R.; Frank; Eibe. (2004). Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms, Lecture Notes in Computer Science, Volume 3056, Page 3

Boz, O. (2000). Converting A Trained Neural Network To A Decision Tree Dectext Decision Tree Extractor, Thesis, Computer Science and Engineering Department, Lehigh University.

Breiman L. Bagging predictors. (1996) Machine Learning, 24(2):123-140.

Breiman, L.; Friedman, J. H.; Olsen, R. A.; Stone, C. J. (1984) Classification and Regression Trees. Wadsworth International Group.

Breiman, L.; Shang, N. (1997). Born again trees. Technical Report, Berkeley, CA: Statistics Department, University of California at Berkeley.

Bridewell, W.; Bani Asadi, N.; Langley, P.; Todorovski, L. (2005). Reducing overfitting in process model induction. Proceedings of the Twenty-Second International Conference on Machine Learning, 81–88.

Brier, G. W. (1950). Verification of Forecasts Expressed in Terms of Probability. Monthly Weather Review, 75, 1-3.

Brighton H. y Mellish C. (2002). Advances in instance selection for instance-based learning algorithms. Data Mining and Knowledge Discovery 6: 153-172.

Brulé, J. y Bount, A. (1989). Knowledge Acquisition. McGraw-Hill. New York.

Cain, T. (1991). The DUCTOR: A theory revision system for propositional domains. In Proceedings of the Eighth International Workshop on Machine Learning, 485-489.

Cestnik, G.; Kononenko, I.; Bratko, I. (1987). "Assistant 86: A Knowledge Acquisition Tool for Sophisticated Users" in Bratko, I.; Lavrac, N. (Eds.) Progress in Machine Learning, Sigma Press.

Chan, P.; Stolfo, S. (1998) Toward scalable learning with non-uniform class and cost distributions. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pages 164–168.

Clark, P.; Boswell, R. (1991). "Rule induction with CN2: Some recent improvements" in Proceedings of the Fifth European Working Session on Learning, pp. 151-163, Springer, Berlin.

Clark, P.; Niblett, T. (1989) "The CN2 induction algorithm" Machine Learning, 3(4): 261-283.

Cohen, I.; Goldszmidt, M. (2004). Properties and Benefits of Calibrated Classifiers, Lecture Notes in Computer Science, Vol. 3202, 125 – 136.

Craven, M. W.; Shavlik, J. W. (1994) Using Sampling and Queries to Extract Rules from Trained Neural Networks, Proc. of the 11th Int. Conf. on Machine Learning, pp: 37-45.

Craven, M.W.; Shavlik, J.W. (1996). Extracting tree-structured representations of trained networks. In Advances in Neural Information Processing Systems 8, Touretzky, D., Mozer, M., & Hasselmo, M., Eds. Cambridge, MA: MIT Press, 24-30.

Dietterich, T.G. (2000). "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization" Machine Learning, Vol 40(2), pp 139-157.

Dietterich, T.G. (2002). Ensemble learning. In The Handbook of Brain Theory and Neural Networks, 2nd edition, Arbib, M.A., Ed. Cambridge, MA: MIT Press.

Debenham, J. (1989). Knowledge System Design. Prentice Hall. Sidney.

Dempster, A.; Laird, N.; Rubin, D. (1977). "Maximun Likelihood from incomplete data via EM algorithm". Journal of the Royal Statistical Society B. 39:1—38.

Domingos, P. (1997a) Knowledge Acquisition from Examples Via Multiple Models. Proc. of the 14th Int. Conf. on Machine Learning, pp:98-106.

Domingos, P. (1997b). Learning Multiple Models without Sacrificing Comprehensibility, Proc. of the 14th National Conf. on AI, pp: 829.

Domingos, P. (1998). Knowledge Discovery Via Multiple Models. Intelligent Data Analysis, 2(1-4): 187-202.

Domingos, P. (1999). MetaCost: A general method for making classifiers cost sensitive. In Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, pages 155–164. ACM Press.

Dowe, D. L.; Farr, G. E.; Hurst, A. J.; Lentin, K. L. (1996). Information-theoretic football tipping. In 3rd Conf. on Maths and Computers in Sport, volume 14, pages 233–241.

Duda, R.O.; Hart, P.E. (1973). Pattern classification and scene analysis. John Wiley and Sons.

Efron, B. (1982) The Jackknife, the Bootstrap and Other Resampling Plans, Philadelphia: SIAM.

Egan, J. P.: (1975). Signal Detection Theory and ROC Analysis, Series in Cognitition and Perception. New York: Academic Press.

Elkan, Charles. (2001). The Foundations of Cost-Sensitive Learning. Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence.

Engelbrecht, A. P.; Schoeman, L. (2002). A Building Block Approach to Genetic Programming for Rule Discovery. in Data Mining: A Heuristic Approach, HA Abbass, R Sarkar, C Newton (eds), Chapter IX, pp 175-189, Idea Group Publishing

Estruch, V.; Ferri, C.; Hernández-Orallo, J.; Ramírez-Quintana, Ma J. (2003). Simple Mimetic Classifiers, Proc. of the Third Int. Conf. on Machine Learning and Data Mining in Pattern Recognition, LNCS 2734, pp:156-171.

Estruch, V.; Hernández-Orallo, J. (2003): Theoretical Issues of Mimetic Classifiers, TR DSIC, http://www.dsic.upv.es/~flip/papers/mim.ps.gz.

Fawcett, T. (2003) ROC Graphs: Notes and Practical Considerations for Data Mining Researchers. Technical Report HPL-2003-4, HP Labs.

Fayyad, U. M. (1994). “Branching on Attribute Values in Decision Tree Generation”. In Proc. Of the Twelfth National Confer-

ence on Artificial Intelligence AAAI-94, pages 601-606, Cambridge, MA, MIT Press.

Fayyad, U.M.; Piatetsky-Shapiro, G., Smyth, P. (1996). "From Data Mining to Knowledge Discovery: An Overview," Advances in Knowledge Discovery and Data Mining, U.M. Fayyad et al., eds.,AAAI/MIT Press, Menlo Park, Calif., pp. 1–34.

Feigenbaum, Edward A. (2003). Some challenges and grand challenges for computational intelligence. Journal of the ACM (JACM), 50:32-40.

Ferri, C.; Hernández-Orallo, J.; Ramírez-Quintana, M.J. (2001). "Learning MDL-guided Decision Trees for Constructor-Based Languages" in 11th International Conference on Inductive Logic Programming (ILP'2001), Work in Progress Track, Research Report 2001/12, LSIIT, 64 Bld Brant, 67400 Illkirch, France.

Ferri, C.; Hernández-Orallo, J.; Ramírez-Quintana, M. J.. (2002a). From Ensemble Methods to Comprehensible Models, Lecture Notes in Computer Science, Volume 2534, Jan 2002, Pages 165 – 177.

Ferri, C.; Flach, P.; Hernández-Orallo, J. (2002b). "Learning Decision Trees Using the Area Under the ROC Curve" in C. Sammut; A. Hoffman (eds.) The 2002 International Conference on Machine Learning, IOS Press, Morgan Kaufmann Publishers, ISBN: 1-55860-873-7, pp. 139-146.

Ferri, C.; Hernández-Orallo, J.; Salido, M. (2003). "Volume Under the ROC Surface for Multiclass Problems" in The 2003 European Conference on Machine Learning, ECML'03, LNAI, Springer Verlag.

Ferri, C.; Flach, P. A.; Hernández-Orallo, J. (2004). Delegating classifiers. ACM International Conference Proceeding Series; Vol. 69 archive. Proceedings of the twenty-first international conference on Machine learning ICML 2004. Banff, Alberta, Canada. Page: 37. ISBN:1-58113-828-5

Freund Y. and Schapire R.E. (1996). Experiments with a new Boosting algorithm. In Proc. 13th International Conference on Machine Learning, pages 148-146. Morgan Kaufmann.

Gallant, S.I. (1983). Connectionist expert systems. Communications of the ACM, 31(2), 152-169.

Gama, J.; Brazdil P. (2000). "Cascade Generalization," Machine Learning, vol. 41, pp. 315—343.

García Martínez, R. (1994). Adquisición de Conocimiento. En Abecasis, S. y

Ginsberg, A. (1990). Theory reduction, theory revision, and retranslation. In Proceedings of the Eighth National Conference on Artificial Intelligence, 777-782.

Gold, E.M. (1967). Language Identification in the Limit. Information and Control, 10:447-474.

Good, I. J. (1968). Corroboration, explanation, evolving probability, simplicity, and a sharpened razor. British Journal of the Philosophy of Science, 19:123–143.

Good, I. J. (1952). Rational decisions. Journal of the Royal Statistical Society, Series B, 14:107–114.

Greenwell, M. 1988. Knowledge Engineering for Expert Systems. Ellis Horwood Limited. Chichester.

Grover, M. (1983). A Pragmatic Knowledge Aquisition Methodology. Proceedings VIII IJCAI. Estados Unidos.

Hand, D.; Till, R. (2001). A simple generalisation of the Area Under the ROC Curve for multiple class classification problems. Machine Learning, 45, 171–186.

Haussler, D.; Kearns, M.; Opper, M.; Schapire, R. (1992). Estimating average-case learning curves using bayesian, statistical physics and vc dimension methods. Advances in Neural Information Processing Systems, 4:855-862.

Hernández Orallo J., Blanco Vega R., Cèsar Ferri Ramírez, Ramírez Quintana Ma J. (2005). Modelos en minería de datos: interpretación, combinación y adaptación. XV Escuela de Verano de Informática "Jornadas Técnicas de Minería de Datos".

Herrera, F.; Hervás, C.; Otero, J.; Sánchez, L. (2004). Un estudio empírico preliminar sobre los test estadísticos más habituales en el aprendizaje automático. Tendencias de la Minería de Datos en España. Red Española de Minería de Datos. Editores: Raúl Giráldez, José C. Riquelme, Jesús S. Aguilar-Ruiz.

Johnson, G.; Nealon, J. L.; Lindsay, R. O. (1996). Using relevance information in the acquisition of rules from a neural network, Proc. of the Workshop on Rule Extraction from Trained Artificial Neural Networks, pp: 68-80.

Kearns, M. and Mansour, Y. (1996). On the boosting ability of top-down decision tree learning algorithms. In Proceedings of the Twenty-Eighth Annual ACM Symposium on theory of Computing (Philadelphia, Pennsylvania, United States, May 22 - 24, 1996). STOC '96. ACM Press, New York, NY, 459-468.

Kibbler D.; Aha D. W. (1987) Learning representative exemplars of concepts: An initial case of study. En Proc. of the Fourth International Workshop on Machine Learning, páginas 24-30.

Kohavi, R. and Sommerfield, D. (1998). Targeting business users with decision table classifiers. Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, New York, NY: AAAI Press, pp. 249–253.

Kohavi, R., Wolpert, D. H. (1996). Bias Plus Variance Decomposition for Zero-One Loss Functions. Proceedings of the Thirteenth International Conference on Machine Learning. Morgan Kaufmann, San Francisco

Kohonen T. 1982. "A simple paradigm for the self-organized formation of structured feature maps". In M. Arbib: S. Amari (Eds.). Lecture notes in biomathematics 45: Competition and cooperation in neural nets, (pp. 248-266). Berlin. Springer-Verlag.

Kohonen T. 1984. Self-organization and associative memory, Springer-Verlag, Berlin.

Kononenko, I. (1990). Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition. In Current Trends in Knowledge Acquisition, B. Wielinga (Ed.). Amsterdam, The Netherlands: IOS Press.

Lachiche, N.; Flach, P. A. (2003). Improving Accuracy and Cost of Two-class and Multi-class Probabilistic Classifiers Using ROC Curves. ICML 2003: 416-423.

Langley, P.; Iba, W.; Thompson. K. (1992). "An Analysis of bayesian Classifiers". In Proceedings of the 10th National Conference on Artificial Intelligence. 223--223.

Langley, P. (1996). Induction of Condensed Determinations. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), pp. 327--330. Menlo Park, CA:AAAI Press

Li, M.; Vitányi, P. (1997). An Introduction to Kolmogorov Complexity and Its Applications, Springer Verlag; 2nd edition.

Liu, H.; Motoda, H. (2001) Instance Selection and Construction for Data Mining. Kluwer Academic Publishers.

Melville, P.; Mooney, R. J. (2005). Creating diversity in ensembles using artificial data. Information Fusion 6(1): 99-111.

Meyer, M. y Booker, J. (1991). Eliciting and Analyzing Expert Judgement. A

McClelland, J.; Rumelhart D. E. (1986). The PDP Research Group. Parallel Distributed Processing, Volume 1 and 2. MIT Press.

Oates, Tim; Jensen, David (1997). The Effects of Training Set Size on Decision Tree Complexity. Proceedings of the Fourteenth International Conference on Machine Learning: 254 – 262. Morgan Kaufmann Publishers Inc.

Ourston, D.; Mooney R. (1994). Theory refinement combining analytical and empirical methods. Volume 66, Issue 2, pp.273-309 Mooney Elsevier Science Publishers Ltd.

Parmanto B; Munro P.; Doyle H. (1996). "Improving Committee Diagnosis with Resampling Techniques" Advances in Neural Information Processing Systems, Vol 8, pp 882-888.

Pazzani, M.; Mani, S.; Shankle, W. R. (1997). Beyond concise and colorful: Learning intelligible rules. Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA: AAAI Press, pp. 235–238.

Pazzani, M. J. 1991. Influence of prior knowledge on concept acquisition: Experimental and computational results. Journal of Experimental Psychology: Learning, Memory, and Cognition, 17:416–432.

Pazos Sierra, J. (1996). Introducción a la Ingeniería del Conocimiento. Unidad

Plant, Richard E.; Stone, Nicholas, (1991). Knowledge-based Systems in Agriculture. McGraw-Hill.

Pop, E.; Hayward, R.; Diederich, J. (1994). RULENEG: Extracting Rules From a Trained ANN by Stepwise negation, technical report, QUT NRC.

Powell, M. J. 1987. Radial basis functions for multivariable interpolation: a review. In Algorithms For Approximation, J. C. Mason and M. G. Cox, Eds. Clarendon Press Institute Of Mathematics And Its Applications Conference Series. Clarendon Press, New York, NY, 143-167.

Provost, F.; Domingos, P. (2003).Tree induction for probability-based rankings, Machine Learning, 52 (3): 199-215.

Provost, Foster y Fawcett, Tom, (2001). Robust Classification for Imprecise Environments, Machine Learning, Volume 42, Issue 3, Pages 203 - 231

Quinlan, J.R. (1983). "Learning Efficient Classification Procedures and Their Application to Chess End Games" in Michalski, R.; Carbonell, J.; Mitchell, T. (Eds.) Machine Learning: An Artificial Intelligence Approach. Morgan Kaufmann, San Mateo, CA.

Quinlan, J.R. (1986). "Induction of Decision Trees". Machine Learning, 1, 81-106.

Quinlan, J. Ross. (1993). C4.5: Programs for machine learning, Morgan Kaufmann Publishers.

Quinlan, J.R. (1996) “Bagging, Boosting, and C4.5” in Proceedins of the 30th National Conference on Artificial Intelligence and the 8th Innovative Applications of Artificial Intelligence, pp 725-730.

Ripley, B.D. (1996) Pattern Recognition and Neural Networks, Cambridge: Cambridge University Press.

Rissanen, J. (1978). Modeling by shortest data description. Automatica, 14:465--471.

Rosenblatt, F. (1962). Principles of Neurodynamics. Spartan Books, New York.

Saito, K.; Nakano, R. (1988), "Medical diagnostic expert system based on PDP model". Proceedings of IEEE International conference on Neural Networks, (vol 1), pp.255-262, San Diego, CA:IEEE.

Saito, K.; Nakano, R., (1990), "Rule extraction from facts and neural networks". INNC-90 Paris: proceedings of the international neural network conference. Paris, France. July 9-13, 1990. Volume 1 ISBN: 0 79230831 X Dordrecht, The Netherlands: Kluwer pp. 379-382.

Salton, G.; McGill, M.J. (1983). Introduction to modern information retrieval. McGraw Hill, New York.

Shepard, D. 1968. A two-dimensional interpolation function for irregularly-spaced data. In Proceedings of the 1968 23rd ACM National Conference (August 27 - 29, 1968). ACM Press, New York, NY, 517-524.

Srinivasan, A. (1999). “Note on the Location of Optimal Classifiers in N-dimensional ROC Space” Technical Report PRG-TR-2-99, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford.

Stone, M. (1978). Cross-validation: A review. Mathematische Operationsforschung Statischen, Serie Statistics, 9 127-139.

Swets, J. A.; Dawes, R. M.; Monahan, J. (2000). Better Decisions through Science. Scientific American 283, 82-87.

Taha, I.; Ghosh, J. (1996) Three techniques for extracting rules from feedforward networks, Intelligent Engineering Systems Through Artificial Neural Networls, 6: 23-28.

Tickle, A.B.; Andrews, R.; Golea, M.; Diederich, J., (1998). "The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks," Neural Networks, IEEE Transactions on vol.9, no.6 pp.1057-1068.

Utgoff, P. (1994). An Improved Algorithm for Incremental Induction of Decision Trees. In Proc. 11th ICML, pages 318-325. Morgan Kaufmann.

Wallace, C.S., Boulton, D.M. (1968). An information measure for classification. Computer Journal 11 185-194.

Wallace, C. S. and Dowe, D. L. (1999). Refinements of MDL and MML Coding. The Computer Journal 1999 42(4):330-337

Walpert, D.; Macready, W. (1997). No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, pages 67-82.

Widmer, G.; Kubat, M. (1996). Learning in the Presence of Concept Drift and Hidden Contexts. Machine Learning Journal.

Witten, I. H., & Frank, E. (2005). Data mining: Practical machine learning tools and techniques with java implementations. Morgan Kaufmann Publishers. 2nd edition.

Wolpert, D.H. (1992). "Stacked Generalization" Neural Networks, Vol 5, pp 241-259.

Yang, J., Parekh, R., Honavar, V. G., and Dobbs, D. (1999). Data-Driven Theory Refinement Using KBDistAl. In Proceedings of the Third international Symposium on Advances in intelligent Data Analysis D. J. Hand, J. N. Kok, and M. R. Berthold, Eds. Lecture Notes In Computer Science, vol. 1642. Springer-Verlag, London, 331-342.

Zadrozny, B.; Langford, J.; Abe, N. (2003). Cost-sensitive learning by cost-proportionate example weighting. In Proceedings of the Third IEEE International Conference on Data Mining, pages 435–442.

Zhang, T.; Oles F. (2000). A probability analisis on the value of unlabeled data in classification problems. In International Conference on Machine Learning.

Zheng, F. and G.I. Webb (2005). A Comparative Study of Seminaive Bayes Methods in Classification Learning. In Simoff, S.J., G.J. Williams, J. Galloway, and I. Kolyshkina (Eds.), Proceedings of the Fourth Australasian Data Mining Conference (AusDM05), pages 141-156. Sydney: University of Technology. Sydney, Australia.

Zhou, Z.-H.; Jiang, Y. (2003). Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble. IEEE Transactions on Information Technology in Biomedicine, 7(1), 37-42.

Zhou, Z.-H.; Jiang, Y. (2004). NeC4.5: neural ensemble based C4.5. IEEE Transactions on Knowledge and Data Engineering, 16(6), 770-773.

Zhou, Z.-H.; Jiang, Y.; Chen, S.-F. (2003). Extracting symbolic rules from trained neural network ensembles. AI Communications, 16(1), 3-15.