ESCUELA TÉCNICA SUPERIOR
DE INGENIERÍAS INFORMÁTICA Y TELECOMUNICACIÓN

MÁSTER EN CIENCIA DE DATOS E INGENIERÍA DE COMPUTADORES

UNIVERSIDAD DE GRANADA

TRABAJO DE FIN DE MÁSTER

COMBINACIÓN DE ÁRBOLES DE DECISIÓN BASADOS EN PROBABILIDADES IMPRECISAS. APLICACIÓN SOBRE BASES DE DATOS CON RUIDO

Autor: Alejandro Alonso Capel
D.N.I.: 76631004-H
e-mail: alejandroc@correo.ugr.es

Tutores:
Joaquín Abellán Mulero
Carlos Javier Mantas Ruiz

14 de septiembre de 2015

El autor de este Trabajo Fin de Máster asume la originalidad del mismo. Para su realización no se han utilizado fuentes sin citarlas debidamente. De esta forma, queda en consideración toda aquella propuesta o trabajo ajeno que ha sido integrado para ayudar a la elaboración de tal investigación.

Alejandro Alonso Capel

Granada, a 14 de septiembre de 2015

A Ángeles, Liborio, Marian y Sergio.

Agradecimientos

Me encantaría mostrar en estas primeras líneas mi incondicional agradecimiento a todas esas personas que, gracias a su apoyo, han hecho posible que llegue a estar donde hoy día estoy. En concreto, y en un nivel más personal, me gustaría, en primer lugar, dar las gracias a mis padres, los cuales han sido, son y serán clave en mi vida a todos los niveles. Todo esfuerzo tiene su recompensa... También me es necesario acordarme de mis hermanos, los cuales saben siempre como alegrarme y ayudarme. Igualmente de amigos que, aunque alejados de este proyecto, siempre mostraban su interés en conocer su estado.

Por último, agradecer a los tutores de este trabajo, tanto a D. Joaquín Abellán Mulero como a D. Carlos Javier Mantas Ruiz por apoyarme y asesorarme para poder conseguir realizar el mismo. Además, por su inestimable colaboración, de la que he disfrutado todos estos meses, y que me ha permitido conocer una aplicación nueva e interesante para mí dentro del mundo del Aprendizaje Automático.

Preface

Actual computers make our life easier in sense to face the hard challenges which several years before were unthinkable. There are many applications that are based on these tools. One of them is called classification, the top point of this document. To make a great introduction to this kind of terms, let's suppose it has a dataset, with several rows and columns. The last one of the latter is the main attribute of the avaiable data, and it is referred as the *class variable*.

The most important aim in classification is to make a great learning process toward forecasting after it. This task consists in designing a based on mathematics model which lets to classify the current data. When it is done, a new unlabeled (no class included) dataset given is classified by the previous rules made. A great learning process should give a great classification results as well.

One of the most popular algorithms in classification problems are named as Decision Trees. Anyone of them is based on *divide and conquer* philosophy, which consists in splitting the main problem into some of fewer dimensions. This process continues until a global solution is obtained, i.e, when each sub-problem is solved. Therefore, Decision Trees try to find the way to split the whole universe (it means the dataset without class variable) by recursively making nodes to add them to the tree. The final branching of the tree refers to a class attribute decision, i.e, what state of it the algorithm finally has taken.

Normally, a Decision Tree takes the decision of splitting by nodes due to a measure called information or uncertainty. This criteria helps to create the entire tree by guaranteeing the choice of the best options. The most popular is the Shannon's entropy that is used in classical Decision Tree Theory. Afterwards, new theorical issues have arisen and researchers have had to keep their efforts in developing new approaches which be able to make a great classifier model. New splitting mode, called Imprecise Info-Gain (IIG), as well as another theoretical point of view, have been proposed by the authors Joaquín Abellán and Carlos Javier Mantas, whose work have originated the called Credal Decision Trees (CDT). This Master Thesis sums up all of the concepts previously commented and let to the reader know about them.

Furthermore, it is going to be explicited a basic introduction to the terms like *noise classification* and *ensemble learning*. Both of them will be linked with CDT to make a new classifi-

cation context recently not used. It is pretended to find a great algorithmic combination which be able to battle against the mentioned problems such as high class noise and uncertainty around a dataset. Therefore, the fundamental target of elaborating this thesis is to compare classification performance between ensembles based on both CDT and classical Decision Trees toward finding a new approach to drive noisy datasets.

Keywords: Decision Tree, Ensemble, Credal Decision Tree, Credal Sets, Noise, Prunning.

Índice general

Justificación

Son muchas las aplicaciones reales en la actualidad para las cuales, herramientas como la Minería de Datos, las predicciones y la estadística juegan un papel fundamental. Hoy día es posible almacenar enormes cantidades de información a nivel computacional. Dicho contenido, recogido a menudo en bases de datos, ayuda a descubrir patrones, tendencias y anomalías que intrínsecamente contienen, aunque en un primer momento no se pueda visualizar o argumentar. Es gracias a los modelos, algoritmos y métodos analíticos creados hasta la fecha, que dichos resultados se pueden obtener, pudiendo conseguir información a través de estos conjuntos de datos, y dicha información convertirla a su vez en conocimiento que permita a los expertos e interesados tomar decisiones con un importante rigor.

Uno de los enfoques más destacados últimamente dentro del campo de la inteligencia artificial es el de regresión y clasificación. El objetivo principal de este tipo de problemas es el de construir una base efectiva que permita pronosticar futuras observaciones con el menor error posible. Más concretamente, en situaciones donde se requiere emplear técnicas de clasificación, el experto puede hacer uso de un gran abanico de posibles algoritmos para resolver el problema en cuestión. Una de estas propuestas corresponde a los denominados Árboles de Decisión, que ocupará el grueso de este trabajo. La mayoría de los algoritmos creados para la construcción de dichos árboles han tenido en cuenta una perspectiva exacta en cuanto a la distribución de probabilidad asociada a la variable de interés bajo estudio que todos los conjuntos de datos que se precian a este tipo de análisis contienen. En este sentido, hay que destacar la teoría que se desarrolla a partir de los teoremas de Bayes y de la Probabilidad Total, gracias a los cuales es posible actualizar dicha distribución de probabilidad cuando se consiguen nuevos aportes de información. Este último punto de vista en la actualidad sigue siendo usado en muchos casos ya que es fundamental, pero en otras situaciones puede resultar insuficiente o ineficaz.

A modo de ejemplo se puede imaginar un conjunto de tamaño $n_1 = 10$ en el nueve de tales instancias pertenecen a la clase $a \in C$ y la restante a la $b \in C$, siendo C la variable clase y a, b sus posibles estados. Además, se puede considerar otro dataset de $n_2 = 1000000$, donde 900000 filas tienen como valor de clase a y las restantes 100000 el b. En ambos casos, $P[C = a] = 0.9$ y $P[C = b] = 0.1$, es decir, la probabilidad de que la variable clase sea de un tipo u otro es el mismo en los dos conjuntos, sin importar el tamaño muestral.

Ejemplos como el de esta última situación pueden poner de manifiesto limitaciones en el uso de algoritmos de clasificación clásicos para este tipo de conjuntos de datos. Afortunadamente, en los últimos años se ha venido desarrollando una gran cantidad de conocimiento en torno a una nueva perspectiva, el de la teoría de las probabilidades imprecisas. Dentro de esta se pueden encontrar multitud de aplicaciones según el contexto para el cual fueron diseñadas (lo cual constituye su principal problema), situándose entre las más conocidas la *Teoría de la Evidencia*, *Teoría de la Posibilidad*, *capacidades de orden 2*, *probabilidades superiores e inferiores* o *conjuntos convexos de distribuciones de probabilidad*. De todas ellas, la que se constituye con un mayor grado de generalidad es esta última, también denominada en la literatura *Conjuntos Credales*.

Llegado el momento, Dempster apareció como el precursor de estos métodos cuestionando el sentido del uso de la teoría de la probabilidad clásica en determinadas situaciones y comenzó a considerar conjuntos convexos de distribuciones de probabilidad (Dempster [24]). Pero quizá el autor más importante en la línea de la justificación del uso de conjuntos convexos es Walley (Walley [58]). Otros trabajos, en consonancia con el anterior, tales como los aportados por Cano, Moral and Verdegay [15] y Moral and de Campos [46] surgen en el mismo año. Posteriormente, Walley [60] presenta el modelo de Dirichlet Impreciso, que ha sido muy utilizado en la literatura hasta nuestros días.

A pesar de tener una gran cantidad de conocimiento sin homogeneizar en un único todo, en los últimos años se ha realizado un esfuerzo para conseguir este propósito. Por otra parte, el hecho de usar la teoría basada en probabilidades imprecisas desde otro enfoque distinto al tradicional para representar la información aportada por el experimento probabilístico en sí conlleva a construir y estudiar medidas que cuantifiquen el grado de incertidumbre o falta de información. Originariamente, el estudio de la incertidumbre surge en los sistemas de telecomunicación, donde Hartley [33] fue el primero en establecer una medida de la incertidumbre, basándose en la teoría de conjuntos, adaptada posteriormente a otros modelos. Más tarde, y constituyéndose como uno de los principales pioneros, Shannon [55] sentó las bases dentro de las probabilidades precisas y la teoría de la probabilidad clásica un nuevo tipo de incertidumbre, conocida como la medida de entropía de Shannon. Posteriormente surgen distintos trabajos donde se pretende conocer cómo medir la incertidumbre, así como el establecer una medida global. Así, en base al esfuerzo de otros muchos autores que también han conseguido importantes avances dentro de este campo, destacan Maeda and Ichihashi [40]. Por otro lado, a raíz del problema anteriormente comentado sobre la diversidad de puntos de vista válidos dentro de esta teoría, para conceptos tan importantes como los de independencia (ver Couso, Moral and Walley [16]) y condicionamiento (ver Dubois and Prade [27]) surgen los congresos *International Symposium on Imprecise Probabilities and Their Applications* (ISIPTA) (1999, 2001, 2003, 2005, 2007, 2009).

Volviendo al punto de interés fundamental, tanto el estudio de la incertidumbre, llevado a cabo en una primera instancia en la teoría de los Conjuntos Credales, como las medidas de información sobre estos se extrapolan a un nuevo terreno donde tiene sentido su aplicación. En concreto se habla de los algoritmos de clasificación basados en Árboles de Decisión, que se analizarán en profundidad en este trabajo. En el ejemplo expuesto anteriormente, los distintos tamaños de muestra inducen, en un primer razonamiento intuitivo, a diferentes representaciones de los modelos de clasificación, o Árboles de Decisión. Parece lógico que para un tamaño de muestra $n_1 = 10$ se creará un árbol con patrones más simples y de menor tamaño que con otro tamaño de $n_2 = 10000000$. La idea subyacente, para entender el motivo de este trabajo, radica en crear combinaciones de algoritmos de Árboles de Decisión basados en probabilidades imprecisas para poder determinar si existe alguna aportación positiva cuando se clasifican bases de datos que poseen ruido, de cara a minimizar lo máximo posible las tasas de error a la hora de etiquetar nuevas instancias. La combinación de tales algoritmos y otros ensembles clásicos, como por ejemplo Bagging, no han sido hasta ahora evaluados y, tras la realización de este trabajo, se podrá concluir con la afirmación de si tiene o no utilidad el uso de estas técnicas en la práctica.

Objetivos

La meta principal que se persigue al realizar este trabajo es la de conseguir construir ensembles o combinadores de varios clasificadores, compuestos en este caso por Árboles de Decisión basados en probabilidades imprecisas para comprobar su rendimiento sobre bases de datos que contienen ruido. Así mismo, se estructura una línea de trabajo experimental basada en la puesta en marcha de distintos clasificadores los cuales toman conjuntos de datos a los que se les impone un porcentaje arbitrario de instancias mal etiquetadas en la variable clase de cada uno de ellos. Además, fundamentando la anterior puesta en práctica, se pretende desarrollar una revisión del estado del arte acerca de la teoría de probabilidades imprecisas y Conjuntos Credales que permita crear un sólida base de conocimiento para entender tales experimentos. Por otra parte, y no siendo estas últimas las únicas, han sido otras las motivaciones que han llevado a este trabajo a realizarse, tales como:

- Comparar la precisión de clasificación de los algoritmos combinados anteriormente comentados con otros procedimientos que han demostrado funcionar con éxito en un contexto de ruido.
- Adquirir habilidad y competencias en el manejo de software especializado para tareas específicas que se requieren para la elaboración de este tipo de estudios.
- Comprobar si existen o no diferencias estadísticamente significativas entre el uso de nuevas combinaciones de clasificadores y otras propuestas basadas en algoritmos de Árboles de Decisión clásicos (tales como Random Forests o C4.5) empleados en numerosos trabajos aportados con anterioridad sobre conjuntos de datos reales.
- Aprender y dar a conocer varios tipos de análisis de comparación entre métodos de clasificación que se emplean habitualmente en este tipo de estudios.
- Intentar dar pie a futuras lineas de investigación basadas en los posibles resultados positivos obtenidos de este trabajo.

De todos ellos, considero que, aunque algunos en mayor y otros en menor medida, se han conseguido alcanzar todos. Además, los resultados obtenidos en el apartado referido al estudio

experimental permiten dar pie a desarrollar nuevas propuestas que permitan seguir estudiando la viabilidad de este tipo de algoritmos bajo este tipo de situaciones (con ruido).

Este trabajo se estructura como sigue: en el Capítulo 1 se presentan conocimientos básicos sobre Aprendizaje Automático. En el Capítulo 2 se introducen los conceptos necesarios para entender el funcionamiento de los Árboles de Decisión. En el Capítulo 3 se desarrolla la teoría de probabilidades, medidas de incertidumbre y Árboles de Decisión en un contexto de imprecisión. En el Capítulo 4 se pretende mostrar desde un punto de vista teórico los ensembles (o combinaciones de algoritmos de clasificación) más conocidos. En el Capítulo 5 se muestran las aplicaciones que los árboles comentados en el tercer capítulo pueden tener, así como una breve documentación sobre los análisis estadísticos a usar en el estudio experimental. Por último, en el Capítulo 6 se pone en práctica todos los conocimientos adquiridos en las anteriores secciones.

Capítulo 1

Introducción al Aprendizaje Automático

La era de la computación se ha consolidado en las últimas décadas a base de avances gigantescos y logros que han permitido a la humanidad poder desempeñar tareas más complejas de un modo más sencillo y automatizado que antaño. El tratar las fuentes que se extraen, cada vez más heterogéneas y de mayor tamaño, con la soltura que los sistemas informáticos permiten hoy día, de cara a convertir dicho banco de información en competencias o sabiduría para el usuario que las maneja, es digno de comentar.

Si se atiende a interpretar dichas fuentes de información como bases de datos, se puede ceñir más el área de aplicación de estos sistemas a un campo de conocimiento específico, como es la estadística. Sin temor a caer en la falacia se puede demostrar que sin la existencia de esta ciencia probablemente no se habrían alcanzado gran parte de los propósitos propuestos por muchos de los grandes autores que han partipado en el avance científico a lo largo de los últimos siglos.

En un inicio, dentro de la Edad Contemporánea, comienzan a surgir conceptos tan elementales para un conocedor de esta disciplina como el de *correlación*, gracias a Francis Galton (1888) o el estadístico *t-student*, por William Sealey Gosset (1908). Pero es sin duda, a partir de 1922, con la llegada de Ronald Arnold Fisher, considerado por muchos como el *padre de esta ciencia*, cuando se produce una gran revolución y avance dentro de la teoría que concierne a lo que hoy día se entiende como estadística, desarrollando y planteando numerosos tipos de experimentos y modelos.

A día de hoy las herramientas estadísticas desarrolladas por los autores ya comentados, entre otros muchos, aportan métodos efectivos para analizar, describir y explicar los distintos resultados que se obtienen a partir de un conjunto de datos. Sin embargo, con el paso de los años, la estadística se ha visto encubierta de otras ramas de conocimiento similares a su propósito y que han provocado que esta no sea la única herramienta indispensable para el análisis formal de un experto que se enfrenta a una base de datos. Dichas ramas comprenden lo que se conoce hoy día como *Minería de Datos* y *Machine Learning*, principalmente. Esta

conjunción de conocimiento ha conllevado a concebir un nuevo perfil profesional dentro del mundo laboral y de investigación, tal y como es el llamado *Científico de Datos*, que ha de conocer y dominar todos estos conceptos. Sin duda alguna, esta profesión está llamada a ser una de las más importantes en cuanto a la aportación que tanto a la sociedad como a nivel privado dentro del mundo empresarial puede dar de sí, y es que se necesitan datos para proporcionar información, en forma de resultados, interesante que posea validez, fiabilidad, consistencia, seriedad y disciplina para cada una de las aplicaciones en las cuales se desempeñen cada una de las técnicas existentes.

1.1. Machine Learning

Durante el transcurso de la historia, la humanidad ha anhelado aquello que no tenía y que de alguna u otra forma, sabía que más tarde o más temprano podría intentar alcanzar. La llegada de los ordenadores, así como el ya comentado avance notorio de la tecnología y de la forma de procesar la información han llevado a la comunidad científica a plantear nuevos retos en los últimos años.

La teoría desarrollada y dedicada a Machine Learning (ML) entra dentro del campo de la inteligencia artificial. El hecho de que las máquinas virtuales aprendan a partir de un conjunto de datos un modelo para poder extrapolar este saber a otros conjuntos de datos futuros y a priori no conocidos y analizados es lo que podría entenderse como Aprendizaje Automático.

En un primer momento, y ante el contexto que acontecía en la década de 1970 dentro del mundo de la inteligencia artificial, un grupo de expertos planteó de manera simple y audaz el poder transmitirle a un ordenador una serie de instrucciones de manera indirecta para afrontar un determinado problema, mediante el uso de ejemplos que permitieran a las máquinas reconocer, aprender y resolver el mismo. La principal dificultad encontrada consistió en cómo proporcionar al hardware dichas sentencias. Fue entonces cuando se desarrolló todo un abanico de algoritmos que fundamentarían las bases de lo que hoy día se conoce como Aprendizaje Automático (Kubat [38]).

En ML, las máquinas aplican técnicas de análisis de datos, aprendizaje estadístico, creación de modelos, etc, para identificar automáticamente patrones o tendencias dentro de los datos. Así mismo, éstas técnicas se utilizan para realizar predicciones con gran grado de precisión, siempre y cuando los pasos previos dados sean lo suficientemente correctos. Una de las tareas fundamentales que se desempeña dentro del Aprendizaje Automático es el de clasificación o regresión. Este trabajo contiene técnicas referidas a la primera de ellas.

Normalmente, dado un conjunto de datos, éste posee las siguientes características: por filas se encuentran las llamadas *instancias* y por columnas las *variables* o *atributos*. Dentro de estos últimos, hay que distinguir entre los que aportan información al conjunto de datos

en sí mismo y por otra parte, una única variable, llamada *clase*, la cual, contiene para cada instancia el estado de la misma.

1.1.1. Minería de datos

Además del auge del Aprendizaje Automático, también se ha hecho hincapié en desarrollar toda una teoría que engloba parte de las técnicas y algoritmos propios de Machine Learning así como otros conceptos e ideas. Intentando buscar alguna diferencia entre ambos conceptos, podría decirse que mientras que el Aprendizaje Automático puede entenderse desde el punto de vista de la máquina, la Minería de Datos es percibida como un proceso o estudio llevado a cabo por el usuario con los datos, herramientas y objetivos que posee y maneja. En otras palabras, el primer concepto engloba toda la batería algorítmica, teórica y de lenguaje abstracto dentro del campo de la inteligencia artificial y la segunda se centra en una perspectiva más clásica del estudio y análisis de los conjuntos de datos como principal herramienta, sin dejar de lado el contexto similar de ML en el que se sitúa.

Como se puede apreciar, ambos conceptos son distintos, entendidos de distinta manera y enfocados a un tipo de tarea en especial. Para finalizar esta comparación sería interesante aportar una definición de lo que puede entenderse como Minería de Datos. Según Aggarwal [13] en su libro, la describe como *el estudio basado en la recolección, limpieza, procesamiento, análisis y la obtención de información útil proveniente de los datos*.

1.2. Clasificación

La principal distinción que se puede realizar sobre las tareas de clasificación y regresión es en cuanto a la naturaleza de la variable clase. Si esta última corresponde a un atributo de tipo discreto, la tarea a realizar es de clasificación, mientras que si es continuo se emplea regresión.

Dado el tipo de conjuntos de datos con los que se va a trabajar en este trabajo en la fase experimental posterior, conviene centrarse en los problemas de clasificación y sus variantes. Además de las características explicadas en la sección 1.1 que tienen los conjuntos de datos de este tipo de problemas, es necesario comentar el procedimiento que se suele llevar a cabo en toda tarea de clasificación. Se ha de poseer dos conjuntos de datos, uno llamado de entrenamiento o *training* y otro de prueba o *test*. El primero contiene todos los atributos comentados en la mencionada anterior sección, mientras que el segundo conjunto no está etiquetado, es decir, no contiene entre sus atributos a la variable clase. Por lo tanto, el objetivo fundamental es, para el conjunto de datos de prueba, crear dicha variable de tal manera que cada instancia quede correctamente valorada. Para lograr esto, existen multitud de algoritmos de clasificación que serán los encargados de crear un modelo a partir de los datos de entrenamiento y con el cual predecir el valor de la clase para cada instancia del conjunto de test.

Finalmente, los problemas de clasificación pueden ser divididos en dos subconjuntos a su vez, los supervisados y no supervisados, en función de la información disponible en los conjuntos de datos con los que el experto tiene que lidiar.

1.2.1. Clasificación supervisada

Este tipo de análisis es viable cuando quien se encarga de realizar un análisis tiene ante sus manos un conjunto de datos de entrenamiento que posee variable clase. Como se decribe en Maimon and Rokach [41], a partir de dicho dataset, se puede descubrir posibles relaciones entre los atributos de entrada y dicha variable *target*. Estas relaciones son normalmente configuradas en un modelo, comentado anteriormente, a usar para la futura predicción de nuevas instancias.

Dada la naturaleza de los problemas relacionados con la clasificación supervisada, se han desarrollado diferentes tipos de algoritmos que funcionan bien en general y que, en contextos particulares, sobresalen unos por encima del resto en términos de precisión y minimización del error de clasificación. Algunos de los principales son:

- **Árboles de decisión:** Surgen entre la década de los 50 y 60. Constan de la creación de un modelo cuya representación gráfica queda reflejada en una estructura de árbol, con nodos y aristas que permiten, en base a criterios matemáticos llegar a clasificar instancias nuevas en función del cumplimiento o no de los requisitos que cada nodo, perteneciente a los atributos de entrada del dataset, impone.

- **Algoritmos genéticos:** Aparecen a partir de 1970. Su nombre se debe a la similitud de su funcionamiento y el propio de la ciencia genética dentro del terreno biológico y evolutivo. Corresponde al tipo de algoritmos basados en búsqueda heurística y en general suelen tener altos costes computacionales en términos de tiempo de ejecución por su modo de operar.

- **Redes neuronales:** Sus comienzos se remontan a 1940. Sus tareas se asemejan a las que realiza el cerebro humano, donde la transmisión de información entre neuronas es su principal herramienta, de ahí su nombre. Suelen emplear un gran número de procesadores que operan en paralelo, cada uno con su propia información y parte de los datos almacenados en su memoria local.

- **Support Vector Machine:** El algoritmo inicial se creó en 1963. Se centra en encontrar el hiperplano que maximiza el margen entre las dos clases (bajo un problema de clasificación binario). Los vectores que definen a dicho hiperplano son los denominados vectores soporte.

- **Redes bayesianas:** Corresponden a modelos gráficos basados en la teoría de la probabilidad. La estructura gráfica es similar a la de árboles de decisión, ya que cada nodo hoja representa a una variable del conjunto de datos, pero en este caso las aristas que unen dichos nodos representan dependencias probabilísticas entre las correspondientes variables.

1.2.2. Clasificación no supervisada

En contraposición a la supervisada, en este caso la variable clase no está presente en el conjunto de datos inicial, por lo que es necesario clasificar a las instancias de otro modo, pudiéndose interpretar como una clasificación a ciegas. La idea fundamental es formar grupos heterogéneos que compartan alguna característica y que permitan a nuevos conjuntos de datos ser etiquetados lo mejor posible. Entre los algoritmos y propuestas más conocidas se encuentran:

- **Reglas de asociación:** Son usadas para identificar y representar dependencias entre elementos (items) de una base de datos. Son reglas que siguen la siguiente notación: $X \rightarrow Y$, donde X e Y son un conjunto de items (itemset).

- **Análisis cluster:** Corresponden a algoritmos que proporcionan grupos dentro de una nube de puntos en base a algún criterio. Suelen ser diferenciados según los llamados métodos jerárquicos y particionales.

1.3. Ruido

La información que se maneja dentro de los trabajos referidos al uso de técnicas de Minería de Datos y Machine Learning no suele estar pensada para ser tratada y analizada en una primera toma de contacto. El sistema de medición de multitud de fenómenos que se desean estudiar están sujetos a recogidas erróneas, como comenta Frénay and Verleysen [30]. En este sentido, la mayoría de los conjuntos de datos reales hoy día contienen ruido, definido este último por Hickey [34] como *cualquier elemento que nuble la relación entre los atributos de entrada de una instancia y su clase*.

Entre otras consecuencias, muchos trabajos aportados en los últimos años, como el artículo de Zhu and Wu [65], han demostrado que la presencia de ruido en un conjunto de datos afecta de manera notable al rendimiento de un algoritmo de clasificación. Es, por tanto, necesario implementar técnicas que eliminen este ruido o reduzcan lo máximo posible sus consecuencias. Es importante destacar esto último también ya que los datos considerados más fiables (sin presencia de anomalías) a menudo son difíciles de conseguir, debido a su alto coste económico

y por requerir mucho tiempo hasta su recogida final, lo que explica la común presencia del ruido en la mayoría de los conjuntos de datos reales creados.

En la literatura se diferencian dos tipos de fuentes de error o ruido: uno causado por los atributos de entrada y otro por la variable clase. Centrándose en este último tipo, se ha de pensar en cómo ocurre este problema. Sáez, Galar, Luengo, and Herrera [51] identifican dos fuentes de error que dan lugar a que aparezca el ruido dentro de esta variable:

- **Ejemplos contradictorios:** Existen filas duplicadas en el conjunto de datos que poseen distintas etiquetas para la última variable del mismo (suponiéndose que esta es la referida a la clase).

- **Clasificación errónea:** Las instancias son etiquetadas con otra clase diferente a la que en realidad debería. Por ejemplo, en un problema de clasificación binario, consistiría en asignar *bajo* a la clase de una fila cuya etiqueta debería ser *alto*.

En Zhu and Wu [65] y Sáez, Galar, Luengo, and Herrera [51] se comenta que el ruido incluido en la variable clase es más peligroso en términos de la futura precisión de clasificación que el mismo existente en los atributos de entrada, lo que manifiesta rotundamente la importancia de tratar con este tipo de errores de medición. Este trabajo, en su fase tanto teórica como experimental, se centrará en este tipo de ruido, comentando sus posibles influencias.

La presencia de ruido en la variable clase es un problema ciertamente cercano a la detección de *outliers* y anomalías Barnett and Lewis [17]. De hecho, las instancias que están etiquetadas erróneamente podrían estar sujetas a ser *outliers* si dicha etiqueta tiene una probabilidad pequeña de ocurrencia dentro de su vecindad. De manera similar, tales instancias podrían considerarse anómalas con respecto a la clase que corresponde su etiqueta incorrecta (figura B.9).

Muchas de las técnicas que se han desarrollado para tratar de resolver el problema de la presencia de *outliers* y anomalías pueden ser usados cuando existe ruido en la variable clase. Sin embargo, es fundamental destacar que las instancias etiquetadas erróneamente no son necesariamente *outliers* o anomalías, por lo tanto habrá que estudiar y analizar en gran detalle la naturaleza de los datos con los que se trabaja de cara a poseer un conjunto de calidad que permita hacer pronósticos fiables.

Por otra parte, hay que comentar también las posibles causas que provocan que exista ruido en un conjunto de datos. Entre otras posibles, la más inmediata surge del propio error humano al codificar los datos que mide. Además, la información que se le proporciona al experto puede que sea difusa o lo suficientemente carente de contenido para que se cometan errores. Por otra parte, existen dispositivos que etiquetan automáticamente cada ejemplo que tampoco están exentos de fallar. En otras situaciones el proceso de clasificación se ve influido por un contexto

subjetivo por parte de quien lo lleva a cabo, por ejemplo, en aplicaciones dentro del terreno de la medicina y que tienen siempre presente el hecho de poder caer en el error.

Basándose de nuevo en la gran labor de Frénay and Verleysen [30] dentro de este campo de conocimiento, se muestra en él una taxonomía específica para el ruido en la variable clase, teniendo en cuenta cuatro aspectos fundamentales: el verdadero valor de la clase, el valor observado de la misma, el conjunto de atributos de entrada del dataset y una última variable indicadora que muestra si el valor real y el observado coinciden.

Haciendo hincapié en las posibles consecuencias de afrontar un problema de clasificación donde el atributo referido a la clase tiene esta característica, se puede valorar que además de las ya comentadas, pueden surgir otras y todas negativas para la consecución de un buen modelo de predicción. Entre ellas se pueden mencionar, en primer lugar, el descenso de la tasa de acierto del clasificador en cuestión, al crear un modelo no ajustado a la realidad. También pueden surgir inconvenientes en cuanto al rendimiento del algoritmo en sí y la complejidad del modelo creado. Además, poniendo en escena de nuevo disciplinas como la medicina, la presencia de ruido altera la distribución de las frecuencias observadas correspondientes a los test clínicos que se realizan y que podrían llevar a conclusiones equivocadas.

Ante esta problemática que plantea el ruido dentro de la variable clase, los investigadores han realizado un gran esfuerzo en los últimos años para poder encontrar algoritmos robustos y lo suficientemente efectivos para combatir este hecho. Los expertos usan una función de pérdida y los clasificadores actúan minimizando la pérdida esperada, denominada usualmente como *riesgo*, para futuras muestras. Cada algoritmo posee su propia función de pérdida. Así, por ejemplo, AdaBoost implementa una función de tipo exponencial, mientras que Support Vector Machine hace uso de la llamada *función de pérdida de bisagra* o *hinge loss function*. En otras palabras, no existen algoritmos que homogéneamente sean robustos frente al ruido.

Por otro lado, se sabe que en presencia de ruido dentro del atributo clase, ensembles (algoritmos de clasificación constituidos de varios algoritmos a su vez que en combinación producen una predicción única) como Bagging consiguen mejores resultados que otros como Adaboost (Dietterich [25]).

En cuanto al tipo de algoritmos que principalmente ocupa a este trabajo, los Árboles de Decisión, la literatura muestra claramente que están fuertemente afectados por el ruido de clase (Abellán and Masegosa [12]). De hecho, esta inestabilidad hace que resulte adecuado combinarlos mediante ensembles (Abellán and Masegosa [11]). En esta misma última cita, se comparan diferentes criterios de división de nodos para ensembles basados en Árboles de Decisión, bajo la presencia de ruido en la variable clase.

Capítulo 2

Árboles de Decisión

Como ya se ha podido comprobar en el anterior capítulo, existen varias alternativas para las cuales un problema de clasificación puede ser representado y modelizado, tales como redes bayesianas, redes neuronales, Árboles de Decisión, etc. En este caso, se estudiarán estos últimos en profundidad, conociendo cómo se estructuran y funcionan internamente.

En el intento de familiarizarse con estos nuevos conceptos, un Árbol de Decisión puede definirse como *una herramienta visual que permite clasificar nuevas instancias a partir de decisiones tomadas en base a los atributos de un conjunto de datos de entrenamiento*. Es, por tanto, un modelo basado en un gráfico dirigido, acíclico y con estructura de árbol, de ahí su nombre.

2.1. Preámbulo

Cada nodo del árbol puede tener o no extremos salientes que lleven a otros nodos. En caso negativo, se dice que es un nodo de decisión o *nodo hoja* y de lo contrario se denomina nodo prueba o nodo de un atributo cualquiera.

Los modelos de clasificación basados en Árboles de Decisión poseen una gran ventaja con respecto a otros algoritmos, ya que están dotados de una estructura fácil de interpretar y constituyen eficientes clasificadores. Se ha de mencionar como punto de partida el algoritmo ID3 de Quinlan [49] en la construcción de estos árboles, basados en la teoría de la información e incertidumbre y en el cálculo de probabilidades clásico. Posteriormente, en 1993, dicho algoritmo fue mejorado y actualizado al conocido C4.5, el cual todavía hoy perdura como modelo de referencia para construir Árboles de Decisión basado en el enfoque estadístico tradicional (Salzberg [50]).

Prácticamente en el mismo periodo que Quinlan desarrollaba ID3, más concretamente en 1986, Leo Breiman, por su parte, dedicó su tiempo a la implementación y estudio de otro algoritmo basado en árboles de decisión, llamado CART (*Classification and Regression*

Trees) (Breiman, Friedman, Olshen, and Stone [20]) y que sirvió de base para formalizar posteriormente otros planteamientos (Dietterich [25]). Este algoritmo no paramétrico crea árboles binarios en base a variables tanto continuas como discretas.

Los Árboles de Decisión se posicionan como una de las alternativas con mayor uso dentro de la inteligencia computacional, incluso cuando otros algoritmos proporcionan modelos más precisos, debido a su simplicidad. Una de las razones fundamentales de su atractivo radica en su comprensibilidad. Los Árboles de Decisión pueden ser expresados de manera sencilla mediante un conjunto de reglas de carácter lógico que permiten describir las decisiones tomadas (Grabczewski [32]). Viendo una definición más formal del término, descrita en Podgorelec and Zorman [48], un Árbol de Decisión puede ser visto como una función, tal que:

$$dt : dom(A_1) \cdot ... \cdot dom(A_n) \mapsto dom(C) \tag{2.1}$$

donde A_1, A_2, ..., A_n y C constituyen las variables aleatorias (incluida la clase) del conjunto de datos en cuestión y $dom(A_i)$ es el dominio para el atributo i, $\forall i = 1, ..., n$. En términos estadísticos, se puede considerar al conjunto de datos D como una extracción aleatoria surgida de una población más numerosa P. Por lo tanto, si se denota $P\left(A', C'\right)$ como la distribución de probabilidad sobre $dom(A_1) \cdot ... \cdot dom(A_n)$ y

$$t = (t.A_1, t.A_2, ..., t.A_n, t.C)$$

como una realización aleatoria proveniente de P, entonces t tiene probabilidad $P\left(A', C'\right)$ de que $(t.A_1, t.A_2, ..., t.A_n) \in A'$ y que $t.C \in C'$. En un término menos complejo, podría interpretarse dicha probabilidad como aquella para la cual la cifra correspondiente al atributo i e instancia j del conjunto de datos D pertenece a la distribución que tiene esta misma variable.

La interpretabilidad de este tipo de clasificadores, como ya se ha ido comentando, constituye una de sus principales ventajas. A modo de ejemplo, si se atiende a la figura B.1, se puede argumentar que, si el ancho del pétalo es menor o igual a 0.6 centímetros, entonces el tipo de flor clasificada es Setosa. Si por el contrario, dicho ancho es mayor que esa cifra y menor o igual que 1.7 centímetros, así como la longitud del mismo pétalo es menor o igual a 4.9 centímetros, entonces la especie de flor candidata es Versicolor, y de esta forma se podría interpretar el resto de los sub-árboles generados por este modelo sobre el conjunto de datos *Iris*.

Está demostrado que la combinación o ensemble de clasificadores puede mejorar la precisión de estos por separado (Dietterich [25]). Este trabajo se va a centrar en el uso ensembles que consisten en la combinación de Árboles de Decisión y algoritmos clásicos de clasificación bajo un enfoque probabilístico distinto al de la perspectiva clásica y sobre bases de datos que poseen ruido, como se podrá comprobar en futuros capítulos.

2.2. Funcionamiento de un Árbol de Decisión

El principal objetivo de un Árbol de Decisión es explorar el espacio de posibles soluciones exhaustivamente y elegir de entre ellas las que, bajo algún criterio, son las más adecuadas. El árbol comienza a construirse sobre un espacio vacío, al que sucesivamente, y según una serie de estos criterios que se ilustrarán posteriormente, le son añadidos nodos de distinta naturaleza. Finalmente, mediante algún otro principio, se paraliza esta construcción, dando lugar al árbol resultante.

Como ya se ha comentado en anteriores secciones, en problemas de clasificación la variable clase es siempre de tipo discreto y se representa por medio de un conjunto de posibles valores o estados. Por otra parte, si para construir el árbol, dentro de un hipotético ejercicio de clasificación, se tiene que hacer uso de un atributo continuo del conjunto de datos, primero ha de discretizarse, es decir, transformar dicho atributo, a través de intervalos lo más precisos posibles, para convertirlo en otro de tipo discreto (Meyers and Kokol [45]).

Cada nodo hoja es etiquetado con una de las posibles clases c de la variable C. Los nodos de prueba son etiquetados a su vez con un atributo $A_i \in (A_1, ..., A_n)$ denominados *atributos para la división* (APD). Cada APD tiene una función f_i asociada. Dicha función determina los extremos salientes o aristas desde estos nodos, basándose en el valor del atributo A_i en una instancia O del conjunto de datos D. Si dicho valor está contenido en Y_i, entonces se elige la correspondiente arista desde el nodo de prueba, siendo Y_i un subconjunto de la variable en cuestión ($Y_i \subset A_i$). Es decir, dentro del propio proceso de clasificación de instancias, cada una de ellas comienza siendo evaluada en el nodo raíz del árbol. Si este nodo es de prueba, se determina el resultado para dicha instancia y el algoritmo continua haciendo uso del sub-árbol correspondiente. Cuando se encuentra con un nodo hoja, su etiqueta proporciona el valor de la clase predicha para cada ejemplo (ver figura B.2).

El principal problema de la construcción de los Árboles de Decisión es que, para un conjunto de datos $D = (t_1, ..., t_d)$ donde t_i son muestras aleatorias independientes provenientes de una distribución de probabilidad desconocida P, se tiene que encontrar el árbol de decisión T el cual minimice la tasa de clasificación errónea, denotada como $R_T(P)$. En otras palabras, si se pretende crear un clasificador eficaz, se tiene que garantizar que el árbol construido sea capaz de clasificar con el mayor éxito posible tanto las instancias del conjunto de entrenamiento como las del conjunto de prueba (test).

El método para la construcción del Árbol de Decisión descrito en Salzberg [50], es el siguiente. Si se tienen j posibilidades para la variable clase, denotadas como $(c_1, c_2, ..., c_j)$ y un conjunto de entrenamiento D, entonces:

- Si D contiene una o más instancias que pertenecen a una única clase c_i, entonces el Árbol de Decisión es constituido por un nodo hoja que identifica a dicha clase.

- Si D no contiene instancias, el Árbol de Decisión corresponde a un nodo hoja determinado por información que no proviene de dicho conjunto de datos.

- Si D posee instancias que pertenecen a varias clases, entonces se crea y elige un nodo prueba, basado en un único atributo que contiene uno o más resultados mutuamente excluyentes $(o_1, o_2, ..., o_n)$. Además, D es particionado en subconjuntos $D_1, D_2, ..., D_n$, donde D_i contiene todas las instancias de D que tienen como tipo de clase las reunidas en o_i, del nodo de prueba elegido. Este mismo proceso se repite recursivamente para cada subconjunto de instancias del conjunto de entrenamiento.

2.2.1. Criterios de partición

Bien es sabido, llegado a este lugar, que el Árbol de Decisión comienza a construirse desde la base o raíz hasta los distintos nodos hoja que contienen las etiquetas para las cuales clasificar futuras instancias, pasando por otros nodos que contienen cada uno de los atributos escogidos en cada etapa. Sin embargo, todavía no se ha comentado este último proceso, mediante el cual, el algoritmo de construcción de este tipo de árboles elige uno en concreto del conjunto de datos para comenzar a dividir y crear subconjuntos de datos en función de los valores escogidos. Para esta tarea, los investigadores se apoyaron en toda una teoría matemática basada en la medida de la información capaz de adaptarse a la naturaleza de los algoritmos de Árboles de Decisión. En este apartado se comentan las más conocidas, como son la medida de entropía de Shannon y el índice de Gini.

2.2.1.1. Entropía y Ganancia de Información

Algoritmos como ID3 o C4.5, entre otros, usan en sus procedimientos cálculos de índole estadística. Además y como ya se ha comentado a lo largo de este trabajo, al hablar de problemas de clasificación, se sabe que la variable respuesta Y asociada a la clase es de tipo categórico, y por lo tanto se puede hacer uso de la teoría de la información para medir cuanto se *conoce* a dicha variable, sabiendo el valor de otro atributo X del dataset, también discreto (Shalizi [54]).

Estos cáculos, corresponden al principio de la *Ganancia de Información* de un atributo del conjunto de datos para construir el árbol. De esta forma, la variable que aporta más información sobre la decisión en el conjunto de entrenamiento es elegida primero, para, posteriormente escoger la siguiente que cumple el mismo requisito para el conjunto de atributos restante, etc (Meyers and Kokol [45]).

Los conceptos de incertibumbre e información están estrechamente ligados e interconectados. Podría interpretarse la primera como una manifestación de alguna deficiencia de la

segunda, mientras que la segunda podría definirse como la capacidad de reducir la incertidumbre (Klir [36]). Si se aplican estos conceptos al campo de los Árboles de Decisión, la entropía, entendida como medida de información e incertidumbre, mide la falta de fiabilidad de un elemento (atributo) como fuente de información. Cuanta mayor cantidad de información se posee, menor es el coeficiente de entropía. Por lo tanto, la Ganancia de Información se entiende como una reducción de la entropía.

Para entender el funcionamiento de los algoritmos anteriormente comentados, primero es necesario conocer el *modo de dividir* el conjunto de datos original, según por el cual, y argumentado por el cálculo de la entropía de cada atributo, se decide qué nodo es más aconsejable de representar en el árbol. Existen dos criterios de división implementados: uno, comentado anteriormente, basado en la *Ganancia de Información* y otro en la *Razón o Tasa de Ganancia de Información*. Para llegar a definir al primero de ellos se ha de desarrollar matemáticamente algunas expresiones previas de la siguiente forma: dado un conjunto de datos D y n_i el número de instancias del conjunto de entrenamiento con clase c_i, se denota al *mensaje* como la selección aleatoria de una instancia perteneciente a dicha clase. Este *mensaje* tiene probabilidad $p_i = n_i/n$, donde n es el número total de instancias de D. Por tanto, la información transmitida por el *mensaje* (en bits) viene dada por:

$$I = -log_2 p_i = -log_2 \frac{n_i}{n} \tag{2.2}$$

Por otra parte, y dada la misma muestra de datos D y un atributo discreto A con rango de posibles valores en T, sea p_a la proporción de observaciones de A con valor a perteneciente a T. La entropía del atributo A es, por tanto, definida como (Grabczewski [32]):

$$H_A = -\sum_{a \in T} p_a log_2 p_a \tag{2.3}$$

Es necesario hablar tanto de la variable clase como de otro atributo del conjunto de datos cualquiera, ya que en el terreno del aprendizaje basado en clasificación supervisada se hacen uso de dos atributos discretos, que son estos mismos, de tal manera que se mide la relación entre ambas. Sea p_{ac} la proporción de observaciones de clase c de la variable clase C con valor a perteneciente a T del atributo A. Sea $p_{a.}$ y $p_{.c}$ las proporciones de observaciones con valor a perteneciente a T al margen de cualquier valor de c y las observaciones de clase c pertenecientes a C sin importar el valor de a, respectivamente. Entonces, la entropía de las particiones definidas por A y C son, respectivamente:

$$H_A = -\sum_{a \in T} p_{a.} log_2 p_{a.} \tag{2.4}$$

$$H_C = -\sum_{c \in C} p_{.c} log_2 p_{.c} \tag{2.5}$$

Y, finalmente, la entropía conjunta de A y C y la condicional de C dada A vienen definidas como:

$$H_{A,C} = -\sum_{a \in T} \sum_{c \in C} p_{ac} log_2 p_{ac} \tag{2.6}$$

$$H_{C|A} = H_{A,C} - H_A \tag{2.7}$$

Esta última entropía ($H_{C|A}$) mide la reducción de la entropía que es de esperar (entropía esperada) cuando se selecciona el atributo A como candidato para la división y construcción del árbol. Por lo tanto, la ganancia de información para el atributo A viene definida por:

$$I_{gain}(A) = H_C - H_{C|A} \tag{2.8}$$

En cada etapa de la construcción del árbol se escoge aquella variable del conjunto de datos cuyo I_{gain}, de entre todos los posibles, posee el mayor valor. Ahora bien, este criterio no está libre de limitaciones y contiene un problema con respecto a la sesgadez para pruebas con múltiples resultados. Para evitar esto se desarrolló el criterio de la Tasa de Ganancia de Información, definida en Podgorelec and Zorman [48] como:

$$I_{gain\ ratio}(A) = \frac{I_{gain}(A)}{H_A} \tag{2.9}$$

Este nuevo criterio selecciona una prueba para maximizar la anterior ecuación (2.9) sujeto a la restricción de que la ganancia de información sea lo suficientemente grande.

Existen otras aplicaciones de estos algoritmos para explorar el espacio de variables e incluirlas al árbol, pasando de la evaluación univariante (variable a variable) a otra *oblicua* en la que se tiene en cuenta combinaciones de atributos simultáneamente, pero este último punto no se va a desarrollar ya que no queda dentro de los objetivos de este trabajo.

2.2.1.2. Índice de Gini

Otros de los algoritmos basados en árboles para realizar la clasificación de nuevas instancias, como el caso de CART, usan otra medida para conocer el modo de dividir un nodo correspondiente a un atributo determinado en una fase concreta. El Índice de Gini es otra de las reglas más usadas para dicha división dentro de la clasificación con árboles.

Según se expone en Timofeev [57], este método hace uso de la siguiente función de impureza:

$$i(t) = \sum_{k \neq l} p(k|t)p(l|t) \tag{2.10}$$

donde $k, l = 1, ..., K$ corresponden al índice de la variable clase y $p(k|t)$ es la probabilidad condicional de que la clase k dado que se está evaluando el nodo t. Aplicando esta última expresión se maximiza el siguiente problema:

$$\underset{x_j \leq x_j^R,\ j=1,...,M}{arg\,max} [i(t_p) - P_l i(t_l) - P_r i(t_r)] \tag{2.11}$$

Siendo t_p el nodo padre y t_l y t_r los nodos hijos a izquierda y derecha del nodo padre, respectivamente. Además, x_j corresponde a la notación usada para la variable j-ésima del conjunto de datos, mientras que x_j^R constituye el mejor valor de la variable x_j para realizar la división del nodo padre. Se obtiene el siguiente cambio de la medida de impureza $\triangle i(t)$:

$$\triangle\, i(t) = -\sum_{k=1}^{K} p^2(k|t_p) + P_l \sum_{k=1}^{K} p^2(k|t_l) + P_r \sum_{k=1}^{K} p^2(k|t_r) \tag{2.12}$$

Por lo tanto, el algoritmo resolverá el siguiente problema:

$$\underset{x_j \leq x_j^R,\ j=1,...,M}{arg\,max} \left[-\sum_{k=1}^{K} p^2(k|t_p) + P_l \sum_{k=1}^{K} p^2(k|t_l) + P_r \sum_{k=1}^{K} p^2(k|t_r) \right] \tag{2.13}$$

Dicho proceso buscará en el conjunto de entrenamiento la clase mayoritaria y la aislará del resto de datos.

2.2.2. Método de búsqueda

Dentro del funcionamiento de un Árbol de Decisión, existen varias propuestas para lograr una mayor eficiencia de cada uno de los algoritmos de clasificación. Sin embargo, el algoritmo normalmente suele tomar una estrategia de búsqueda Greedy o Hill-Climbing (algoritmos de optimización local), que permiten minimizar la impureza global del árbol en sí mismo.

Una heurística de Hill-Climbing comienza con una solución inicial. En cada paso se generan uno o más vecinos que constituyen nuevas soluciones, de entre las cuales se escoge la mejor en base a algún criterio y se continúa el proceso hasta que no existen mejores soluciones (vecinos). Normalmente conduce a una única solución y es necesario para su correcto funcionamiento conocer cómo evaluar la solución y cómo generar el vecindario.

Por su parte, la búsqueda Greedy usa la información de la que se dispone para producir una solución única. Constituye un problema con múltiples decisiones en el que cada vez que

se toma una, nunca se vuelve a cuestionar. Estos algoritmos funcionan de manera recursiva, considerando sub-problemas más pequeños y resolviéndolos para poder alcanzar la solución global.

2.2.3. Criterio de parada

Durante la fase de construcción del árbol, sobre un nodo cualquiera en una etapa cualquiera, la muestra que contiene (y que hace referencia al conjunto de datos de entrenamiento) y a su vez todo el árbol en conjunto deben determinar si una nueva ramificación debe ser llevada a cabo o no. Según se recoge en Grabczewski [32], existen varios motivos para los cuales finalizar el procedimiento de un algoritmo de estas características:

- **Precisión:** Normalmente el proceso de división finaliza cuando el nodo en cuestión es *suficientemente puro*. Esta pureza puede ser vista desde varias perspectivas, pero la más común dentro del contexto del criterio de parada es la referida a la precisión del nodo. Algunas veces se requiere obtener la máxima precisión posible, mientras que en otros casos se acepta alguna cantidad de error medida por un umbral que determina el número o porcentaje de errores admisible.

- **Tamaño de los nodos:** Algunos autores toman la decisión de no dividir sobre nodos de pequeño tamaño. De nuevo, un umbral acerca de este valor debe ser implantado para precisar qué se entiende por nodo de tamaño pequeño.

- **Número de hojas:** Se finaliza la división cuando el árbol alcanza una complejidad (número de hojas) a priori valorada. Tal estrategia se usa normalmente cuando se necesita ahorrar tiempo de computación o se quiere construir árboles de un tamaño en especial. Para evitar que el crecimiento vaya en la dirección de solo una única rama, el algoritmo de clasificación debería de ser usado junto con un estimador que otorgue una perspectiva de división adecuada y que garantice que los nodos mejor situados se dividan antes que los peores, en términos, por ejemplo, de la Ganancia de Información. Otro modo de prevenir que surjan árboles desequilibrados es usar métodos de búsqueda que hagan crecer el árbol de manera uniforme y balanceada, como por ejemplo la técnica de búsqueda en anchura.

- **Test estadísticos:** Existen algoritmos que, cuando no se puede rechazar ninguna hipótesis en relación a la existencia de independencia entre atributos, se sospecha que puede que sea poco útil dividir y crear un nuevo nodo en el árbol y, por tanto, la parada del algoritmo debería de ser efectuada.

- **Cumplimiento o incumplimiento:** Otros algoritmos mezclan varios criterios de parada y proponen que se finalice el proceso cuando todos ellos cumplan sus respectivos requisitos o bien cuando ninguno de ellos se preste.

2.2.4. Poda

Son varios los motivos por los cuales un Árbol de Decisión no constituye un gran clasificador para un conjunto de datos en cuestión. Uno de ellos corresponde a la propia imperfección atribuida al conjunto de datos de entrenamiento con el que se trabaja. Dichas limitaciones pueden venir, por ejemplo, por la presencia de ruido en dicho dataset (sección 1.3). Por otra parte, ocurre a menudo que un conjunto de datos resulta no ser representativo del problema que se pretende resolver, no conteniendo de esta manera información sobre las relaciones y dependencias a aprender para clasificar correctamente (Grabczewski [32]).

Si un Árbol de Decisión supone una estructura demasiado compleja que permite aprender con mucha exactitud lo que ocurre en un conjunto de entrenamiento, pero al aplicar dicho modelo en un nuevo dataset la precisión decae enormemente, la tarea, por parte del experto, debe de ser revisada, ya que el clasificador no ha aprendido con exactitud las relaciones de interés. En otras palabras, en este contexto está ocurriendo lo que se conoce dentro de Machine Learning como *overfitting* o sobreajuste de un modelo de clasificación.

Otra situación bien distinta puede darse cuando, por distintos motivos, la persona encargada de montar el clasificador decide construir el Árbol de Decisión correspondiente sin que se forme toda la ramificación que un algoritmo, en principio, podría desarrollar para un conjunto de datos dado. Una de estas posibles razones para realizar esta limpieza, sería evitar el ya conocido sobreajuste.

Cuando el comentado *overfitting* afecta a gran parte del árbol, comenzando a presentarse cerca del nodo raíz del mismo, entonces el modelo creado es totalmente impreciso y en general poco se puede hacer para solucionar este problema. Aun así, el podar algunas ramificaciones puede suponer alguna mejora en términos de rendimiento. Afortunadamente, dicho nodo y sus vecinos más cercanos son creados normalmente sobre grandes muestras de datos y, debido a ello, suelen evitar este problema.

Una nueva situación grave y que afecta a la precisión del clasificador ocurre cuando el número de atributos que describen el conjunto de datos es muy grande en relación con el número de instancias del mismo. Entonces, es probable que los datos contengan variables correladas de manera accidental con la variable de salida. En teoría, tales atributos deberían de ser elegidos por el algoritmo de Árbol de Decisión empleado por ser las más informativas y podrían, por ende, desvirtuar el modelo final. En este contexto, las técnicas de poda son menos útiles.

Dentro de la literatura referida a este tema, se destacan dos categorías fundamentales

de técnicas de poda, a saber, la conocida *pre-poda*, por un lado, en el que los algoritmos de construcción de Árboles de Decisión pueden bloquear la continuación del crecimiento del modelo por un nodo determinado y, por otro, la *post-poda*, donde una vez se ha construido el Árbol de Decisión completo, sin objeciones, se eliminan ciertas áreas mediante algún criterio y finalidad. En cuanto a los primeros, también son conocidos como los criterios de parada, ya mencionados en la sección 2.2.3.

Entonces, la poda surge como una posible medida a priori o a posteriori, según como se plantee, para poder solucionar el problema del sobreajuste. Con un modelo de ramificación menos elaborado, quizá se recoja mayor información (es decir, se aprenda mejor) que con el árbol completo. Es decir, debe de existir un equilibrio fundamental entre la precisión del clasificador y la sencillez de la representación que plantea el mismo. Estas dos bazas suponen los dos requerimientos básicos para que se cree un modelo de decisiones de garantías y que ayude a resolver problemas que mantienen relación con la manipulación de datasets reales.

Trabajando directamente con el Árbol de Decisión completo (dentro de la post-poda, por tanto), los métodos existentes o bien actúan directamente sobre éste y la información acerca de la distribución de los datos del conjunto de entrenamiento (métodos directos) o bien se hace uso de métodos de validación, que usan muestras de datos diferentes a la de entrenamiento en un proceso para determinar qué nodos no serán capaces de rendir bien con los datos futuros de test (métodos de validación). Además, a menudo estas técnicas analizan el proceso de aprendizaje con test estadísticos y, en base a sus resultados, eliminan nodos o mantienen las ramificaciones existentes.

Cada algoritmo de poda de un Árbol de Decisión emplea su propia configuración, entendida esta en términos de combinaciones de diversos parámetros. Sin embargo, casi todos (o incluso todos) ellos, poseen elementos en común que pueden usarse siempre que sea posible para, mediante dos técnicas distintas, obtener un árbol podado similar. Dos de estos parámetros modifican la forma de calcular el error, de cara a la validación y selección de un modelo: uno corresponde al error estándar y otro al error de entrenamiento. En la literatura pueden ser ampliamente estudiados ambos de ellos.

2.3. C4.5

Los Árboles de Decisión pueden ser vistos tanto como estructuras disponibles para realizar clasificación como clasificadores jerárquicos. Dentro del terreno que se está tratando en este capítulo, es decir, los clasificadores clásicos, se han comentado ya las principales aportaciones que han supuesto un gran avance en el desarrollo de esta teoría. Una de ellas corresponde al algoritmo C4.5, el cual, como ya se sabe, surge para mejorar al previamente propuesto ID3, ambos de Quinlan. Esta propuesta sigue siendo considerada como un modelo estándar

para realizar clasificación. Además, ha sido aplicado en análisis de datos en abundancia y en diferentes áreas de conocimiento, tales como la astronomía, biología, medicina, etc.

Para conseguir mejorar el rendimiento provisto en términos generales por ID3, C4.5 hace uso como criterio de división la ya comentada Tasa de Ganancia de Información o IGR (2.9) en lugar de la Ganancia de Información o IG usado por el primero (2.8). El primero de ellos penaliza a las variables que contienen gran cantidad de posibles estados con respecto al resto de atributos (podría contemplarse como una operación de normalización de IG). Por lo tanto, el algoritmo C4.5 lleva a cabo un procedimiento más refinado, teniendo además en cuenta la presencia de variables continuas y valores perdidos. También es interesante comentar que posee en sus cálculos una compleja operación de poda de cara a obtener árboles más interpretables. Todo esto ayuda a que C4.5 sea superior a ID3.

En cuanto al funcionamiento, se puede terminar por explicar el criterio de división de este algoritmo, que si bien ya se sabe que corresponde al IGR, hay que especificar que se escoge en cada nodo el atributo con el mayor valor de dicha medida y cuyo IG es el más grande en promedio con respecto al resto de atributos válidos (numéricos o cuyo número de estados no supera el 30 % de la cantidad de instancias pertenecientes a tal subconjunto). Además, el algoritmo finaliza cuando no existe ningún atributo del dataset con IGR positivo o bien cuando los nodos hoja creados corresponden a subconjuntos de datos con un mínimo de instancias, usualmente de 2. Pero haciendo referencia de nuevo a los atributos válidos, el algoritmo también puede finalizar cuando no queda ninguna variable de estas características. Por otra parte, los subconjuntos de datos creados en cada etapa son derivados de atributos binarios, no teniéndose en cuenta, por tanto, las variables continuas.

Como se ha introducido previamente, también este tipo de Árbol de Decisión resuelve el problema de tratar con valores perdidos. Se asume que dichos valores están distribuidos aleatoriamente. Para calcular cada valor, las instancias son divididas en varias partes. El peso inicial de una instancia es igual a la unidad, pero conforme el árbol comienza a formarse el peso se corresponde a la parte proporcional de instancias del subconjunto de datos en cuestión (con suma obviamente de la unidad). Cuando se realizan las predicciones, C4.5 fusiona todas las proporcionadas por el subconjunto del árbol que es consistente con la instancia en cuestión, para marginalizar la variable perdida (existen varias ramificaciones que poseen valores perdidos) haciendo uso de los pesos previamente calculados.

Por último, la propuesta para llevar a cabo la poda a posteriori de este algoritmo es la llamada *Pessimistic Error Pruning*. Este método calcula un límite superior para la tasa de error estimado de un subconjunto del árbol dado, empleando la conocida corrección por continuidad de la distribución Binomial. Cuando dicho límite perteneciente a un nodo dado es mayor que el mismo para los errores producidos por las estimaciones de este nodo suponiendo su comportamiento como el de un nodo hoja, entonces el sub-árbol asociado se poda. En otros

términos, Mantas and Abellán [43] definen este proceso como la estimación del error global de un conjunto de datos. Dicha estimación consiste en el incremento del error del conjunto de entrenamiento, que es:

$$e_{gen}(N) = e_{tr}(N) + e_{inc}(N) \tag{2.14}$$

donde $e_{gen}(N)$ es el error general del nodo N, $e_{tr}(N)$ el asociado al conjunto de entrenamiento y $e_{inc}(N)$ el incremento del anterior. Si el error global del nodo N es mayor que el mismo para su nodo padre, entonces se poda el nodo.

Capítulo 3

Árboles de Decisión basados en probabilidades imprecisas

Una vez revisada una gran cantidad de información acerca de conceptos básicos dentro del campo de la ciencia de datos y más concretamente de la clasificación de instancias de un conjunto de datos, así como varios aspectos relacionados con la familia de clasificadores que constituyen los Árboles de Decisión clásicos, se procede a dar un paso más en la asimilación de estos conocimientos.

Dichos algoritmos tradicionales desafortunadamente no carecen de imperfecciones de cara, en muchas ocasiones, a su falta de adaptatividad al contexto que impone cada conjunto de datos. Por ejemplo, en el proceso de selección de variables para la construcción del Árbol de Decisión, clasificadores como ID3 o C4.5 presentan el problema de que no tienen en cuenta el tamaño del conjunto de entrenamiento en cada nodo. Esta y otras dificultades han provocado el surgimiento de nuevas propuestas e ideas, sustentadas por una modelización matemática y estadística profunda, pero que da como resultado otro tipo de clasificadores basados también en los Árboles de Decisión. En este sentido, se han desarrollado diversas teorías sobre el uso de probabilidades imprecisas, dejando de lado las precisas que fueron usadas por los primeros algoritmos desarrollados.

En este capítulo se pretende explicar qué es un Árbol de Decisión basado en probabilidades imprecisas y medidas de incertidumbre, también denominados *Árboles Credales*. Así mismo, se pretenden analizar los Conjuntos Credales, cómo funcionan, y qué resultados han dado en la práctica debido a su puesta en marcha por parte de entendidos en técnicas de Minería de Datos y Machine Learning.

3.1. Probabilidades imprecisas

La teoría de la probabilidad clásica determina que cada una de las probabilidades referidas a todas las posibilidades de un experimento (sucesos elementales) son números reales exactos. Con estas cifras, se pueden obtener otras probabilidades ligadas a otros tipos de sucesos, en virtud de las propiedades o leyes conocidas de la mencionada teoría y constituyen, a su vez, números reales precisos. Este último requisito es sumamente restrictivo, pues en muchas ocasiones en los problemas que surgen del mundo real los datos pueden pertenecer a más de una distribución de probabilidad (Klir [36]).

Estudios como los propuestos por Wang [61], Walley [59] y Weichselberger [62] establecen una teoría fundamentada en el uso de probabilidades imprecisas. La utilización de este nuevo planteamiento, según comentan Mantas and Abellán [42], genera ciertas ventajas: por un lado, se resuelve el problema de manipular la ignorancia total, mientras que por otro, la indeterminación y la inconsistencia están adecuadamente representadas.

En un ejemplo propuesto por Klir [36] se refleja claramente el concepto de probabilidades imprecisas. Para un problema bidimensional de probabilidad, donde los posibles estados son $X = \{x_1, x_2\}$ e $Y = \{y_1, y_2\}$, se determina que, sabiendo que $P[X = x_1] = 0.8$ y $P[Y = y_1] = 0.6$, la probabilidad conjunta es:

$$P_{11} = P[X = x_1, Y = y_1] \in [0.4, 0.6]$$

Es decir, dicha medida no tiene un valor único, puesto que todos ellos verifican la propiedad básica de la probabilidad, esto es, que el resto de probabilidades de la distribución (P_{12}, P_{21} y P_{22}) son mayores o iguales que cero.

En resumen, cuando se hace uso de modelos matemáticos para representar la información disponible, se obtienen normalmente conjuntos cerrados y convexos de distribuciones de probabilidad. Dichos conjuntos que engloban probabilidades en intervalos barajando distintos posibles valores se les conocen en la literatura como *Conjuntos Credales* y su desarrollo teórico será visto en este trabajo de manera introductoria en la siguiente sección.

3.1.1. Conjuntos convexos de distribuciones de probabilidad

Existen varios modelos matemáticos propuestos en la literatura para representar la información sobre cualquier experimento probabilístico. Por ejemplo, aquel que tiene conocimientos estadísticos contempla la posibilidad de representar los posibles resultados obtenidos, mediante intervalos de probabilidad, ya que resulta natural el hecho de especificar la credibilidad de un experimento por medio de este método. Sin embargo, dicho modelo no es lo suficientemente general, ya que el resultado de un ensayo aleatorio no siempre se puede representar mediante intervalos. Por lo tanto, a veces es interesante generalizar los términos para representar de

forma coherente la información que se posee. Esta última apreciación conlleva a definir un modelo basado en conjuntos convexos de distribuciones de probabilidad o *Conjuntos Credales*. Tales conjuntos pueden entenderse como poliedros de espacio n-dimensional cerrados y convexos que representan distribuciones de probabilidad. Dicho conjunto puede determinarse a partir de varias restricciones lineales o mediante la enumeración de sus vértices, según el estudio que se esté llevando a cabo.

Un conjunto H de estas características en $\mathbb{R}^n$, dados cualesquiera p, $q \in H$ y $\alpha \in [0, 1]$, se puede definir como un conjunto convexo de distribuciones de probabilidad si verifica:

$$\alpha p + (1 - \alpha)q \in H \tag{3.1}$$

El espacio tiene una dimensión correspondiente al número de elementos o posibles estados de una variable aleatoria X. Se denota como $\mathcal{P}(X)$ al conjunto de todas las distribuciones de probabilidad sobre este atributo.

3.1.2. Teoría de la Evidencia

Además de la teoría clásica de la probabilidad, se encuentran otras propuestas que generalizan a esta y que representan la incertidumbre. Es el caso de la introducida por Shafer [53]. En los estudios previos al surgimiento de esta teoría, tanto Dempster como Shafer encontraron problemas a la hora de usar la teoría de la probabilidad clásica para poder representar la incertidumbre de los experimentos considerados, así como para manejar la necesidad de que la suma de las probabilidades asignadas a un suceso y a su contrario de la unidad.

La teoría de la Evidencia es a menudo llamada también como la teoría de Dempster-Shafer. Esta propuesta puede ser vista desde distintas, pero equivalentes, formas matemáticas. A grandes rasgos, se pueden identificar aquellas que se basan en la teoría de la probabilidad y otras en teorías axiomáticas. En este caso, se proporcionará desde el primer punto de vista comentado. Para su puesta en marcha no necesita un modelo de probabilidad completo, ya que intenta sacar beneficio en base a un conjunto de hipótesis en lugar de las mismas por separado. Además, se procura facilitar la reasignación de las probabilidades para cada suceso en las hipótesis cuando se cambian las evidencias y se persigue modelar la disminución del conjunto de hipótesis de trabajo a partir de la acumulación de tales evidencias.

Esta teoría supone que existe un conjunto exhaustivo de hipótesis mutuamente exluyentes $\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$, denominado *Marco de Discernimiento* y sobre el que se pretende razonar considerando el impacto de las evidencias que puedan aparecer. Por otra parte, se ha de considerar el impacto de las evidencias, no solo sobre las hipótesis individuales originales, sino también sobre los grupos de éstas, que corresponden a subconjuntos de Θ y conforman también otras hipótesis. De esta manera las nuevas hipótesis son las posibles disyunciones de

las hipótesis originales.

Dicha teoría utiliza una función m para un conjunto finito X, tal que $m : \wp(\Omega_X) \subseteq [0,1]$, denominada *Asignación Básica de Probabilidad*, que cumple las siguientes propiedades (Abellán, Klir, and Moral [10]):

- $m(\emptyset) = 0$
- $\sum_{A \subseteq \wp(\Omega_X)} m(A) = 1$

Cualquier conjunto $A \subseteq \wp(\Omega_X)$ tal que $m(A) > 0$ se le conoce como *elemento focal* de m y al conjunto de elementos focales de una Asignación Básica de Probabilidad m se denota como $\mathcal{F}_m$. Por su parte, $m(A)$ representa el grado de creencia sobre un conjunto A, es decir, la certeza de que la variable X bajo estudio tome valor en A. Sin embargo no se distingue sobre la creencia de los distintos elementos de A, como ocurriría en una distribución asociada a la teoría clásica de la probabilidad.

Por otra parte, existe lo que se denomina como un *cuerpo de evidencia*, es decir, un par de medidas $(\mathcal{F}_m, m)$. Consta de una medida de creencia, denotada por bel y otra de plausibilidad, pl, definidas en todos los conjuntos A tal que $A \subseteq \wp(\Omega_X)$ de la siguiente forma:

$$\begin{aligned} bel(A) &= \sum_{B \subset A} m(B), \\ pl(A) &= \sum_{B \bigcap A \neq \emptyset} m(B) \end{aligned} \tag{3.2}$$

La primera ecuación anterior puede ser interpretada como el grado seguro de creencia de que el verdadero valor de la variable A pertenece a A verdaderamente y la segunda como el mayor grado de creencia de que el verdadero valor de X está en A. Realmente figuran como las probabilidades superior e inferior de A, como se introdujo en Dempster [24]. Además, de estas dos expresiones 3.2 se sabe también que:

$$pl(A) = 1 - bel(A^c) \tag{3.3}$$

que demuestra la dualidad de estas medidas y lo que da lugar a un intervalo de creencia sobre cada subconjunto A, de la forma $(bel(A), pl(A))$, ya que $pl(A) \geq bel(A)$.

Según demostró Shafer [53], las evidencias son capacidades Choquet de orden infinito. Esto implica cumplir la siguiente propiedad:

$$pl(A_1 \cap A_2 \cap ... \cap A_n) \leq \sum_i pl(A_i) - \sum_{i \leq j} pl(A_i \cup A_j) + ... + (-1)^{n+1} pl(A_1 \cup A_2 \cup ... \cup A_n),$$
$$bel(A_1 \cup A_2 \cup ... \cup A_n) \geq \sum_i bel(A_i) - \sum_{i \leq j} bel(A_i \cap A_j) + ... + (-1)^{n+1} bel(A_1 \cap A_2 \cap ... \cap A_n)$$

Estas últimas expresiones pueden ser vistas como un sistema de intervalos de probabilidad de la forma (bel, pl) que al ser capacidades de orden dos son, por tanto, también coherentes. Como tales, siempre se puede calcular su conjunto convexo asociado (Dempster [24]). De esta forma, se sabe que cada Asignación Básica de Probabilidad sobre X con un conjunto finito de posibles estados tiene asociado un conjunto convexo de distribución de probabilidad C de la siguiente forma:

$$C = \left\{ p \in \wp(\Omega_X) | bel(A) \leq \sum_{x \in A} p(x) \leq pl(A);\ \ \forall A \subseteq \wp(\Omega_X) \right\} \tag{3.4}$$

3.1.3. Intervalos de probabilidad

El tipo de probabilidades que tienen la misma estructura que la obtenida en el ejemplo ilustrado del punto 3.1 dan pie a crear una definición formal de probabilidades superiores e inferiores dentro de Conjuntos Credales. De nuevo, Klir [36] es quien la expone.

Sea X una variable que toma valores en Ω_X=$\{x_1$,x_2,...,$x_n\}$. Un sistema de intervalos de probabilidad es una familia de intervalos:

$$L = \{[l_i, u_i] : i \in \{1, 2, ..., n\}\}, \tag{3.5}$$

verificando que $0 \leq l_i \leq u_i \leq 1$. Entonces al subconjunto $\mathcal{M}$ de todas las distribuciones de probabilidad sobre Ω_X, $\mathcal{P}(\Omega_X)$, definido como:

$$\mathcal{M} = \{p \in \mathcal{P}(\Omega_X) | l_i \leq p_i \leq u_i, \forall i\} \tag{3.6}$$

expresando a p_i=$p(\{x_i\})$, se le denomina conjunto convexo asociado al conjunto de intervalos de probabilidad L.

Una condición para que el conjunto sea no vacío, y al igual que en probabilidades inferiores y superiores no haya "pérdida segura" es:

$$\sum_i l_i \leq 1 \leq \sum_i u_i \tag{3.7}$$

Es posible caracterizar la coherencia, tal y como fue definida, ahora para conjuntos de intervalos de probabilidad, a través de la condición de intervalos de probabilidad *alcanzables* que son aquellas que verifican:

$$\begin{aligned} \sum_{j\neq i}(l_j) + u_i \leq 1, \\ \sum_{j\neq i}(u_j) + l_i \geq 1 \end{aligned} \tag{3.8}$$

para cualquier i. Por lo que existirán probabilidades en $\mathcal{M}$ que tengan como valores a los l_i y u_j para cualesquiera componentes $i,j\in\{1,...,n\}$. La coherencia es completamente equivalente a tener un sistema de intervalos alcanzables. Si no tenemos asegurada la coherencia es posible obtenerla a través de la siguiente propiedad:

Propiedad 1 *Dado un conjunto de intervalos de probabilidad* $L=\{[l_i,u_i]: i\in\{1,2,...,n\}\}$ *se define* $L'=\{[l'_i,u'_i]: i\in\{1,2,...,n\}\}$ *donde:*

$$l'_i = max_i\{l_i, 1 - \sum_{i\neq j} u_j\} \tag{3.9}$$

$$u'_i = min_i\{u_i, 1 - \sum_{i\neq j} l_j\} \tag{3.10}$$

determinan el mismo conjunto de probabilidades, $\mathcal{M}=\mathcal{M}'$*, siendo* $\mathcal{M}'$ *conjunto de* **intervalos alcanzables**.

También, existen operaciones interesantes de intervalos de probabilidad, así como su relación con otros modelos:

- Los intervalos de probabilidad son un caso concreto de probabilidades inferiores y superiores, puesto que dado un conjunto de intervalos de probabilidad L se puede obtener el par $\{\overline{P},\underline{P}\}$:

  $$\begin{aligned} \underline{P}(A) = inf_{p\in P(\omega)}\{p(A)\} \\ \overline{P}(A) = sup_{p\in P(\omega)}\{p(A)\} \end{aligned} \tag{3.11}$$

  en cambio un conjunto de probabilidades superiores e inferiores en general, no dan un conjunto de intervalos de probabilidad.

- Los intervalos de probabilidad son siempre un tipo de capacidad orden 2.

3.2. Medidas de incertidumbre

En los últimos años, se han presentado estudios sobre las propiedades y el comportamiento de medidas de información e incertidumbre en teorías más generalistas que la probabilidad. En una primera instancia, surgió, como medida de la cuantificación de la información e incertidumbre la medida de entropía de Shannon (ver 2.3). Posteriormente a este avance, surge la posibilidad de ampliar las definiciones dadas por la anterior medida a problemas donde se hacen uso de probabilidades imprecisas. Así, principalmente destacan ejemplos como los proporcionados por la Teoría de la Evidencia (Dempster and Shafer Theory), donde se hace una división de la medida de incertidumbre en dos tipos: el conflicto y la no-especificidad y, por su parte, la teoría general de los conjuntos cerrados y convexos de distribuciones de probabilidad, también conocidos como Conjuntos Credales, comentados y analizados en este capítulo.

3.2.1. Conflicto y no especificidad

Es necesario, en primer lugar, contextualizar y localizar bien el surgimiento de conceptos como los que definen el título de esta sección y además aprenderlos desde varios puntos de vista si es posible. Trabajos como los aportados por Abellán [2] definen muy bien este aspecto: para situaciones en las cuales la representación probabilística tradicional no es adecuada, la función de entropía de Shannon (descrita en el apartado 2.2.1.1) ha servido como punto de partida para el desarrollo de otro tipo de funciones que miden la cantidad de incertidumbre. Muchas de estas teorías dentro del terreno de las probabilidades imprecisas están basadas en la generalización de la teoría de la probabilidad, como son la teoría de Dempster-Shafer (Dempster [24] y Shafer [53]), intervalos de probabilidad alcanzables (De Campos, Huete, and Moral [23]), capadidades de orden 2 (Choquet [22]), probabilidades superior-inferior (Suppes [56] y (Fine [28])), etc. Cada una de estas representan un tipo específico de Conjunto Credal y por supuesto, constituyen conjuntos cerrados y convexos de distribuciones de probabilidad con un conjunto finito de puntos extremos. La figura B.3 muestra un resumen las principales propuestas anteriores.

Esta situación de incertidumbre generada por la creación de esta representación involucra dos tipos de medidas, denominadas *conflicto o aleatoriedad* y *no especificidad*. En cualquiera de las teorías o Conjuntos Credales anteriormente comentados, un índice que sea capaz de medir la incertidumbre debe cuantificar las partes de conflicto y no-especificidad a su vez. Por ejemplo, en la teoría de Dempster-Shafer (DST) comienzan el estudio de estas dos componentes, donde se define al conflicto como *la parte de incertidumbre asociada a los casos donde la información se centra en conjuntos con instersecciones vacías*, mientras que la no-especificidad es catalogada como *otra fuente de la comentada incertidumbre asociada a aquellas situaciones*

donde la información se centra en conjuntos en los cuales su cardinalidad[1] *es mayor que uno*.

El funcionamiento de la entropía de Shannon para probabilidades llevó a estudiar funciones similares en teorías más generales. Los primeros estudios en la Teoría de la Evidencia llevaron a considerar que se encontraban dos tipos de incertidumbre, una de tipo conflicto y otra de tipo no especificidad. Klir [36] propone una medida agregada, es decir, un funcional S^* el cual para cada índice de creencia Bel, dentro de la Teoría de Dempster-Shafer es definido por el máximo valor de la entropía de Shannon dentro del conjunto de todas las funciones de distribución de probabilidad que están asociadas a una función de creencia. Formalmente (Abellán, Klir, and Moral [10]):

$$S^*(Bel) = \max_{p \in C} \left[-\sum_{x \in X} p(x) log_2 p(x) \right] \tag{3.12}$$

donde C denota el conjunto de todas las distribuciones de probabilidad, p, tales que $Bel(A) \leq \sum_{x \in A} p(x)$ para todo $A \subseteq X$. Sin embargo, S^* no es aceptable ya que no muestra una clara separación entre las partes de conflicto y no especificidad. Por ello se propuso la siguiente expresión:

$$S^* = GH + GS \tag{3.13}$$

donde GH y GS denotan las generalizaciones de la medida de Hartley (como medida de no especificidad) y la entropía de Shannon (como medida de conflicto). Por lo tanto, la medida total de incertidumbre, TU, se define como el par:

$$TU = (GH, GS) \tag{3.14}$$

A partir de aquí, se define S^* como una *medida de entropía superior*. En Klir and Smith [37] se sugiere que esta medida puede ser usada no solo para los Conjuntos Credales asociados con la Teoría de la Evidencia, sino también para el resto. Además, la medida de incertidumbre total anteriormente comentada se formula como:

$$TU = (GH, S^* - GH) \tag{3.15}$$

[1]La cardinalidad de un conjunto A corresponde al número de elementos de dicho conjunto. Suele denotarse como n(A).

Por otra parte, en Abellán and Moral [3] se analiza otro tipo de medida, denominada en este caso *entropía inferior*. Esta representa el grado de contradicción interna y tiene una expresión similar a la función E, usada en la Teoría de la Evidencia para intentar generalizar la entropía de Shannon:

$$E(m) = -\sum_{A \subseteq X} m(A) log_2 Pl|A| \tag{3.16}$$

Con todo, finalmente en Abellán, Klir, and Moral [10] se llega a la conclusión de que la forma de cuantificar la incertidumbre de cada Conjunto Credal de manera desagregada o dividida se puede hacer mediante dos partes bien diferenciadas: por un lado, una medida de no especificidad, obtenida a través de la diferencia entre su entropía superior e inferior; por otro, una medida de conflicto, representada por la propia entropía inferior. Se demuestra que tal separación de la medida de incertidumbre, al contrario que otras propuestas, puede ser aplicada a cualquier tipo de Conjunto Credal, debido a que ambas componentes cumplen una serie de propiedades que les permiten funcionar correctamente. En definitiva, la nueva medida total de la incertidumbre queda reflejada como:

$$TU = (S^* - S_*, S_*) \tag{3.17}$$

3.3. Creación de árboles con probabilidades imprecisas

En Walley [60] se desarrolló una teoría basada en probabilidades imprecisas atribuida al propio Walley, mediante la cual se representan las probabilidades de cada suceso por intervalos. Esta teoría tiene como principal objetivo inferir sobre una variable categórica. Además, ha sido utilizado por Abellán and Moral [7], [1], así como Mantas and Abellán [42] para introducir las medidas de incertidumbre dentro de los problemas de clasificación.

3.3.1. El modelo multinomial

Sea $\Omega = \{x_1, x_2, ..., x_k\}$, $\forall k \geq 2$ el espacio muestral, exhaustivo y mutuamente excluyente, asociado a un experimento de probabilidad cualquiera. Este modelo se basa en la realización de N observaciones independientes del conjunto Ω con la misma distribución de probabilidad para todas ellas, esto es, $P(x_j) = \theta_j$, $\forall j = 1, 2, ..., k$; y donde $\theta_j \geq 0$ y $\sum_j \theta_j = 1$.

Sea n_j la frecuencia asociada al valor x_j, por lo que $\sum_j n_j = N$. Ahora bien, si se tiene el vector $n = (n_1, n_2, ..., n_k)$, de acuerdo con las probabilidades multinomiales se asocia una función de verosimilitud sobre los valores del parámetro $\theta = (\theta_1, \theta_2, ..., \theta_k)$, dada por:

$$L(\theta|n) \propto \prod_j \theta_j^{n_j} \tag{3.18}$$

3.3.2. La distribución de Dirichlet

El modelo de Dirichlet permite modelizar experimentos asociados a vairables aleatorias que toman valores entre cero y uno y cuya suma es igual a la unidad. Esto permite estimar los valores de una distribución de probabilidad desconocida.

Tomando un punto de vista bayesiano, la distribución a priori de Dirichlet de parámetros s y t del vector θ, donde $t = (t_1, t_2, ...t_k)$ tiene la siguiente forma:

$$\pi(\theta|n) \propto \prod_j \theta_j^{st_j - 1} \tag{3.19}$$

para $s > 0$, $0 < t_j < 1$, $\forall j = 1, 2, ..., k$ y $\sum_j t_j = 1$. La constante que falta para obtener el valor de π se obtiene aplicando que la integral sobre todos los valores de θ vale uno[2].

Multiplicando la función de verosimilitud de la distribución multinomial dada por 3.18 por la distribución de Dirichlet anterior se obtiene la función de densidad a posteriori siguiente:

$$\pi(\theta|n) \propto \prod_j \theta_j^{n_j + st_j - 1} \tag{3.20}$$

la cual corresponde a una distribución de Dirichlet de parámetros $N + s$ y t^*, con $t_j^* = (n_j + st_j)/(N + s)$.

3.3.3. Modelo de Dirichlet Impreciso (IDM)

Se pueden obtener los valores superiores e inferiores de la distribución de probabilidad a posteriori para un suceso cualquiera. Si se denota A_j al suceso en el cual el valor x_j se manifiesta en una determinada prueba, estos valores:

$$\begin{aligned} \overline{P}(A_j|n) &= \frac{n_j + s}{N + s}, \\ \underline{P}(A_j|n) &= \frac{n_j}{N + s} \end{aligned} \tag{3.21}$$

[2]Esta no es la expresión normal de la distribución de Dirichlet, ya que se debe de fijar previamente el valor de s, tomando generalmente los parámetros $\alpha_j = st_j$

alcanzándose cuando $t_j \to 1$ y cuando $t_j \to 0$, respectivamente. Entonces, para un suceso cualquiera A, con frecuencia $n(A) = \sum_{x_j \in A} n_j$ la estimación del valor de la probabilidad de A, $P(A|n)$, bajo la distribución a posteriori de Dirichlet de parámetros $N + s$ y t^* tiene un valor esperado:

$$P(A|n) = \frac{n(A) + st(A)}{N + s} \tag{3.22}$$

que produce las siguientes probabilidades superior e inferior:

$$\begin{aligned} \overline{P}(A|n) &= \frac{n(A) + s}{N + s}, \\ \underline{P}(A|n) &= \frac{n(A)}{N + s} \end{aligned} \tag{3.23}$$

El valor del parámetro s, según Walley [60], define el *número de observaciones ocultas (no ocurridas)* y le otorga un valor de 1 o 2. Dicho valor representa la máxima cantidad para la cual aceptar la frecuencia para un suceso que no ocurre.

3.3.4. Medidas de incertidumbre para IDM

3.3.4.1. Conocimientos previos

Supóngase una variable aleatoria Z cuyos posibles valores son $\{z_1, z_2, ..., z_k\}$. Además, téngase en cuenta una distribución de probabilidad $p(z_j)$; $\forall j = 1, ..., k$ asociada a dicha variable. Se calcula un intervalo de probabilidad a partir del conjunto de datos para cada valor z_j de la variable Z. En virtud del Modelo de Dirichlet Impreciso (IDM) se estima que las probabilidades de cada posible estado z_j están dentro del intervalo:

$$p(z_j) \in \left[\frac{n_{z_j}}{N + s}, \frac{n_{z_j} + s}{N + s} \right]; \;\; j = 1, ..., k \tag{3.24}$$

donde n_{z_j} es la frecuencia dada en el conjunto de datos para todos los valores z_j de la variable Z. Así mismo, N es el valor asociado al tamaño de muestra y s un hiperparámetro que no depende del espacio muestral (Walley [60]). Su valor determina la rapidez con la que los valores de probabilidad superior e inferior convergen cuando el tamaño de muestra aumenta. Cuanto mayor sea el valor de este último, menos amplio será el intervalo resultante, es decir, se obtendrán estimaciones más precisas y mayor serán las precauciones a tomar a la hora de realizar inferencia. En Walley [60] no se proporciona un valor definitivo para este parámetro, pero se sugieren dos candidatos: $s = 1$ o $s = 2$. El autor aconseja asignar un valor de 1.

Esta representación origina un conjunto convexo de distribuciones de probabilidad específico

para la variable Z, denotado por $K(Z)$. El conjunto es definido como sigue:

$$K(Z) = \left\{ p | p(z_j) \in \left[\frac{n_{z_j}}{N+s}, \frac{n_{z_j}+s}{N+s} \right] ; \ j = 1, ..., k \right\} \tag{3.25}$$

3.3.4.2. Máxima entropía de un Conjunto Credal

Haciendo uso de la teoría mostrada en la anterior sección, más concretamente en la expresión 3.25, se pueden definir y dar paso a medidas de incertidumbre sobre conjuntos convexos de distribuciones de probabilidad. La entropía de estos conjuntos se estima a partir del máximo de dicha función para todas las distribuciones de probabilidad que pertenecen a este conjunto. Esta nueva expresión, denotada por H^*, tiene la siguiente estructura:

$$H^*(K(Z)) = max \left\{ H(p) | p \in K(Z) \right\} \tag{3.26}$$

donde H es la función de entropía de Shannon y H^* una medida de incertidumbre total bien adaptada para este tipo de conjuntos. Su cálculo conlleva un coste computacional bajo, para valores del hiperparámetro $s \in [1, 2]$ de la función asociada a conjuntos convexos (ver 3.25). Siéndose más específico y dado por las características propias del Modelo de Dirichlet Impreciso (IDM), se alcanza el coste más leve para $s = 1$. Además, el calcular H^* resulta bastante simple con este valor.

A modo de intentar explicar cómo funciona este cálculo se puede comenzar por determinar, en primer lugar, los elementos del conjunto A. Los integrantes de dicho conjunto comparten la cualidad de que la frecuencia de la variable en cuestión coincide con el del atributo con menos ejemplos de entre todos los existentes del dataset, en otros términos:

$$A = \left\{ z_j | n_{z_j} = min_i \left\{ n_{z_i} \right\} \right\} ; \ \forall i, j = 1, ..., k \tag{3.27}$$

Por tanto, la distribución que tiene la mayor entropía es:

$$p^*(z_i) = \begin{cases} \frac{n_{z_i}}{N+s} & \text{Si } z_i \notin A \\ \frac{n_{z_i}+s/n(A)}{N+s} & \text{Si } z_i \in A \end{cases} \tag{3.28}$$

Siendo $n(A)$ el número de elementos o cardinalidad del conjunto A, $\forall i = 1, ..., k$. Como se ha comentado con anterioridad, los intervalos que recogen la imprecisión son más amplios cuando el tamaño de muestra es más pequeño, con lo cual, es de esperar valores de H^* grandes con respecto a los que se obtendrían con muestras de mayor tamaño. Esta propiedad será importante para diferenciar el rendimiento de los CDT en comparación con el comportamiento de otros tipos de Árboles de Decisión. Además, se ha de usar un valor de $s = 1$. Para valores mayores de este parámetro se puede usar el algoritmo de Abellán [2].

3.3.5. Algoritmos de creación de Árboles de Decisión basados en probabilidades imprecisas

3.3.5.1. Credal Decision Trees

Los Árboles Credales (en inglés *Credal Decision Trees*) son algoritmos dedicados al diseño de clasificadores basado en probabilidades imprecisas y medidas de incertidumbre dentro de la familia de los Árboles de Decisión. Los cálculos computacionales correspondientes construyen ramificaciones para resolver problemas de clasificación, asumiendo que el conjunto de entrenamiento no es demasiado fiable. Este algoritmo es especialmente adecuado para clasificar conjuntos de datos en los que existe ruido en la variable clase (Marios Polycarpou André C.P.L.F. de Carvalho [44]).

3.3.5.1.1. Construcción de un CDT

Haciendo referencia de nuevo al trabajo aportado por Marios Polycarpou André C.P.L.F. de Carvalho [44], se explica que el proceso de selección de variables para el algoritmo de Árboles Credales está basado en probabilidades imprecisas y medidas de incertidumbre en Conjuntos Credales, es decir, conjuntos cerrados y convexos de distribuciones de probabilidad. De esta manera el algoritmo considera que el conjunto de entrenamiento no es de calidad cuando dicho proceso es llevado a cabo.

La construcción de este tipo de estructuras es similar a la de los clásicos, explicados en previos capítulos. Estos nuevos árboles comienzan con un espacio vacío, donde, para ramificar en cada nodo, se selecciona la variable del conjunto de datos que tiene el menor grado de incertidumbre total con respecto a la variable clase. En teoría de la probabilidad, dicha ramificación siempre implica una reducción en la entropía. Es, por lo tanto, necesario introducir un criterio adicional con el fin de no crear modelos excesivamente complejos con dependencia dentro de los datos. Dicho criterio es el de la incertidumbre total, ya comentada en la sección 3.2.1 y 3.2, es decir, una función de entropía que cuantifica el grado de conflicto y no especificidad.

El criterio de división llevado a cabo en árboles de este tipo y que lo diferencia de otros Árboles de Decisión como ID3, es el de la IIG (*Imprecise Info-Gain* o Ganancia de Información Imprecisa en castellano). Este está basado en la aplicación de medidas de incertidumbre sobre conjuntos convexos de distribuciones de probabilidad. Para ser más preciso, los intervalos de probabilidad son extraídos del conjunto de datos para cada valor c de la variable clase C usando el Modelo Dirichlet Impreciso (IDM) de Walley [60], el cual representa un tipo de distribuciones de probabilidad específico con la que se estima la entropía máxima (Abellán [2]). Esta última es una medida de incertidumbre bien conocida para este tipo de conjuntos y expresada en la fórmula 3.26 (Abellán, Klir, and Moral [10]).

Algorithm 1 Building a CDT

Input: $(N_o, \mathcal{L})$
Output: CDT($N_o, \mathcal{L}$)

if $\mathcal{L} = \emptyset$ **then**
    Exit
Let $\mathcal{D}$ be the partition associated with node N_o
Compute the value: $\alpha = max_{X_j \in \mathcal{L}} \left\{ IIG^{\mathcal{D}} \left\{ C, X_j \right\} \right\}$
if $\alpha \leq 0$ **then**
    Exit
else
    Let X_l be the variable for which the maximum α is attained
    Remove X_l from $\mathcal{L}$
Assign X_l to node N_o
for each possible value x_l of X_l **do**
    Add a node N_{ol}
    Make N_{ol} a child of N_o
    Call CDT($N_{ol}, \mathcal{L}$)

Este nuevo criterio se ha de definir en el contexto de problemas de clasificación. Sea C la anteriormente comentada variable clase y $\{X_1, X_2, ..., X_m\}$ el resto de atributos del conjunto de datos.

La Ganancia de Información Imprecisa de cualquier variable X del mismo es:

$$IIG^{\mathcal{D}}(C, X) = H^*(K^{\mathcal{D}}(C)) - \sum_i P(X = x_i) H^*(K^{\mathcal{D}}(C|X = x_i)) \tag{3.29}$$

en el cual, $H^*()$ determina la máxima entropía de los Conjuntos Credales $K^{\mathcal{D}}(C)$ y $K^{\mathcal{D}}(C|X = x_i)$, respectivamente. Estos últimos conjuntos se obtienen gracias al Modelo de Dirichlet Impreciso tanto para la variable clase C como para la misma condicionada al valor del atributo i-ésimo $(C|X = x_i)$. Además, $P(X = x_i)$; $\forall i = 1, ..., n$ denota la distribución de probabilidad precisa correspondiente. La restante expresión, $\mathcal{D}$, hace referencia a la partición del dataset con la cual se calcula esta medida. Por último, se puede demostrar que la Ganacia de Información clásica corresponde a un caso particular de esta otra medida, sin más que tomar $s = 0$ en 3.25. Entonces, la construcción del Árbol de Decisión bajo este criterio se lleva a cabo mediante la selección en cada nodo del atributo X que verifica (Abellán and Masegosa [4]):

$$X = argMin_{\{X_1, X_2, ..., X_m\}} \sum_i P(X = x_i) H^*(K^{\mathcal{D}}(C|X = x_i)) \tag{3.30}$$

Como se puede apreciar, las expresiones utilizadas, por un lado, para obtener la Ganancia de Información vista en el capítulo anterior y, por otro, la Ganancia de Información Imprecisa

vista en este difieren entre sí. En este caso, esta última medida está basada en el principio de máxima incertidumbre o máxima entropía, ampliamente usado en la teoría de la información clásica (Jaynes [35]). El uso de esta medida se justifica en Abellán and Moral [1] para usarse en el procedimiento de construcción de Árboles de Decisión Credales.

Además, es necesario añadir que esta nueva medida de información puede obtener resultados negativos, dado un atributo X y una partición $\mathcal{D}$. Esto no es una característica propia de los indicadores clásicos (Ganancia de Información o la Tasa de Ganancia de Información). Esto permite al criterio IIG hallar los atributos que perturban la información con la variable clase y de este modo poder considerar con certeza una hipotética poda a posteriori.

Como ya se conoce, cada nodo del Árbol, denotado por N_o, provoca una partición del conjunto de datos. Por ejemplo, para el nodo raíz, $\mathcal{D}$ es concebido como el conjunto de datos completo. Además, cada nodo N_o tiene una lista asociada $\mathcal{L}$ que contiene etiquetas de atributos (que no están en el camino desde el nodo raíz hasta N_o).

Un complemento esencial para todos estos conceptos viene especificado en Mantas and Abellán [42], donde se adjunta el pseudocódigo (ver el Algoritmo 1) que configura la construcción de este tipo de Árboles de Decisión. Si se atiende a este, se comprende que la situación expresada como *Exit* se puede dar o bien cuando no quedan más atributos a insertar en nuevos nodos (caso en que $\mathcal{L} = \emptyset$, paso 1) o cuando el cálculo de la medida de incertidumbre no obtiene una reducción con respecto a la anterior (paso 5), y por lo tanto, tanto en uno como en otro caso se crean nodos hojas para finalizar la construcción. Sin embargo, si esto no ocurre, se crea un nuevo nodo correspondiente al atributo X_l y se elimina tal variable de la lista $\mathcal{L}$ (pasos 8 y 9).

Además, el criterio de parada para finalizar la construcción de los Árboles Credales está basado también en las componentes de la entropía de conflicto y no especificidad, es decir, cuando en la ramificación siguiente se produce un aumento de la medida de incertidumbre (es decir, que un descenso en el primero no está compensado por un aumento del segundo) se decide finalizar la construcción. Finalmente, debería de usarse un criterio basado en las frecuencias para obtener el valor de la variable para ser clasificada en la correspondiente hoja/nodo. Como ya se sabe, una vez se alcanza un nodo hoja, se incorpora al Árbol el estado o valor de la variable clase más probable para la partición del conjunto de datos asociada con tal nodo. En otras palabras, la etiqueta de la clase para el nodo hoja N_o asociado a la partición $\mathcal{D}$ es:

$$Class(N_o, \mathcal{D}) = \underset{c_i \in C}{max} |\{I_j \in \mathcal{D}/class(I_j) = c_i \;\; j = 1, ..., |\mathcal{D}|\}| \tag{3.31}$$

donde $class(I_j)$ es la clase de la instancia $I_j \in \mathcal{D}$ y $|\mathcal{D}|$ corresponde al número de instancias de $\mathcal{D}$. En el caso de que no se tenga un único estado mayoritario, el nodo hoja quedaría sin clasificar. Para evitar este problema, se le asigna el valor de la clase que posee su nodo padre, como Abellán and Masegosa [4] indican.

Los Árboles Credales pueden resultar de menor tamaño que otros producidos con el uso de algoritmos clásicos de Árboles de Decisión (Abellán and Masegosa [5]). Este hecho normalmente produce una reducción del sobreajuste (*overfitting*) del modelo y con ello, todas las consecuencias positivas derivadas y ya vistas en anteriores capítulos (Abellán and Moral [1]). Por otro lado, tanto este tipo de árboles como el criterio IIG han sido utilizados con éxito en otras tareas y herramientas de minería de datos, por ejemplo, como parte del procedimiento de selección de variables, en clasificación de conjuntos de datos con ruido en la variable clase, así como, la renovación de dicho criterio para la definición de un clasificador semi-naïve.

3.3.5.2. Complete Credal Decision Trees

Los Árboles de Decisión hasta ahora expuestos han sido vistos tanto desde una perspectiva clásica, donde se han comentado las principales aportaciones desde sus inicios hasta hoy día, como desde otra que como se ha podido comprobar surge como alternativa en determinadas situaciones, como por ejemplo, en las que la base de datos a usar contiene ruido en la variable clase. Sin embargo, este último hecho también puede coexistir en el resto de atributos del dataset, por lo que puede afectar al rendimiento del algoritmo en sí mismo si no se actúa frente a este problema. En otras palabras, se pueden presentar datasets en los cuales la información que se posee tanto para la variable clase como para el resto de atributos de entrada es considerada poco fiable de cara a la futura clasificación, por lo cual podría resultar interesante estudiar cada una de las distribuciones de probabilidad asociadas basándose en la teoría de la imprecisión o de las probabilidades imprecisas.

En Mantas and Abellán [42] y [43] se habla de una nueva adaptación de los Árboles Credales que se nombran como *Complete Credal Decision Trees (CCDT)*. En ellos, todos los atributos del conjunto de datos, incluida la clase, están modelados por el Modelo de Dirichlet Impreciso (IDM). De este modo, el valor de cada variable se estima en base a un conjunto convexo de distribuciones de probabilidad. El procedimiento para la construcción de un CCDT es semejante al empleado por los Árboles Credales, salvo el empleo del criterio IIG para dividir el espacio por el *Complete Imprecise Info-Gain (CIIG)*, que se define como:

$$CIIG^{\mathcal{D}}(C,X) = H^*(K^{\mathcal{D}}(C)) - \sum_i P^{\mathcal{D}}(X=x_i)H^*(K^{\mathcal{D}}(C|X=x_i)) \tag{3.32}$$

en el cual, $P^{\mathcal{D}}(X=x_i)$, $\forall i=1,...,n$ es la distribución de probabilidad perteneciente al conjunto convexo $K^{\mathcal{D}}(X)$. Los CCDTs funcionan usando el principio de máxima incertidumbre para elegir la distribución $P^{\mathcal{D}}$ adecuada del conjunto $K^{\mathcal{D}}(X)$ (es decir, la distribución de probabilidad con la mayor entropía), al igual que el resto de CDTs. De este modo, se garantiza la construcción de árboles de menor tamaño que los generados por las propuestas clásicas y

adaptados, en general, para aquellos conjuntos de datos con presencia de ruido.

Algorithm 2 Building a CompleteCredalTree

Input: $(N_o, \mathcal{L})$
Output: CCDT($N_o, \mathcal{L}$)

if $\mathcal{L} = \emptyset$ **then**
    Exit
Let $\mathcal{D}$ be the partition associated with node N_o
Calculate $P^{\mathcal{D}}(X = x_i)$; $\forall i = 1, ..., n$ on the convex set $K^{\mathcal{D}}(X)$
Compute the value $\alpha = max_{X_j \in \mathcal{L}} \left\{ CIIG^{\mathcal{D}} \left\{ C, X_j \right\} \right\}$
if $\alpha \leq 0$ **then**
    Exit
else
    Let X_l be the variable for which the maximum α is attained
    Remove X_l from $\mathcal{L}$
Assign X_l to node N_o
for each possible value x_l of X_l **do**
    Add a node N_{ol}
    Make N_{ol} a child of N_o
    Call CCDT($N_{ol}, \mathcal{L}$)

Una de las principales ventajas que plantea el uso de este tipo de árboles, y en general de los Árboles Credales, es precisamente esta última cuestión comentada. El hecho de tener árboles de menor tamaño puede resultar fundamental en problemas donde otros algoritmos generan reglas para clasificar más complejas. La razón por la cual esto es así es por la posibilidad de obtener valores negativos en el uso de los criterios CIIG e IIG, respectivamente. Como ambos tipos de algoritmos funcionan de manera similar, se puede directamente concluir que:

$$CIIG^{\mathcal{D}}(C, X) \leq IIG^{\mathcal{D}}(C, X) \tag{3.33}$$

Por lo tanto, la propiedad vista para los CDTs se ve reforzada aún más para el caso de los CCDTs. Sin embargo, dicha cuestión no se cumple en todos los casos, como Mantas and Abellán [42] comentan. Además, este algoritmo se definió sin poda de ningún tipo.

En estudios realizados por los autores comentados en este punto se ha demostrado la efectividad de los CDTs y sobre todo de los CCDTs frente al problema de la existencia de ruido en distintos conjuntos de datos. Sin embargo, las diferencias entre estos últimos y otros algoritmos Credales no resultaban estadísticamente significativas. Si a esta situación se añade que la propia definición del algoritmo CCDT es de mayor complejidad que la del resto de CDTs, al considerar mayor cantidad de imprecisión, hace que en este trabajo se tome la decisión de no considerarse de cara a la futura fase experimental.

3.3.5.3. Credal-C4.5

A partir de toda la definición enunciada por la teoría anteriormente expuesta, los autores Mantas and Abellán [43] proponen un algoritmo que en esencia mantiene bastantes similitudes con el algoritmo C4.5 de Salzberg [50]. La principal diferencia es que esta nueva propuesta calcula probabilidades con respecto a los atributos y la clase del conjunto de datos mediante el uso de la teoría de probabilidades imprecisas. El proceso para crear un nuevo nodo en el árbol esta apoyado, como en el caso del resto de Árboles Credales, en el uso de la medida de incertidumbre para conjuntos de este tipo, comentada ya con profundidad en el anterior punto de este capítulo (3.29). En este sentido, no se hace uso de la Tasa de Ganancia de Información (2.9), sino la Tasa de Ganancia de Información Imprecisa o *Imprecise Info-Gain Ratio* (IIGR). Este nuevo criterio se define como sigue:

$$IIGR^{\mathcal{D}}(C,X) = \frac{CIIG^{\mathcal{D}}(C,X)}{H(X)} \tag{3.34}$$

donde C hace referencia a la variable clase y X a uno de los atributos de entrada disponibles. En el cálculo del $CIIG^{\mathcal{D}}(C,X)$ se ha de escoger la distribución de probabilidad $P^{\mathcal{D}}$ que tenga la expresión similar a la recogida en 3.28.

En el algoritmo Credal-C4.5 se considera que el conjunto de entrenamiento no es demasiado fiable en el sentido de que dichos datos pueden contener problemas en cuanto a ruido, bien en la variable clase, en el resto de atributos, o en ambas. Por lo tanto, este tipo de Árboles de Decisión son, en principio, ideales para problemas de Aprendizaje Automático en el cual existe este inconveniente.

El modo de proceder para construir un árbol de estas características es semejante al comentado en el apartado anterior sobre la construcción en general de un CDT. Los principales elementos de este algoritmo son:

- **Criterio para dividir:** Al igual que otros algoritmos, como C4.5, se elige el atributo con el mayor valor del ya comentado $IIGR$ y cuyo IIG es mayor que el IIG promedio del conjunto de atributos válidos (es decir, variables numéricas o cuyo número de estados es menor que el treinta por ciento de la cantidad de instancias que están en dicha ramificación).

- **Etiqueta de los nodos hoja:** El valor más probable de la variable clase en la partición $\mathcal{D}$ asociada se escoge como etiqueta para el nodo hoja. La expresión matemática para este proceso ya se ilustró en 3.31.

- **Criterio de parada:** El proceso de construcción se ve obstaculizado en cuanto la medida de incertidumbre vinculada a este algoritmo no se reduce para la siguiente ramificación (paso 6, pseudo-código 3) o cuando no existe un número mínimo de instancias por nodo

Algorithm 3 Building a CredalC4.5Tree

Input: $(N_o, \mathcal{L})$
Output: CredalC4.5($N_o, \mathcal{L}$)

```
1: if L = ∅ then
2:     Exit
3: Let D be the partition associated with node N_o
4: if |D| < minimum number of instances then
5:     Exit
6: Calculate P^D(X = x_i); ∀i = 1, ..., n on the convex set K^D(X)
7: Compute the value α = max_{X_j ∈ M} {IIGR^D {C, X_j}}
8: if α ≤ 0 then
9:     Exit
10: else
11:    Let X_l be the variable for which the maximum α is attained
12:    Remove X_l from L
13: Assign X_l to node N_o
14: for each possible value x_l of X_l do
15:    Add a node N_ol
16:    Make N_ol a child of N_o
17:    Call CredalC4.5(N_ol, L)
```

hoja (paso 3). Dicha construcción también se puede ver finalizada por no haber ningún atributo válido, como se comentó anteriormente, en el conjunto de datos.

- **Atributos numéricos:** Los atributos numéricos son tratados del mismo modo que lo hace el algoritmo C4.5, diferenciándose únicamente en el uso del criterio IIG en lugar del IG.

- **Valores perdidos:** La descripción para este contexto es la misma que la que se ha proporcionado en el punto anterior.

- **Post-poda:** Se emplea, al igual que en el algoritmo C4.5, el llamado *Pessimistic Error Prunning*.

Por último, en el código 3 se expone en la línea 7 el conjunto $\mathcal{M}$, el cual representa:

$$\mathcal{M} = \left\{ X_j \in \mathcal{L} / CIIG^{\mathcal{D}}(C, X_j) > avg_{X_j \in \mathcal{L}} \left\{ CIIG^{\mathcal{D}}(C, X_j) \right\} \right\} \tag{3.35}$$

Capítulo 4

Ensemble learning

Durante las últimas décadas se ha ido matizando el proceso de clasificación de nuevas instancias no etiquetadas. Dicho proceso de refinamiento ha dado lugar a utilizar varios clasificadores simultáneamente para resolver un mismo problema en lugar de uno solo, como a priori podría resultar más obvio.

Aunque originariamente fueron pensados para poder reducir la varianza creada por cada clasificador, (y, por lo tanto, aumentar la precisión) esta nueva metodología ha producido una gran expectación dentro de la comunidad de usuarios de este tipo de algoritmos, pues su puesta en marcha ha cubierto con gran efectividad un amplio abanico de problemas reales y enfoques técnicos propios de Machine Learning, como pueden ser la selección de características, aprendizaje incremental, clasificación no balanceada, etc. Si bien es cierto que la investigación y desarrollo de este concepto han sido muy intensos desde su aparición, hay que mencionar que este progreso ha comenzado muy recientemente y en cierta medida no existe en la literatura una gran variedad de trabajos que expongan un conocimiento contrastado y significativo.

Sin embargo, existen elaboraciones recientes, como el aportado por Zhang and Ma [64] en el que se comenta que el objetivo principal para la construcción de un ensemble es el de poder mejorar la fiabilidad de las decisiones tomadas por un modelo de clasificación mediante la ponderación y combinación de varios clasificadores a través de algún proceso ideado para alcanzar una decisión final (ver figuras B.7 y B.8). A pesar de ello, existen muchas otras aplicaciones y definiciones dentro del Aprendizaje Automático, como se ha comentado anteriormente.

Las principales diferencias que existen entre las distintas familias de ensembles se pueden localizar en torno a varios aspectos fundamentales, entre los cuales destacan los siguientes: cómo crear las muestras sobre el conjunto de todos los datos para conformar el conjunto de entrenamiento en cada clasificador, el modo de proceder para generar los clasificadores y, por último, la técnica de combinación para obtener la decisión final.

Este capítulo pretende exponer algunas de las propuestas más conocidas y usadas dentro de este mundo, así como proporcionar un conocimiento general sobre los clasificadores combinados

o *ensembles* que permita comprender el estudio experimental que se llevará a cabo en el siguiente capítulo sobre algoritmos de este tipo.

4.1. Boosting

Algunos autores ya habían sugerido algunas ideas, pero probablemente la primera vez que se demostró la funcionalidad de algoritmos de este tipo fue en 1989, gracias a Schapire [52]. Los primeros experimentos con estos primerizos ensembles corrieron a cargo de Drucker, Harris and Schapire [26] haciendo uso de redes neuronales. Estas bases dieron lugar a la construcción del algoritmo basado en combinación de clasificadores que hoy día se conoce por el nombre de AdaBoost (de los términos **Ada**ptative **Boost**ing) (Freund and Schapire [63]). Su primera aparición fue en 1995, donde se eliminaban muchas de las deficiencias que mostraban los primeros planteamientos en torno a algoritmos boosting.

El nombre de este tipo de algoritmos, al igual que otros tipos de ensembles como por ejemplo bagging, puede ser aplicado en muchos métodos de aprendizaje estadísticos de regresión y clasificación. En este caso, solo se comentarán los algoritmos boosting para Árboles de Decisión. Además, el desarrollo de AdaBoost ha sido extendido con el paso del tiempo a otras versiones (AdaBoost.M1 y AdaBoostM2), así como AdaBoost.R para los comentados problemas de regresión. En este caso se muestra el algoritmo AdaBoost.M1 que es el más conocido.

El funcionamiento básico del algoritmo es el siguiente (en el cuadro 4 se muestra el pseudocódigo para este algoritmo): se toma como entrada un conjunto de entrenamiento, donde se tienen N instancias etiquetadas, según una variable clase X que es binaria (por ejemplo, $X = 1$ se asocia a la clase positiva mientras que $X = 0$ a la negativa). El procedimiento es iterativo, con lo cual, se tienen en cuenta T etapas o repeticiones y con una distribución de pesos o ponderaciones para cada instancia, inicialmente con valor proporcional $1/N$. Esto último indica que todas las instancias tienen en principio la misma probabilidad de ser escogidas para formar parte de la muestra. En cada una de las iteraciones, se realiza la clasificación de cada instancia, y una vez obtenido el resultado en términos del error de clasificación mediante la suma de la distribución de los pesos de las instancias mal clasificadas, se recalculan los mismos, dando mayor importancia a aquellas que cumplen este criterio. AdaBoost.M1 requiere que este error sea menor o igual a $1/2$ para obtener β_t, ya que dicho valor oscila entre 0 y 1.

Entonces, y como se ha explicado, se procede a recalcular dicho peso, en el sentido de que aquella fila que no ha sido clasificada correctamente se le asigna un mayor peso mientras que a otra que haya sido correctamente etiquetada se le resta importancia. De esta forma el futuro clasificador de la etapa t_{i+1} se esfuerza en centrarse en mayor medida en este tipo de ejemplos del conjunto de entrenamiento que lo que lo hizo el clasificador de la etapa t_i. La decisión

Algorithm 4 AdaBoost.M1

Input: $\{x_i, y_i\}$ (training data), T (iteration number), $y_i \in \{\omega_1, ..., \omega_C\}$; $\forall i = 1, ..., N$
Output: Class with the highest V_c

1: Initial weights: $D_1(i) \leftarrow (1/N, ..., 1/N)$
2: **for** $t \leftarrow$ **to** T **do**
3: Draw training subset S_t from the distribution D_t.
4: Train a base classifier on S_t, receive hypothesis $h_t : X \rightarrow Y$
5: Compute the error of h_t: $\epsilon_t = \sum_i I\left[h_t(x_i \neq y_i)\right] D_t(x_i)$
6: **if** $\epsilon_t > 1/2$ **then**
7: **Stop**
8: Set: $\beta_t = \frac{\epsilon_t}{(1-\epsilon_t)}$
9: Update sampling distribution (I): $D_{t+1}(i) = \frac{D_t(i)}{\sum_i D_t(i)}\beta_t$, if $h_t(x_i) = y_i$
10: Update sampling distribution (II): $D_{t+1}(i) = \frac{D_t(i)}{\sum_i D_t(i)}$, otherwise
11: **return** Weighted Majority Voting: $V_c = \sum_{t:h_t(z)=\omega_c} log\left(\frac{1}{\beta_t}\right), c = 1, ..., C$

final correspondiente a la combinación de T clasificadores corresponde al método basado en voto mayoritario ponderado (o suma ponderada de cada peso con su respectivo resultado de clasificación). Es importante remarcar que en cada etapa del algoritmo se crea un Árbol de Decisión que es independiente del resto. Además, cada uno de ellos se construye en base a la información aportada por los Árboles construidos en las etapas anteriores.

4.2. Bagging

El algoritmo Bagging (en abreviación de los términos **B**ootstrap **Agg**regat**ing**) propuesto por Breiman [18] es uno de los más primitivos y simples, pero aún hoy día efectivo planteamiento basado en combinación de clasificadores.

Dado un conjunto de datos de entrenamiento D de cardinalidad $n(D)$ y una familia de clasificadores (tales como Árboles de Decisión, Redes Neuronales, etc), este algoritmo crea un modelo con T clasificadores independientes, cada uno a partir de un subconjunto de D de entrenamiento obtenido a partir de una operación de muestreo con reemplazamiento con un total de $n(D)$ instancias (o al menos un porcentaje de éste).

La diversidad proporcionada por este algoritmo está garantizada gracias a las variaciones que existen entre cada una de las réplicas o muestras bootstrap creadas para cada clasificador. Una vez se han creado cada uno de los modelos, para cada instancia del conjunto de test se realiza la predicción en cada uno de ellos y mediante la combinación de estos resultados, es decir, en el caso de problemas de clasificación que nos ocupa por voto mayoritario, se decide

Algorithm 5 Bagging

Input: D_m (dataset), $class(.)$ (state of class atribute), T (iteration number), N (sample size)
Output: $h^{(t)}$; $t = 1, ..., T$

1: **for** $t \leftarrow$ **to** T **do**
2: D_m^t — Sample N instances from D_m with replacement
3: $h^{(t)} \leftarrow class(D_m^t)$ — t Base classifier on D_m^t
4: **return** $f^{(T)}(\cdot) = \sum_{t=1}^{T} h^{(t)}(\cdot)$ — "Vote" of T classifiers

qué estado de la variable clase se toma.

Por último es importante destacar por qué este algoritmo aún está tan bien considerado. El hecho es que funciona con efectividad en muchos casos donde otros algoritmos o ensembles no. Esta propuesta suele ser útil cuando se tienen modelos sobreajustados, mientras que no lo es cuando existe gran cantidad de sesgo[1] existente. y el modelo es robusto frente a cambios en el conjunto de entrenamiento (debido a la fase de muestreo).

4.3. Random Forests

Este último tipo de clasificador combinado basado en Árboles de Decisión fue primeramente desarrollado también por Breiman [19], inspirado por el trabajo previo de Amit, and Geman [14]. En realidad, este nuevo tipo de ensemble figura como una extensión de la idea que el mismo Leo Breiman tuvo para crear el algoritmo de Bagging. Esto último justifica, por ejemplo, que para la construcción de las muestras con la que realizar la construcción de cada árbol se tome como método el muestreo con reemplazamiento comentado en la sección anterior. Además, este planteamiento surgió con el propósito de dar competencia a los algoritmos Boosting. Como en los anteriores casos, se puede emplear para problemas de regresión y clasificación, aunque debido al propósito de este trabajo se estudiará únicamente esta última con la presencia de variables categóricas.

Desde un punto de vista computacional, este algoritmo surge para proporcionar un clasificador más rápido y eficaz, el cual dependa de pocos parámetros y pueda afrontar problemas de grandes dimensiones. Atendiendo a otro, como es el estadístico, Random Forest resulta interesante pues añade nuevas funcionalidades que otros algoritmos no aportan, tales como medidas para la importancia de cada variable, distintos pesos para el atributo clase, visualizaciones de distinta índole, detección de outliers, imputación de valores perdidos, etc.

[1]Fuera de una definición formal, se puede concebir al sesgo como al error asociado a la falta de similitud entre la distribución real de los datos y la existente dentro del conjunto de entrenamiento. Por su parte, la varianza corresponde a la dispersión que existe dentro de los datos de entrenamiento con respecto a los valores medios de cada atributo.

Algorithm 6 Random Forests

Input: $\mathcal{D} = \{x_i, y_i\}$ (training data), where $x_i = (x_{i,1}, ..., x_{i,p})^T$; $\forall i = 1, ..., N$

Output: Prediction of the response variable at x using the jth tree: $\hat{h}_j(x)$; $\forall j = 1, ..., J$

1: Take a bootstrap sample $\mathcal{D}_j$ of size N from $\mathcal{D}$.

2: Using the bootstrap sample $\mathcal{D}_j$ as the training data, fit a tree using binary recursive partitioning.

   a) Start with all observations in a single node.

   b) Repeat the following steps recursively for each unsplit node until the stopping criterion is met:

      I) Select m predictors at random from the p available predictors.

      II) Find the best binary split among all binary splits on the m predictors from step I).

      III) Split the node into two descendant nodes using the split from Step II).

3: To make a prediction at a new point x: $\hat{f}(x) = arg\, max_y \sum_{j=1}^{J} I\left(\hat{h}_j(x) = y\right)$

En el anterior pseudocódigo, expuesto originariamente en Zhang and Ma [64], se muestra el proceso que llevan a cabo algoritmos de este tipo. Además del ya comentado método para obtener cada una de las muestras, otro punto de interés de este ensemble es el modo de seleccionar las variables para ramificar el árbol. Aleatoriamente en cada nodo se escoge un subconjunto de menor tamaño que el del conjunto total de atributos. Por lo tanto, solamente m de las p variables son consideradas al dividir (siendo $p > m$). Toda esta conjunción aleatoria, tanto a nivel muestral como a nivel de atributos seleccionados para la construcción de estos árboles hacen que este ensemble genere una cantidad considerable de rendimientos de clasificación diferentes, en función de los valores tomados. Esta variedad permite confluir en una decisión final robusta y que mejora con respecto a muchos clasificadores tradicionales.

Por otra parte, uno de los parámetros que se requiere para el funcionamiento de este algoritmo es el del número de árboles a construir. Este valor puede fluctuar, dependiendo el problema que se intenta resolver, desde unos pocos hasta una cantidad considerable, por ejemplo, medio millar. Al finalizar cada una de dichas construcciones, los resultados de los árboles resultantes son combinados por medio del método basado en el voto ponderado, es decir, si 80 de los 100 árboles entrenados pronostican que mañana lloverá, la decisión final del algoritmo combinado será esta misma. El número de votos en favor de una proposición proporciona a su vez un grado de seguridad de tal predicción, es decir, si en el ejemplo anterior en lugar de 80 los árboles que afirman que al día siguiente habrá precipitaciones fueran 51, el grado de fiabilidad sería claramente menor.

Además, Random Forests posee otra gran ventaja en virtud de su estructura de funcionamiento, ya que el considerar un subconjunto de variables en lugar del total disponible permite alcanzar soluciones de una manera más eficiente en términos computacionales. El tiempo de cálculo se ve significativamente reducido debido al modo de ramificar y construir cada árbol. Cuando se crea cada árbol, el algoritmo generalmente no realiza ninguna poda y esto puede dar lugar a *overfitting*. Sin embargo, un modelo Random Forest con árboles que contienen sobreajuste puede proporcionar un gran clasificador que rinde igual de bien o mejor con nuevos datos, a diferencia de lo que ocurre con Árboles de Decisión tradicionales, conclusiones que se han podido obtener con mayor profundidad en el apartado 2.2.4.

Capítulo 5

Aplicaciones de árboles simples basados en probabilidades imprecisas sobre bases de datos con ruido

Todas las propuestas que se han ilustrado en este trabajo hasta el momento han sido usadas con frecuencia dentro de los problemas de clasificación. Los distintos Árboles de Decisión que han ido apareciendo a lo largo de todo este tiempo han servido para evolucionar y mejorar las prestaciones de las que dispone un científico de datos a la hora de enfrentarse a un problema de estas características. El método usual para conseguir dichos progresos se basa en la comparación estadística, en la que se confrontan, por un lado, una propuesta contrastada, en este caso un algoritmo, con otra en fase experimental y que se perfila como sustituta de la primera.

En este sentido, existe para la temática que se está tratando en este trabajo un ejemplo de este tipo de puestas en práctica, más concretamente la ya mencionada aportación de Mantas and Abellán [43], donde se evalúan los rendimientos de los algoritmos C4.5, CDT y Credal-C4.5 sobre bases de datos a las cuales se les añaden ruido arbitrariamente.

El comparar métodos que tienen en cuenta la falta de precisión en los datos disponibles con los que no permite contrastar si existen diferencias significativas entre ellos. C4.5 tan solo considera dicha imprecisión en el método de poda, mientras que Credal-C4.5 asume esta misma hipótesis antes y después de clasificar. Por lo tanto, este último algoritmo es especialmente adecuado cuando se clasifica con conjuntos de datos con ruido. En el trabajo anteriormente citado se demuestra la existencia de diferencias relevantes en el rendimiento de ambos métodos. Primeramente se compara un método previo que también asume la imprecisión de los datos junto con Credal-C4.5, y se muestra que este último mejora en todas las comparativas (con o sin ruido presente). En un segundo estudio experimental se compara el algoritmo C4.5 clásico con el mismo Credal-C4.5 y además una versión mejorada del conocido ID3, llamado MID3.

Sin ruido, los tres métodos anteriores tienen un rendimiento similar, con la única diferencia de que Credal-C4.5 crea árboles de tamaño notablemente menor que el resto. Sin embargo, conforme aumenta el porcentaje de imprecisión es este último quien claramente obtiene mejores resultados y también el que crea árboles con un número de nodos menor que el del resto de propuestas.

Hay que especificar que los conjuntos de datos generados para resolver problemas reales no son perfectos, es decir, normalmente tienden a presentar algún nivel de ruido. Con la aplicación de este trabajo se suponía que podría ser bastante interesante utilizar el algoritmo Credal-C4.5 en datasets reales, para analizar los resultados y extraer conocimiento del problema en cuestión gracias a un Árbol Credal. Estos estudios se plantearon como futuras líneas de trabajo, y, junto con otras motivaciones llevan a plantear el realizar este Trabajo Fin de Máster, donde se plantearán combinaciones de algoritmos que, de la experiencia previa por trabajos como los aportados, se saben que funcionan bien con ruido, de cara a comprobar si existen mejoras con respecto a los resultados de dichos trabajos previos.

A continuación se procede a exponer en qué consisten las pruebas más conocidas para realizar la comparación a priori y a posteriori de algoritmos de clasificación. Posteriormente, se comentarán los resultados de estas en la experimentación realizada en Mantas and Abellán [43].

5.1. Test de Friedman

Corresponde a un análisis estadístico de la varianza no paramétrico atribuida a Friedman [31] basado en rangos. Es, por tanto, la versión no paramétrica de la prueba ANOVA de una vía con medidas repetidas y se usa para identificar diferencias entre tratamientos a través de múltiples pruebas. Esto significa, por tanto, que este test está exento de considerar cualquier restricción en cuanto a la distribución de los datos, la igualdad en varianzas u otra que pudiera plantearse. Ahora bien, dicha libertad también implica una pérdida de potencia con respecto al test paramétrico. La hipótesis nula para la comparación comentada contempla el escenario en el que las distribuciones de todos los algoritmos son idénticos. Decir esto último equivale a argumentar que todos los algoritmos rinden de igual manera. El caso contrario es el que constituye la hipótesis alternativa.

Una vez se realizan todos los cálculos, cada algoritmo posee un valor por separado para cada dataset, donde el mejor algoritmo es asignado con el valor 1, el segundo con el 2, etc. El estadístico de contraste para este test:

$$F = \frac{12D}{n(n+1)} \left[\sum_j R_j^2 - \frac{n(n+1)^2}{4} \right] \tag{5.1}$$

sigue una distribución Chi Cuadrado con $n-1$ grados de libertad, donde n es el número de algoritmos a comparar. D corresponde al número de datasets a comparar. Además, R_j^2 se define como el ranking promedio de los algoritmos empleados:

$$R_j = \frac{1}{D}\sum_i r_i^j, \tag{5.2}$$

donde r_i^j es el ranking asociado al j-ésimo algoritmo de entre los n disponibles. Si se rechaza la hipótesis nula se tiene que hacer uso de una prueba a posteriori que determine las causas de esta decisión.

5.1.1. Caso práctico

Como anteriormente se ha aclarado, este apartado se centra en mostrar, a modo ilustrativo, una investigación real llevada a cabo, tanto en Mantas and Abellán [42] como en Mantas and Abellán [43]. En ellos se contrasta el rendimiento de varios clasificadores con bases de datos que poseen ruido. La tabla A.1 muestra la precisión en media según el nivel de ruido para el primer artículo, mientras que en las tablas A.6 y A.7 se muestran la misma información para el segundo, salvo que en este caso también se distingue entre clasificación con y sin poda. Los algoritmos usados en el primer caso son, por un lado, Árboles de Decisión cuyo criterio para ramificar el árbol está basado en criterios básicos como *Info-Gain* (IGT) e *Info-Gain Ratio* (IGRT). Por otra parte, se encuentran los Árboles de Decisión basados en probabilidades imprecisas: Credal Decision Trees (CDT) y Complete Credal Decision Trees (CCDT). Para el segundo estudio se hacen uso de algoritmos clásicos, como C4.5 y MID3, así como otra propuesta que tiene en cuenta el contexto de imprecisión, tal y como es Credal-C4.5.

Además, en la tabla A.2 se muestran los resultados derivados de la aplicación del test de Friedman para estos algoritmos y conjuntos de datos. En este caso, para todos los niveles de ruido existentes en la variable clase, el algoritmos que mejor rinde es CCDT, ya que obtiene rankings más bajos que el resto de propuestas. En la segunda práctica (ver tabla A.8) C4.5 es el mejor algoritmo con 0 % de ruido y Credal-C4.5 para un 10 % y 30 % del mismo.

5.2. Post-hoc de Nemenyi

Si en el resultado obtenido por el test de Friedman se obtiene un p-valor significativo implica que alguno de los n algoritmos comparados tienen diferente distribución al resto, pero no se sabe cuál o cuales son. Por lo tanto, existe un análisis post-hoc elaborado por Nemenyi [47] para realizar esta tarea.

La prueba de Nemenyi es similar al test de Tukey para un ANOVA, realizándose un total de $C(n,2)$ pruebas (donde $C(n,2)$ corresponde a la combinación de ambas cifras). Este test

exige que cada grupo contenga el mismo tamaño de muestra y los rangos dados por el test previo de Friedman. Se dice que dos algoritmos rinden significativamente de manera dispar si la diferencia entre los rangos promedios correspondiente a tales algoritmos equivale, al menos, al valor que configura la denominada *diferencia crítica (CD)*:

$$CD = q_\alpha \sqrt{\frac{n(n+1)}{6D}} \tag{5.3}$$

donde D es el número de datasets usados en la comparación y q_α un valor tabulado correspondiente al rango estudentizado. El estadístico de contraste usado para la comparación del i-ésimo y j-ésimo clasificador es:

$$z = \frac{(R_i - R_j)}{\sqrt{\frac{n(n+1)}{6D}}} \tag{5.4}$$

El valor de z se usa para encontrar la correspondiente probabilidad en la tabla de la distribución Normal, para, posteriormente ser comparado con el nivel de significación impuesto para determinar la decisión final.

5.2.1. Caso práctico

Al igual que en el test de Friedman, las mismas investigaciones anteriormente citadas aportan los resultados de los análisis post-hoc de Holm (no explicados en este trabajo) y de Nemenyi, respectivamente. Prestando atención, por tanto, a este último análisis, en las gráficas A.9 y A.10 se muestran los resultados de esta última prueba para los mismos niveles de significación (10 %) sobre los mismos datos y algoritmos, pero con distintos niveles de ruido. Para un 10 % de ruido, por ejemplo, existen diferencias significativas entre los algoritmos C4.5 y Credal-C4.5, así como entre este último y MID3, mientras que con un 30 % también existen tales diferencias para la pareja de algoritmos restante, es decir, C4.5 y MID3.

Capítulo 6

Estudio experimental

Del capítulo cuatro ya se conoce que la combinación de varios Árboles de Decisión proporciona el mejor balance entre rendimiento y simplicidad dentro de esta familia de clasificadores. La idea que surge consiste en generar un conjunto de árboles diferentes y combinarlos con un criterio de voto por mayoría. En otras palabras, cuando aparece una instancia no etiquetada (perteneciente al conjunto de test), cada clasificador realiza una predicción y la instancia es asignada al valor de la clase con el mayor número de votos.

Sin embargo, a la hora de construir dichos ensembles aparecen una diversidad de problemas que pueden ser críticos. Si dichas muestras son uniformes, el método entonces crea modelos basados en árboles similares y combinará, por tanto, las mismas reglas, no teniendo en cuenta otras no señaladas y quizá importantes para el problema de clasificación en cuestión (Breiman [18] y Freund and Schapire [29]). Ahora bien, si el ensemble se genera en base a un conjunto de árboles que contienen decisiones que son diferentes entre sí y exhiben buen rendimiento por separado, dicha combinación será muy robusta y proporcionará una mejor capacidad de predicción.

Una situación similar a la anterior surge cuando se generan muchos árboles para un conjunto de datos. En este caso, el uso de árboles con reglas repetidas provocará otorgar menor importancia a otras reglas menos presentes. Por lo tanto, es necesario estipular el número óptimo de árboles a generar por el algoritmo. Por otra parte, si se quiere optimizar la precisión de alguno de estos métodos, se sabe que, en general, con un gran número de árboles generados, normalmente, se obtienen mejores cifras de clasificaciones correctas, mientras que se obtienen peores registros en cuanto al tiempo de ejecución. Se puede comprobar que el número óptimo de árboles a usar depende del tipo de dataset que se tenga, pero no se sabe cómo calcularlo. El equilibrio óptimo entre precisión y tiempo de ejecución es todavía un problema abierto para algunos de estos métodos.

Existen bastantes métodos diferentes para la constucción de árboles de decisión basados en ensembles, pero Bagging (Breiman [18]), Random Forest (Breiman [19]) y AdaBoost (Freund and Schapire [29]) se mantienen como las versiones más conocidas y usadas. El método de Bagging de Breiman (con bootstrap) es uno de los primeros casos de ensembles con Árboles de Decisión. Como ya se ha comentado en capítulos anteriores, la diversidad de árboles en el método Bagging se obtiene gracias a la realización de réplicas bootstrap del conjunto de datos original (es decir, se configuran distintos conjuntos de entrenamiento aleatoriamente con remplazamiento). Posteriormente, un Árbol de Decisión se construye con los datos de entrenamiento de cada réplica con el uso del método estándar (Breiman, Friedman, Stone, and Olshen [21]). Así, cada árbol puede ser definido por un conjunto de diferentes variables, nodos y hojas. Finalmente, sus predicciones son combinadas mediante voto por mayoría.

Bagging puede ser empleado con diferentes tipos de Árbol de Decisión, aunque no hay ninguno considerado como estándar. Existen estudios donde este método es usado junto con ID3, CART o C4.5. y algunas veces estos algoritmos emplean algún método de poda a posteriori. Por ejemplo, el trabajo citado en numerosas ocasiones de Dietterich [25] emplea el algoritmo C4.5 (con y sin poda) para representar al método Bagging como un ensemble eficaz bajo conjuntos de datos con ruido en la variable clase, pero no hay ninguna sugerencia definitiva sobre el empleo o no de la poda.

6.1. Metodología

Siguiendo toda la estructura teórica que se ha proporcionado en este trabajo, se puede proporcionar un estudio experimental que permita establecer posibles diferencias entre los diferentes métodos estudiados con la presencia o ausencia de ruido. De este modo, se han centrado los esfuerzos en obtener resultados derivados de las siguientes combinaciones de algoritmos:

1. Boosting
    - AdaBoost.M1+C4.5
    - AdaBoost.M1+J48realCompleteIPTree (CredalC4.5)
    - AdaBoost.M1+ImpreciseREPTree (CDT)
2. Bagging
    - Bagging+C4.5
    - Bagging+J48realCompleteIPTree (CredalC4.5)
    - Bagging+ImpreciseREPTree (CDT)

3. Random Forests

Todos estos términos referidos a los distintos algoritmos se han visto en profundidad en los apartados teóricos anteriores. La comparativa de ensembles se ha llevado a cabo para distintos niveles de ruido, en concreto con 0, 10, 20 y 30 %.

6.2. Resultados

Una vez se han realizado todas las pruebas, las diferentes salidas se pueden comprobar en las secciones A para los resultados tabulados y B para las gráficas, respectivamente. En cuanto a las primeras, se puede desglosar del siguiente modo: en primer lugar, figuran cada uno de los resultados con diferente ruido para cada dataset y algoritmo (tablas A.11, A.12, A.13 y A.14). En segundo lugar figuran, también según el nivel de ruido, los resultados asociados a los test de comparación múltiple y por pares de Friedman y Nemenyi, respectivamente (de la tabla A.15 hasta la A.22). Las representaciones gráficas se distribuyen desde la B.4 hasta la B.6.

Los conjuntos de datos empleados en estas pruebas corresponden a un total de 25, extraídos del repositorio de Aprendizaje Automático UCI[1] con diferentes tamaños de muestra, atributos y características intrínsecas (valores perdidos, presencia de ruido, outliers, etc). La tabla A.28 muestra esta información.

Con toda esta información numérica y visual se puede extraer información de interés para dar sentido a este trabajo. En las primeras cuatro tablas, citadas anteriormente, se muestra el promedio de todos los datasets para cada uno de los siete ensembles comparados. Para una nula presencia de ruido, el que parece funcionar mejor es el número dos, asociado a la mezcla entre CredalC4.5 y AdaBoost. Sin embargo, conforme aumenta el valor de ruido presente en la clase, este último no se mantiene como la mejor propuesta, ya que, por ejemplo, para un 30 % de ruido el mejor ensemble en promedio es el que combina Bagging y CDT (ImpreciseREPTree).

Pero las anteriores conjeturas no tienen una validez global para cualquier otro dataset, al margen de los utilizados en este estudio, que se presente. Por lo tanto, es necesario recurrir a pruebas estadísticas, tales como los dos test expuestos y comentados en el anterior capítulo. En cuanto al test de Friedman, donde estadísticamente se busca encontrar diferencias significativas, se puede argumentar que los ensembles que mejor rinden según el nivel de ruido son los mostrados en A.27.

donde EC corresponde al estadístico de contraste asociado al test de hipótesis comentado en la sección 5.1. Atendiendo a estos y a los valores asociados a los p-valores obtenidos se concluye con rechazar la hipótesis nula de tal contraste y, por lo tanto, se tiene que todos los algoritmos no funcionan de igual manera.

[1] `https://archive.ics.uci.edu/ml/index.html`

6.2.1. Efectos del ruido

6.2.1.1. Datos con 0 % de ruido

Para cuando no existe ruido en la variable clase, la mejor combinación surge de emplear como base el ensemble clásico AdaBoost. Sin embargo, cuando el nivel de dicho ruido aumenta, el rendimiento de los ensembles creados en base a este ensemble decrece considerablemente con respecto a sus competidores, esto es Random Forests y Bagging (ver figura B.6). Esto se debe a la forma en la que operan estos algoritmos. En primer lugar, como ya se comentó en anteriores capítulos, AdaBoost tiene en cuenta en sus requisitos tomar conjuntos de datos con variable clase binaria, lo cual, para un problema multi-clase, hace que el rendimiento empeore, pues no está diseñado para afrontarlos. Si además, en cada iteración del algoritmo se ponderan instancias que no se consiguen clasificar bien y que, al existir alto nivel de ruido no artificial (por ejemplo, debido a errores humanos), se sabe que no están correctamente etiquetadas, el algoritmo contínuamente trata de clasificar instancias erróneas y con lo cual, la tasa de acierto decrece considerablemente con respecto a los ensembles basados en Bagging y Random Forests.

6.2.1.2. Datos con 10 % de ruido

Para un 10 % de ruido, los algoritmos que mejor rinden son los basados en Bagging y también por su parte, Random Forest. AdaBoost, como ya se ha comentado, comienza a perder efectividad. El ensemble Bagging tiene la capacidad de resolver con efectividad los problemas asociados a la existencia de ruido, debido a que en su funcionamiento interno logra reducir la varianza de los datos. Dicha disminución de la dispersión puede ser suficiente para conjuntos de datos con este porcentaje de ruido, por lo tanto, el método con mayor capacidad de ajuste y, por ende, sesgo creado, es el que obtiene mejores resultados en las pruebas, en este caso, el tándem Bagging & C4.5 con un 86.63 % de precisión (ver tabla A.12). En definitiva, el algoritmo C4.5 tiene mayor capacidad de profundización o ajuste que los Árboles Credal. Estos últimos sacrifican su capacidad de ajuste, creando menor sesgo, al considerar que se está manipulando datos con imprecisión. Argumentando estadísticamente estos resultados, el test de Friedman atribuye al algoritmo 4 (el mismo Bagging & C4.5) como el mejor de todos los ensembles (ver tabla A.17).

6.2.1.3. Datos con 20 % de ruido

Al aumentar la cifra de ruido a un 20 %, son de nuevo los ensembles basados en Bagging los que mejores resultados obtienen. En este caso, Random Forests comienza a decaer en efectividad. De entre los tres ensembles basados en Bagging, el que mejor registro obtiene es el 6, asociado a la combinación de Bagging & CDT. Además, el restante ensemble creado

con Árboles Credales (Bagging & Credal-C4.5) también obtiene mejor cifra que el propuesto por Bagging & C4.5. El test de Friedman permite evidenciar tales resultados (ver tabla A.19). Tales resultados se pueden argumentar con motivos similares a los comentados en el anterior párrafo, solo que en este caso con mayor evidencia. El hecho de tener presente una cantidad considerable de ruido hace que la reducción de varianza ofrecida en el uso de Bagging no sea suficiente para suprimir toda la cantidad de errores que contienen los datos. Sin embargo, por su parte los algoritmos basados en Árboles Credal como siempre consideran un contexto de imprecisión que comienza a darse realmente en los datos usados y, por tanto, hace que estas propuestas clasifiquen con mayo éxito que otras como por ejemplo C4.5, el cual proporciona un mayor sobreajuste, inconveniente crucial cuando se tiene un conjunto de datos con ruido para trabajar. Por su parte, Random Forests baja su rendimiento con respecto al anterior nivel de ruido debido al incremento de este último. Aunque se base en la idea de obtener muestras con reemplazamiento al igual que Bagging, si se generan un total de cien árboles que contienen un alto nivel de ruido (y por ende, una alta probabilidad de clasificar erróneamente) y que se promedian por voto ponderado, entonces parece lógico que la precisión vaya en descenso.

6.2.1.4. Datos con 30 % de ruido

Por último, con un 30 % de ruido de nuevo son los ensembles basados en Bagging y Árboles Credales los que mejor rinden, en este caso con mayor razón si cabe, debido a los mismos argumentos expuestos en anteriores líneas. Es en este caso, Bagging & Credal-C4.5 el ensemble que mejor rinde en promedio con un 82.04 % de precisión. El test de Friedman refuerza este hecho (ver figura A.21), no existiendo grandes diferencias entre los ensembles basados en Árboles Credales (algoritmos 5 y 6 de la anterior tabla citada).

Ante estas situaciones, se lleva a cabo la prueba post-hoc de Nemenyi para comprobar qué combinaciones difieren entre sí en cada nivel de ruido. Dicho test se ejecuta a un nivel de significación del 5 %. Si se observan las tablas asociadas a dichos test (por ejemplo, para un 0 % se encuentra la A.16) los ensembles que difieren entre sí para cada nivel de ruido son los mostrados desde la tabla A.23 hasta la A.26.

Para finalizar, se pueden comprobar las mismas diferencias existentes que ya se han comprobado entre el rendimiento de los siete ensembles para cada nivel de ruido basados, en primer lugar, en Bagging (figura B.4), posteriormente en AdaBoost (figura B.5) y por último en ambos y Random Forest (figura B.6), prestando atención a las gráficas citadas que muestran su evolución. Esta información ya se ha obtenido numéricamente en el anterior análisis numérico. Así, en concreto, observando la última representación mencionada se aprecia como, cuando no existe ruido, todos los ensembles funcionan de manera similar, pero conforme aumenta el nivel de este en la variable clase las combinaciones basadas en AdaBoost quedan claramente por debajo, en términos de rendimiento, de las fundamentadas por Bagging. Finalmente, la última

propuesta, Random Forest, queda entre los dos tipos de ensembles anteriormente comentados, siendo, por lo tanto, peor que las propuestas basadas en Bagging y algoritmos compuestos por probabilidades imprecisas cuando existe ruido.

6.3. Conclusiones

A modo de cierre de este estudio experimental, este proyecto ha querido cubrir tanto a nivel teórico como práctico conceptos usuales de Machine Learning y Minería de datos, tales como los ensembles, Árboles de Decisión, Conjuntos Credales, etc. En concreto, este trabajo ha pretendido consolidar la teoría de Árboles de Decisión basado en probabilidades imprecisas y su utilidad frente a bases de datos con alto nivel de ruido en la variable clase. Las pruebas realizadas con diferentes conjuntos de datos han permitido concluir cuáles han sido las mejores y las peores propuestas.

Además, se ha profundizado a nivel teórico en conceptos tan actuales como son el ruido en un dataset, el exhaustivo funcionamiento de un Árbol de Decisión, algoritmos basados en Conjuntos Credales, como por ejemplo CredalC4.5, los distintos tipos de clasificación y test estadísticos para realizar análisis profundos, así como los algoritmos clásicos basados en combinaciones de varios clasificadores.

Por último, queda de manifiesto que la mejor propuesta para afrontar un problema de clasificación con Árboles de Decisión donde la variable clase tiene ruido es la aportada por los ensembles basados en Bagging y Conjuntos Credales, tales como la combinación del primero con el algoritmo CDT, la cual se desenvuelve con soltura cuando existen grandes cantidades de errores en el conjunto de datos.

Existen varias posibilidades para ampliar esta investigación y conformarlo como un trabajo futuro. Se puede continuar esta labor probando la efectividad de otros algoritmos basados en probabilidades imprecisas, tales como el CCDT y comprobar si se encuentran mejoras o no. Además, se pueden considerar otros tipos de Árboles de Decisión (CART, Random Forest,...) y crear algoritmos nuevos basados en probabilidades imprecisas para estudiar la misma situación que plantea la anterior propuesta.

Apéndice A

Tablas

A.1. Experimentos de Carlos Mantas y Joaquín Abellán (I)

Algoritmo	Prec. (0 %)	Prec. (5 %)	Prec. (10 %)	Prec. (20 %)	Prec. (30 %)
IGT	78.96	78.00	77.49	76.02	74.76
IGRT	78.97	78.09	77.66	76.38	75.14
CDT	79.56	78.84	78.65	77.51	76.72
CCDT	79.63	78.99	78.66	77.57	76.74

Cuadro A.1: Tasa media de acierto para IGT, IGRT, CDT y CCDT cuando se aplican en conjuntos de datos con una cantidad de ruido del $0\,\%, 5\,\%, 10\,\%, 20\,\%$ y $30\,\%$.

Algoritmo	Rank (0 %)	Rank (5 %)	Rank (10 %)	Rank (20 %)	Rank (30 %)
IGT	2.625	2.9	2.9167	3.0917	2.9667
IGRT	2.775	2.9333	2.8333	2.7917	2.7833
CDT	2.475	2.2917	2.2	2.2333	2.2667
CCDT	2.125	1.875	2.05	1.8833	1.9833

Cuadro A.2: Resultados del test de Friendman para $\alpha = 0.05$ de los algoritmos IGT, IGRT, CDT y CCDT cuando se aplican sobre conjuntos de datos con una cantidad de ruido del $0\,\%, 5\,\%, 10\,\%, 20\,\%$ y $30\,\%$.

i	**Métodos**	**P-valor**
6	IGRT vs. CCDT	0.008333
5	IGT vs. CCDT	0.01
4	CDT vs. CCDT	0.0125
3	IGRT vs. CDT	0.016667
2	IGT vs. IGRT	0.025
1	IGT vs. CDT	0.05

Cuadro A.3: P-valores asociados al test de Holm para un $\alpha = 0.05$ con los métodos IGT, IGRT, CDT y CCDT cuando se aplican sobre conjuntos de datos con $0\,\%$ de ruido. Dicho test rechaza aquellas hipótesis que tienen un p-valor≤ 0.01.

i	**Métodos**	**P-valor**
6	IGRT vs. CCDT	0.008333
5	IGT vs. CCDT	0.01
4	IGRT vs. CDT	0.0125
3	IGT vs. CDT	0.016667
2	CDT vs. CCDT	0.025
1	IGT vs. IGRT	0.05

Cuadro A.4: P-valores asociados al test de Holm para un $\alpha = 0.05$ con los métodos IGT, IGRT, CDT y CCDT cuando se aplican sobre conjuntos de datos con $5\,\%$ de ruido. Dicho test rechaza aquellas hipótesis que tienen un p-valor≤ 0.025.

i	**Métodos**	**P-valor**
6	IGRT vs. CCDT	0.008333
5	IGT vs. CCDT	0.01
4	CDT vs. CCDT	0.0125
3	IGRT vs. CDT	0.016667
2	IGT vs. IGRT	0.025
1	IGT vs. CDT	0.05

Cuadro A.5: P-valores asociados al test de Holm para un $\alpha = 0.05$ con los métodos IGT, IGRT, CDT y CCDT cuando se aplican sobre conjuntos de datos con $10\,\%$, $20\,\%$ y $30\,\%$ de ruido. Dicho test rechaza aquellas hipótesis que tienen un p-valor≤ 0.025 para un $10\,\%$ y p-valor≤ 0.016667 para un $20\,\%$ y $30\,\%$.

A.2. Experimentos de Carlos Mantas y Joaquín Abellán (II)

Algoritmo	Prec. (0 %)	Prec. (10 %)	Prec. (30 %)
C4.5	82.62	80.77	74.14
Credal-C4.5	82.30	81.25	76.58
MID3	82.37	80.29	72.88

Cuadro A.6: Rendimiento promedio de C4.5, Credal-C4.5 y MID3 cuando son aplicados en conjuntos de datos con $0\,\%, 10\,\%$ y $30\,\%$ de ruido.

Algoritmo	Prec. (0 %) BP	Prec. (10 %) BP	Prec. (30 %) BP
C4.5	82.13	77.74	65.86
Credal-C4.5	82.23	80.72	73.21
MID3	81.81	77.06	64.82

Cuadro A.7: Rendimiento promedio de C4.5, Credal-C4.5 y MID3 cuando son aplicados en conjuntos de datos con $0\,\%, 10\,\%$ y $30\,\%$ de ruido y después de realizar la poda (BP) del árbol.

Algoritmo	Ruido (0 %)	Ruido (10 %)	Ruido (30 %)
C4.5	1.90	2.07	2.07
Credal-C4.5	2.08	1.60	1.40
MID3	2.02	2.33	2.53

Cuadro A.8: Resultados del test de Friendman para $\alpha = 0.1$ de los algoritmos C4.5, Credal-C4.5 y MID3 (con poda) cuando se aplican sobre conjuntos de datos con una cantidad de ruido del $0\,\%, 10\,\%$, y $30\,\%$.

i	Algoritmos	P-valores
3	Credal-C4.5 vs. MID3	0.000262
2	C4.5 vs. Credal-C4.5	0.018773
1	C4.5 vs. MID3	0.193601

Cuadro A.9: P-valores asociados al test de Nemenyi para un $\alpha = 0.1$ con los métodos C4.5, Credal-C4.5 y MID3 (con poda) cuando se aplican sobre conjuntos de datos con $10\,\%$ de ruido. Dicho test rechaza aquellas hipótesis que tienen un p-valor≤ 0.033333.

i	**Algoritmos**	**P-valores**
3	Credal-C4.5 vs. MID3	0
2	C4.5 vs. Credal-C4.5	0.000808
1	C4.5 vs. MID3	0.021448

Cuadro A.10: P-valores asociados al test de Nemenyi para un $\alpha = 0.1$ con los métodos C4.5, Credal-C4.5 y MID3 (con poda) cuando se aplican sobre conjuntos de datos con $30\,\%$ de ruido. Dicho test rechaza aquellas hipótesis que tienen un p-valor≤ 0.033333.

A.3. Experimentos asociados al Trabajo Fin de Máster

A.3.1. Clasificación

Dataset	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**
anneal	99.64	99.63 •	99.09 •	98.79 •	98.79 •	98.59 •	99.68 ◦
audiology	84.66	84.29 •	74.22 •	80.75 •	81.94 •	74.35 •	80.36 •
autos	86.92	85.23 •	71.04 •	84.39 •	80.29 •	72.65 •	84.29 •
breast-cancer	66.43	67.26 ◦	68.17 ◦	73.09 ◦	73.83 ◦	72.35 ◦	70.02 ◦
cmc	50.10	50.99 ◦	53.76 ◦	53.12 ◦	53.69 ◦	56.02 ◦	50.69 ◦
horse-colic	81.74	81.74 •	81.27 •	85.21 ◦	85.61 ◦	85.21 ◦	85.59 ◦
german-credit	74.40	74.59 ◦	73.06 •	74.73 ◦	74.53 ◦	75.26 ◦	76.08 ◦
pima-diabetes	73.78	73.67 •	73.88 ◦	76.17 ◦	75.83 ◦	75.92 ◦	76.01 ◦
glass2	88.65	88.77 ◦	82.61 •	81.97 •	81.23 •	80.44 •	87.90 •
hepatitis	84.74	85.90 ◦	79.69 •	81.37 •	82.22 •	81.57 •	83.58 •
hypothyroid	99.70	99.72 ◦	99.53 •	99.61 •	99.58 •	99.55 •	99.51 •
ionosphere	93.93	93.93 ◦	90.86 •	92.54 •	92.17 •	90.77 •	93.48 •
kr-vs-kp	99.60	99.62 ◦	99.31 •	99.44 •	99.46 •	98.92 •	99.27 •
labor	88.90	86.57 •	83.60 •	84.63 •	83.30 •	84.03 •	87.10 •
lymphography	84.59	85.77 ◦	78.79 •	79.69 •	80.02 •	77.51 •	83.42 •
mushroom	100.00	99.96 •	99.97 •	100.00	100.00 •	100.00	100.00
segment	98.58	98.61 ◦	96.38 •	97.64 •	97.59 •	96.74 •	98.16 •
sick	99.06	99.03 •	98.66 •	98.85 •	98.90 •	98.54 •	98.43 •
solar-flare	96.07	96.54 ◦	97.25 ◦	97.84 ◦	97.84 ◦	97.84 ◦	96.91 ◦
sonar	85.25	86.15 ◦	77.84 •	80.40 •	80.21 •	77.57 •	84.63 •
soybean	93.32	93.48 ◦	91.10 •	93.10 •	92.87 •	88.81 •	93.31 •
sponge	93.25	93.84 ◦	92.23 •	92.63 •	92.63 •	92.50 •	95.00 ◦
vote	95.28	95.12 •	95.03 •	96.69 ◦	96.64 ◦	95.52 ◦	96.43 ◦
vowel	96.27	96.41 ◦	81.03 •	92.14 •	91.87 •	87.54 •	96.65 ◦
zoo	96.35	95.65 •	91.99 •	92.50 •	91.92 •	92.61 •	96.33 •
Average	88.45	88.50	85.21	87.49	87.32	86.03	88.51

◦, • Improvement or degradation

Cuadro A.11: Resultados con $0\,\%$ de ruido

Dataset	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**
anneal	91.66	92.43 ○	97.73 ○	98.64 ○	98.60 ○	98.36 ○	96.44 ○
audiology	78.45	78.05 •	71.59 •	81.01 ○	80.61 ○	75.68 •	75.72 •
autos	73.91	75.75 ○	57.53 •	79.99 ○	78.92 ○	67.43 •	77.21 ○
breast-cancer	63.72	64.46 ○	67.25 ○	72.04 ○	72.39 ○	71.44 ○	66.77 ○
cmc	48.88	50.52 ○	52.26 ○	51.56 ○	52.34 ○	54.75 ○	48.51 •
horse-colic	77.48	76.61 •	79.10 ○	85.07 ○	84.67 ○	84.64 ○	83.61 ○
german-credit	71.95	72.44 ○	69.80 •	73.92 ○	73.76 ○	74.66 ○	74.79 ○
pima-diabetes	71.22	72.84 ○	71.56 ○	75.53 ○	75.75 ○	75.84 ○	74.24 ○
glass2	75.73	74.50 •	76.88 ○	78.20 ○	78.02 ○	79.79 ○	81.54 ○
hepatitis	80.46	80.24 •	78.20 •	82.06 ○	82.25 ○	82.64 ○	82.71 ○
hypothyroid	97.20	97.85 ○	99.40 ○	99.50 ○	99.50 ○	99.47 ○	99.24 ○
ionosphere	89.81	89.12 •	85.76 •	91.71 ○	91.40 ○	91.35 ○	92.31 ○
kr-vs-kp	90.17	90.50 ○	95.36 ○	99.17 ○	99.13 ○	98.79 ○	96.57 ○
labor	81.97	84.20 ○	80.90 •	81.67 •	78.57 •	83.43 ○	86.90 ○
lymphography	79.42	79.50 ○	76.13 •	78.63 •	78.42 •	78.10 •	83.09 ○
mushroom	91.76	91.94 ○	97.86 ○	99.99 ○	99.99 ○	99.98 ○	99.68 ○
segment	94.43	95.04 ○	94.56 ○	97.14 ○	97.17 ○	96.49 ○	95.92 ○
sick	94.16	94.71 ○	95.76 ○	98.43 ○	98.45 ○	98.45 ○	98.17 ○
solar-flare	92.29	92.27 •	96.26 ○	97.81 ○	97.81 ○	97.81 ○	93.22 ○
sonar	80.83	81.68 ○	72.76 •	77.60 •	77.59 •	76.99 •	81.61 ○
soybean	87.57	88.59 ○	84.95 •	92.72 ○	93.10 ○	88.45 ○	90.41 ○
sponge	89.45	88.59 •	90.95 ○	92.34 ○	92.30 ○	92.50 ○	92.98 ○
vote	90.73	90.78 ○	94.18 ○	95.91 ○	95.84 ○	95.56 ○	94.11 ○
vowel	91.74	91.86 ○	72.55 •	91.38 •	90.85 •	85.83 •	93.22 ○
zoo	92.58	92.60 ○	87.95 •	93.77 ○	93.37 ○	93.77 ○	92.97 ○
Average	83.10	83.48	81.89	86.63	86.43	85.69	86.08

○, • Improvement or degradation

Cuadro A.12: Resultados con 10 % de ruido

Dataset	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**
anneal	85.12	87.06 ○	96.97 ○	98.04 ○	98.05 ○	98.10 ○	91.16 ○
audiology	71.99	71.43 •	68.43 •	78.37 ○	78.67 ○	72.84 ○	71.28 •
autos	66.50	69.22 ○	51.64 •	73.88 ○	75.85 ○	63.51 •	70.63 ○
breast-cancer	60.61	62.21 ○	63.04 ○	70.95 ○	71.27 ○	69.94 ○	62.02 ○
cmc	47.30	49.24 ○	50.99 ○	50.39 ○	50.87 ○	53.21 ○	46.58 •
horse-colic	70.16	70.54 ○	73.10 ○	83.96 ○	84.01 ○	83.44 ○	80.70 ○
german-credit	68.25	68.20 •	66.29 •	71.90 ○	71.65 ○	73.85 ○	71.80 ○
pima-diabetes	71.37	72.41 ○	66.77 •	74.76 ○	75.09 ○	75.30 ○	71.85 ○
glass2	69.42	70.29 ○	71.11 ○	75.22 ○	75.06 ○	77.62 ○	76.88 ○
hepatitis	74.51	73.91 •	73.95 •	80.63 ○	80.57 ○	81.38 ○	79.69 ○
hypothyroid	93.06	95.22 ○	99.19 ○	99.29 ○	99.39 ○	99.36 ○	98.65 ○
ionosphere	82.68	81.99 •	79.24 •	88.01 ○	89.18 ○	90.41 ○	88.39 ○
kr-vs-kp	80.07	79.91 •	87.88 ○	97.50 ○	97.38 ○	97.99 ○	90.37 ○
labor	74.87	71.30 •	75.20 ○	79.63 ○	76.97 ○	82.33 ○	80.53 ○
lymphography	73.08	72.41 •	74.44 ○	77.49 ○	77.83 ○	77.44 ○	78.08 ○
mushroom	81.16	81.20 ○	90.35 ○	99.84 ○	99.84 ○	99.90 ○	96.76 ○
segment	90.75	91.74 ○	93.65 ○	95.81 ○	96.30 ○	96.28 ○	93.48 ○
sick	87.35	90.42 ○	88.46 ○	97.29 ○	98.04 ○	98.29 ○	96.82 ○
solar-flare	87.82	86.46 •	93.82 ○	97.66 ○	97.56 ○	97.66 ○	87.13 •
sonar	75.06	75.41 ○	68.30 •	74.86 •	75.30 ○	76.22 ○	78.54 ○
soybean	82.48	87.04 ○	82.14 •	91.93 ○	92.59 ○	85.51 ○	84.83 ○
sponge	82.32	83.02 ○	91.39 ○	91.79 ○	92.14 ○	92.50 ○	89.45 ○
vote	84.94	85.68 ○	90.19 ○	95.17 ○	95.24 ○	95.49 ○	90.55 ○
vowel	85.59	85.67 ○	64.86 •	88.76 ○	88.54 ○	84.87 •	88.31 ○
zoo	91.22	91.35 ○	85.08 •	93.99 ○	93.60 ○	91.39 ○	87.83 •
Average	77.51	78.13	77.86	85.08	85.24	84.59	82.09

○, • Improvement or degradation

Cuadro A.13: Resultados con 20 % de ruido

Dataset	(1)	(2)	(3)	(4)	(5)	(6)	(7)
anneal	77.69	81.73 ○	95.58 ○	95.99 ○	96.06 ○	97.54 ○	83.29 ○
audiology	70.58	71.30 ○	60.71 •	77.21 ○	77.04 ○	69.01 •	66.02 •
autos	57.93	59.39 ○	47.76 •	64.91 ○	68.80 ○	57.87 •	61.73 ○
breast-cancer	58.18	57.40 •	60.44 ○	64.31 ○	65.22 ○	64.62 ○	59.10 ○
cmc	44.96	47.02 ○	47.77 ○	47.75 ○	48.51 ○	51.32 ○	43.54 •
horse-colic	65.01	65.09 ○	65.09 ○	80.43 ○	81.67 ○	78.12 ○	74.34 ○
german-credit	62.50	62.81 ○	61.61 •	67.27 ○	67.31 ○	70.65 ○	66.93 ○
pima-diabetes	68.79	69.94 ○	62.44 •	71.09 ○	72.93 ○	71.53 ○	67.04 •
glass2	63.60	61.90 •	62.81 •	69.01 ○	69.37 ○	72.54 ○	67.31 ○
hepatitis	67.60	67.00 •	67.23 •	75.18 ○	77.26 ○	80.33 ○	75.24 ○
hypothyroid	86.28	90.95 ○	98.63 ○	97.43 ○	98.60 ○	99.15 ○	97.31 ○
ionosphere	75.68	77.02 ○	70.72 •	80.06 ○	82.65 ○	84.37 ○	81.01 ○
kr-vs-kp	70.13	69.81 •	77.48 ○	88.91 ○	88.92 ○	95.14 ○	79.88 ○
labor	69.57	67.50 •	69.00 •	77.17 ○	75.27 ○	80.37 ○	74.37 ○
lymphography	65.15	67.95 ○	70.20 ○	75.82 ○	75.68 ○	76.53 ○	72.06 ○
mushroom	70.56	70.51 •	79.41 ○	96.31 ○	96.33 ○	98.30 ○	87.93 ○
segment	85.48	87.59 ○	92.73 ○	92.33 ○	93.87 ○	95.99 ○	90.13 ○
sick	84.79	89.26 ○	77.86 •	92.47 ○	95.57 ○	97.25 ○	91.44 ○
solar-flare	80.11	79.32 •	86.17 ○	95.76 ○	95.02 ○	96.01 ○	78.68 •
sonar	68.97	67.35 •	62.98 •	69.47 ○	71.30 ○	73.22 ○	72.75 ○
soybean	81.31	87.60 ○	76.43 •	90.82 ○	91.01 ○	81.61 ○	79.31 •
sponge	77.59	70.68 •	89.73 ○	89.16 ○	86.73 ○	91.95 ○	81.07 ○
vote	77.65	76.98 •	82.45 ○	91.54 ○	91.65 ○	94.00 ○	83.33 ○
vowel	78.45	78.89 ○	60.67 •	84.00 ○	84.55 ○	83.11 ○	80.88 ○
zoo	86.66	87.04 ○	82.38 •	91.31 ○	91.41 ○	90.53 ○	80.50 •
Average	71.81	72.48	72.33	81.03	81.71	82.04	75.81

○, • Improvement or degradation

Cuadro A.14: Resultados con 30 % de ruido

A.3.2. Tests de Friedman y Nemenyi

A.3.2.1. 0 % Ruido

A.3.2.1.1. Rankings promedios del test de Friedman. Los resultados quedan tabulados y representados en:

Algoritmo	Ranking
AdaBoost.M1+C4.5 (1)	3.08
AdaBoost.M1+CredalC4.5 (2)	3
AdaBoost.M1+CDT (3)	5.8
Bagging+C4.5 (4)	3.7
Bagging+CredalC4.5 (5)	4.08
Bagging+CDT (6)	5.02
Random Forests (7)	3.32

Cuadro A.15: Rankings promedios de los algoritmos

El estadístico de contraste, distribuido según una Chi-Cuadrado con seis grados de libertad, vale 35.815714. Por su parte, el p-valor calculado para este test es: 2.9935787022594695E-6.

A.3.2.1.2. Comparaciones Post hoc. Resultados obtenidos con la técnica post-hoc de Nemenyi para $\alpha = 0.05$.

i	**Algoritmos**	$z = (R_0 - R_i)/SE$	p
21	Algorithm 2 vs. Algorithm 3	4.582576	0.000005
20	Algorithm 1 vs. Algorithm 3	4.451645	0.000009
19	Algorithm 3 vs. Algorithm 7	4.058853	0.000049
18	Algorithm 3 vs. Algorithm 4	3.436932	0.000588
17	Algorithm 2 vs. Algorithm 6	3.306001	0.000946
16	Algorithm 1 vs. Algorithm 6	3.17507	0.001498
15	Algorithm 3 vs. Algorithm 5	2.815011	0.004878
14	Algorithm 6 vs. Algorithm 7	2.782278	0.005398
13	Algorithm 4 vs. Algorithm 6	2.160357	0.030745
12	Algorithm 2 vs. Algorithm 5	1.767565	0.077134
11	Algorithm 1 vs. Algorithm 5	1.636634	0.101707
10	Algorithm 5 vs. Algorithm 6	1.538436	0.123942
9	Algorithm 3 vs. Algorithm 6	1.276575	0.201752
8	Algorithm 5 vs. Algorithm 7	1.243842	0.213558
7	Algorithm 2 vs. Algorithm 4	1.145644	0.251943
6	Algorithm 1 vs. Algorithm 4	1.014713	0.310243
5	Algorithm 4 vs. Algorithm 7	0.621921	0.533994
4	Algorithm 4 vs. Algorithm 5	0.621921	0.533994
3	Algorithm 2 vs. Algorithm 7	0.523723	0.600471
2	Algorithm 1 vs. Algorithm 7	0.392792	0.694473
1	Algorithm 1 vs. Algorithm 2	0.130931	0.89583

Cuadro A.16: Tabla de p-valores para $\alpha = 0.05$

La técnica de Nemenyi rechaza aquellas hipótesis que tienen un p-valor no ajustado≤ 0.002381.

A.3.2.2. 10 % Ruido

A.3.2.2.1. Rankings promedios del test de Friedman. Los resultados quedan tabulados y representados en:

Algoritmo	Ranking
AdaBoost.M1+C4.5 (1)	5.68
AdaBoost.M1+CredalC4.5 (2)	5.04
AdaBoost.M1+CDT (3)	5.64
Bagging+C4.5 (4)	2.42
Bagging+CredalC4.5 (5)	2.74
Bagging+CDT (6)	3.24
Random Forests (7)	3.24

Cuadro A.17: Rankings promedios de los algoritmos

El estadístico de contraste, distribuido según una Chi-Cuadrado con seis grados de libertad, vale 63.39. Por su parte, el p-valor calculado para este test es: 5.3486215456644004E-11.

A.3.2.2.2. Comparaciones Post hoc. Resultados obtenidos con la técnica post-hoc de Nemenyi para $\alpha = 0.05$.

i	Algoritmos	$z = (R_0 - R_i)/SE$	p
21	Algorithm 1 vs. Algorithm 4	5.335427	0
20	Algorithm 3 vs. Algorithm 4	5.269962	0
19	Algorithm 1 vs. Algorithm 5	4.811704	0.000001
18	Algorithm 3 vs. Algorithm 5	4.746239	0.000002
17	Algorithm 2 vs. Algorithm 4	4.287982	0.000018
16	Algorithm 1 vs. Algorithm 6	3.993387	0.000065
15	Algorithm 1 vs. Algorithm 7	3.993387	0.000065
14	Algorithm 3 vs. Algorithm 6	3.927922	0.000086
13	Algorithm 3 vs. Algorithm 7	3.927922	0.000086
12	Algorithm 2 vs. Algorithm 5	3.764259	0.000167
11	Algorithm 2 vs. Algorithm 6	2.945942	0.00322
10	Algorithm 2 vs. Algorithm 7	2.945942	0.00322
9	Algorithm 4 vs. Algorithm 7	1.34204	0.179583
8	Algorithm 4 vs. Algorithm 6	1.34204	0.179583
7	Algorithm 1 vs. Algorithm 2	1.047446	0.294894
6	Algorithm 2 vs. Algorithm 3	0.981981	0.326109
5	Algorithm 5 vs. Algorithm 7	0.818317	0.413176
4	Algorithm 5 vs. Algorithm 6	0.818317	0.413176
3	Algorithm 4 vs. Algorithm 5	0.523723	0.600471
2	Algorithm 1 vs. Algorithm 3	0.065465	0.947803
1	Algorithm 6 vs. Algorithm 7	0	1

Cuadro A.18: Tabla de p-valores para $\alpha = 0.05$

La técnica de Nemenyi rechaza aquellas hipótesis que tienen un p-valor no ajustado ≤ 0.002381.

A.3.2.3. 20 % Ruido

A.3.2.3.1. Rankings promedios del test de Friedman. Los resultados quedan tabulados y representados en:

Algoritmo	Ranking
AdaBoost.M1+C4.5 (1)	6
AdaBoost.M1+CredalC4.5 (2)	5.48
AdaBoost.M1+CDT (3)	5.44
Bagging+C4.5 (4)	2.64
Bagging+CredalC4.5 (5)	2.22
Bagging+CDT (6)	2.1
Random Forests (7)	4.12

Cuadro A.19: Rankings promedios de los algoritmos

El estadístico de contraste, distribuido según una Chi-Cuadrado con seis grados de libertad, vale 90.57. Por su parte, el p-valor calculado para este test es: 8.096390224920924E-11.

A.3.2.3.2. Comparaciones Post hoc. Resultados obtenidos con la técnica post-hoc de Nemenyi para $\alpha = 0.05$.

i	Algoritmos	$z = (R_0 - R_i)/SE$	p
21	Algorithm 1 vs. Algorithm 6	6.382873	0
20	Algorithm 1 vs. Algorithm 5	6.186477	0
19	Algorithm 2 vs. Algorithm 6	5.531824	0
18	Algorithm 1 vs. Algorithm 4	5.499091	0
17	Algorithm 3 vs. Algorithm 6	5.466358	0
16	Algorithm 2 vs. Algorithm 5	5.335427	0
15	Algorithm 3 vs. Algorithm 5	5.269962	0
14	Algorithm 2 vs. Algorithm 4	4.648041	0.000003
13	Algorithm 3 vs. Algorithm 4	4.582576	0.000005
12	Algorithm 6 vs. Algorithm 7	3.306001	0.000946
11	Algorithm 5 vs. Algorithm 7	3.109605	0.001873
10	Algorithm 1 vs. Algorithm 7	3.076872	0.002092
9	Algorithm 4 vs. Algorithm 7	2.422219	0.015426
8	Algorithm 2 vs. Algorithm 7	2.225822	0.026026
7	Algorithm 3 vs. Algorithm 7	2.160357	0.030745
6	Algorithm 1 vs. Algorithm 3	0.916515	0.359397
5	Algorithm 4 vs. Algorithm 6	0.883782	0.376814
4	Algorithm 1 vs. Algorithm 2	0.85105	0.394742
3	Algorithm 4 vs. Algorithm 5	0.687386	0.491839
2	Algorithm 5 vs. Algorithm 6	0.196396	0.8443
1	Algorithm 2 vs. Algorithm 3	0.065465	0.947803

Cuadro A.20: Tabla de p-valores para $\alpha = 0.05$

La técnica de Nemenyi rechaza aquellas hipótesis que tienen un p-valor no ajustado ≤ 0.002381.

A.3.2.4. 30 % Ruido

A.3.2.4.1. Rankings promedios del test de Friedman. Los resultados quedan tabulados y representados en:

Algoritmo	Ranking
AdaBoost.M1+C4.5 (1)	5.76
AdaBoost.M1+CredalC4.5 (2)	5.62
AdaBoost.M1+CDT (3)	5.38
Bagging+C4.5 (4)	2.84
Bagging+CredalC4.5 (5)	2
Bagging+CDT (6)	1.8
Random Forests (7)	4.6

Cuadro A.21: Rankings promedios de los algoritmos

El estadístico de contraste, distribuido según una Chi-Cuadrado con seis grados de libertad, vale 97.35. Por su parte, el p-valor calculado para este test es: 8.0062734220121E-11.

A.3.2.4.2. Comparaciones Post hoc. Resultados obtenidos con la técnica post-hoc de Nemenyi para $\alpha = 0.05$.

i	Algoritmos	$z = (R_0 - R_i)/SE$	p
21	Algorithm 1 vs. Algorithm 6	6.481071	0
20	Algorithm 2 vs. Algorithm 6	6.251943	0
19	Algorithm 1 vs. Algorithm 5	6.153745	0
18	Algorithm 2 vs. Algorithm 5	5.924616	0
17	Algorithm 3 vs. Algorithm 6	5.85915	0
16	Algorithm 3 vs. Algorithm 5	5.531824	0
15	Algorithm 1 vs. Algorithm 4	4.778972	0.000002
14	Algorithm 6 vs. Algorithm 7	4.582576	0.000005
13	Algorithm 2 vs. Algorithm 4	4.549843	0.000005
12	Algorithm 5 vs. Algorithm 7	4.255249	0.000021
11	Algorithm 3 vs. Algorithm 4	4.157051	0.000032
10	Algorithm 4 vs. Algorithm 7	2.880476	0.003971
9	Algorithm 1 vs. Algorithm 7	1.898496	0.057631
8	Algorithm 4 vs. Algorithm 6	1.7021	0.088737
7	Algorithm 2 vs. Algorithm 7	1.669367	0.095045
6	Algorithm 4 vs. Algorithm 5	1.374773	0.169202
5	Algorithm 3 vs. Algorithm 7	1.276575	0.201752
4	Algorithm 1 vs. Algorithm 3	0.621921	0.533994
3	Algorithm 2 vs. Algorithm 3	0.392792	0.694473
2	Algorithm 5 vs. Algorithm 6	0.327327	0.743421
1	Algorithm 1 vs. Algorithm 2	0.229129	0.818769

Cuadro A.22: Tabla de p-valores para $\alpha = 0.05$

La técnica de Nemenyi rechaza aquellas hipótesis que tienen un p-valor no ajustado $\leq$ 0.002381.

A.3.3. Otras tablas

Combinación (I)	**Combinación (II)**
AdaBoost.M1 & CredalC4.5 y Bagging & CredalC4.5	AdaBoost.M1 & C4.5 y Bagging & C4.5
AdaBoost.M1 & C4.5 y Bagging & CredalC4.5	Bagging & C4.5 y Random Forest
Bagging & CredalC4.5 y Bagging & CDT	Bagging & C4.5 y Bagging & CredalC4.5
AdaBoost.M1 & CDT y Bagging & CDT	AdaBoost.M1 & CredalC4.5 y Random Forest
Bagging & CredalC4.5 y Random Forest	AdaBoost.M1 & C4.5 y Random Forest
AdaBoost.M1 & CredalC4.5 y Bagging & C4.5	AdaBoost.M1 & C4.5 y AdaBoost.M1 & CredalC4.5

Cuadro A.23: Diferencias existentes entre ensembles para un 0 % de ruido

Combinación
Bagging & C4.5 y Random Forest
Bagging & C4.5 y Bagging & CDT
AdaBoost.M1 & C4.5 y AdaBoost.M1 & CredalC4.5
AdaBoost.M1 & CredalC4.5 y AdaBoost.M1 & CDT
Bagging & CredalC4.5 y Random Forest
Bagging & CredalC4.5 y Bagging & CDT
Bagging & C4.5 y Bagging & CredalC4.5
AdaBoost.M1 & C4.5 y AdaBoost.M1 & CDT
Bagging & CDT y Random Forest

Cuadro A.24: Diferencias existentes entre ensembles para un 10 % de ruido

Combinación
AdaBoost.M1 & C4.5 y AdaBoost.M1 & CDT
Bagging & C4.5 y Bagging & CDT
AdaBoost.M1 & C4.5 y AdaBoost.M1 & CredalC4.5
Bagging & C4.5 y Bagging & CredalC4.5
Bagging & CredalC4.5 y Bagging & CDT
AdaBoost.M1 & CredalC4.5 y AdaBoost.M1 & CDT

Cuadro A.25: Diferencias existentes entre ensembles para un 20 % de ruido

Combinación
AdaBoost.M1 & C4.5 y AdaBoost.M1 & CDT
Bagging & C4.5 y Bagging & CredalC4.5
AdaBoost.M1 & C4.5 y AdaBoost.M1 & CredalC4.5
Bagging & C4.5 y Bagging & CredalC4.5
Bagging & CredalC4.5 y Bagging & CDT
AdaBoost.M1 & CDT y Random Forest

Cuadro A.26: Diferencias existentes entre ensembles para un 30 % de ruido

Ruido (%)	Mejor combinación	EC	P-valor
0	AdaBoost.M1 & CredalC4.5	35.82	$2.99e^{-6}$
10	Bagging & C4.5	63.39	$5.35e^{-11}$
20	Bagging & CDT	90.57	$8.09e^{-11}$
30	Bagging & CDT	97.35	$8.01e^{-11}$

Cuadro A.27: Mejores resultados obtenidos

Dataset	Atributos	Instancias	Clases
anneal	39	898	5
audiology	70	226	24
autos	26	205	6
breast-cancer	10	286	2
cmc	10	1473	3
horse-colic	23	368	2
german-credit	21	1000	2
pima-diabetes	9	768	2
glass2	10	163	2
hepatitis	20	155	2
hypothyroid	30	3772	4
ionosphere	35	351	2
kr-vs-kp	37	3196	2
labor	17	57	2
lymphography	148	19	8
mushroom	23	8124	2
segment	20	2310	7
sick	30	3772	2
solar-flare	13	323	2
sonar	208	61	2
soybean	36	683	19
sponge	45	76	3
vote	17	435	2
vowell	12	990	11
zoo	17	101	7

Cuadro A.28: Bases de datos de UCI

Apéndice B

Gráficos

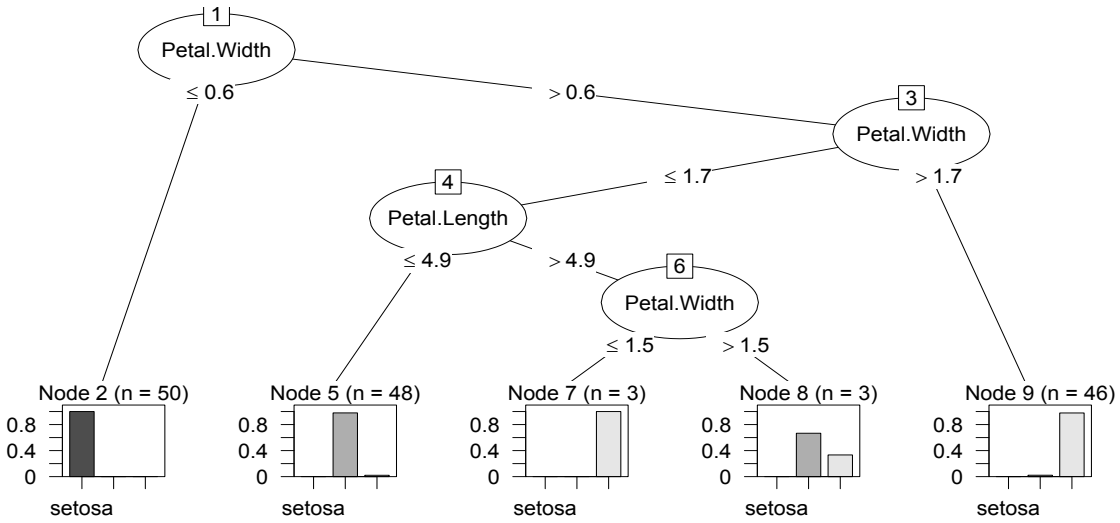

Figura B.1: Árbol de decisión (J48)

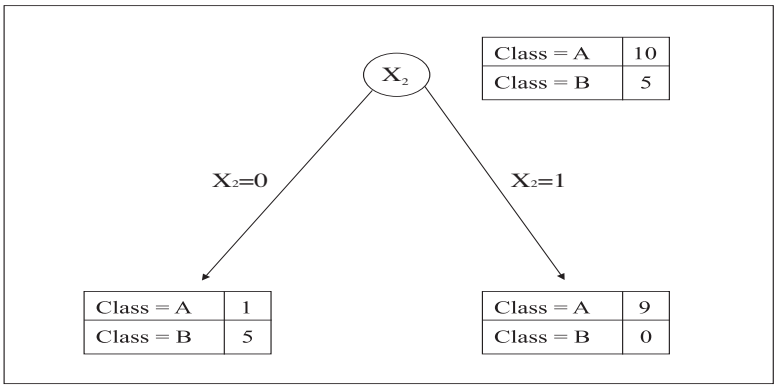

Figura B.2: Ruido

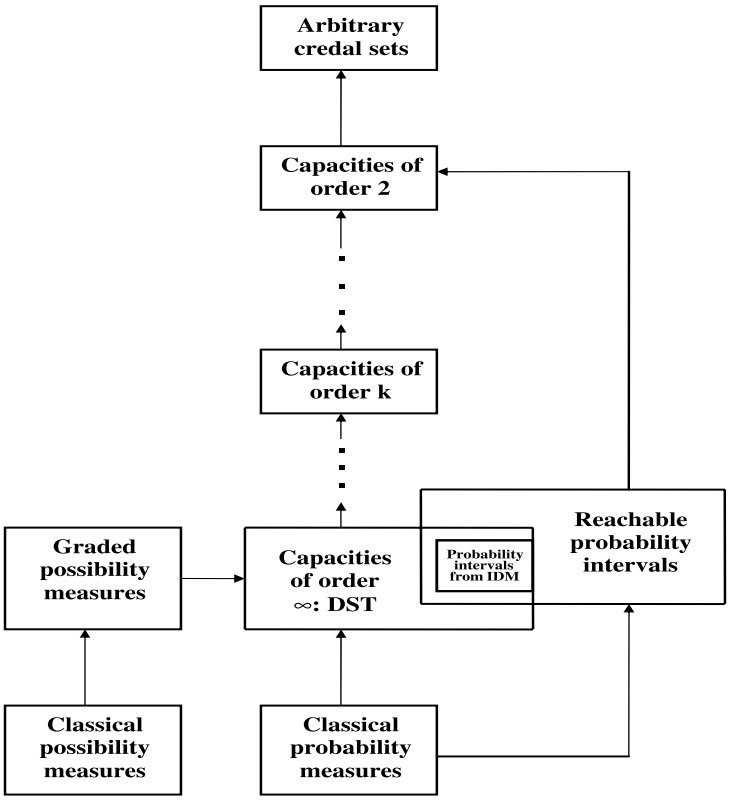

Figura B.3: Resumen de los Conjuntos Credales principales

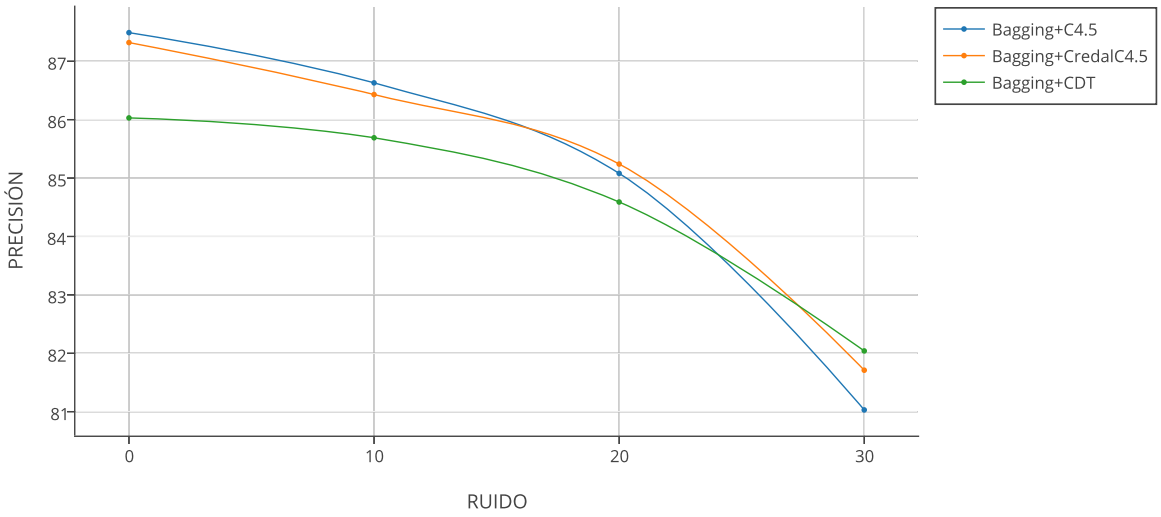

Figura B.4: Rendimiento de distintos ensembles basados en bagging con diferente ruido

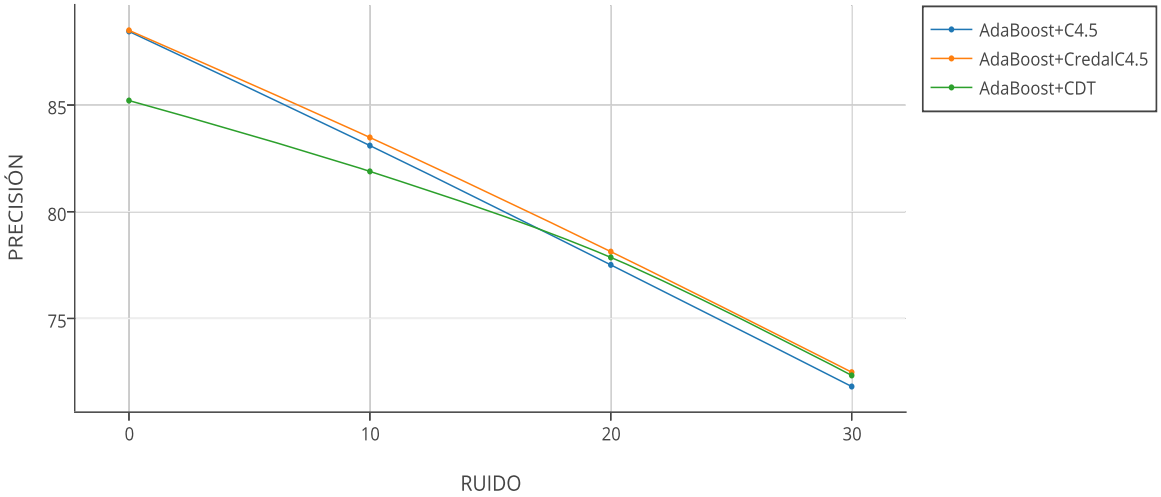

Figura B.5: Rendimiento de distintos ensembles basados en AdaBoost con diferente ruido

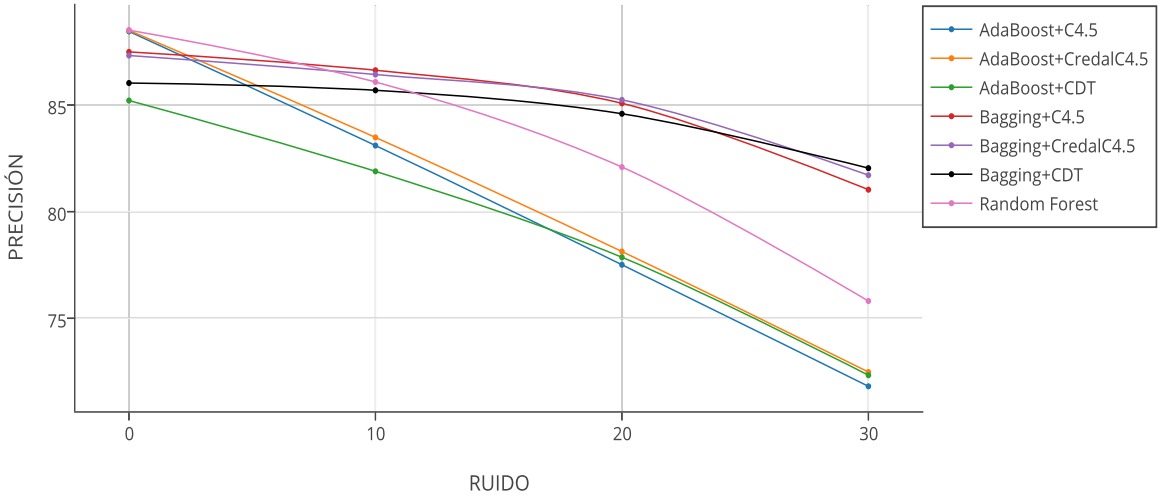

Figura B.6: Rendimiento de distintos ensembles basados en AdaBoost, Bagging y Random Forest con diferente ruido

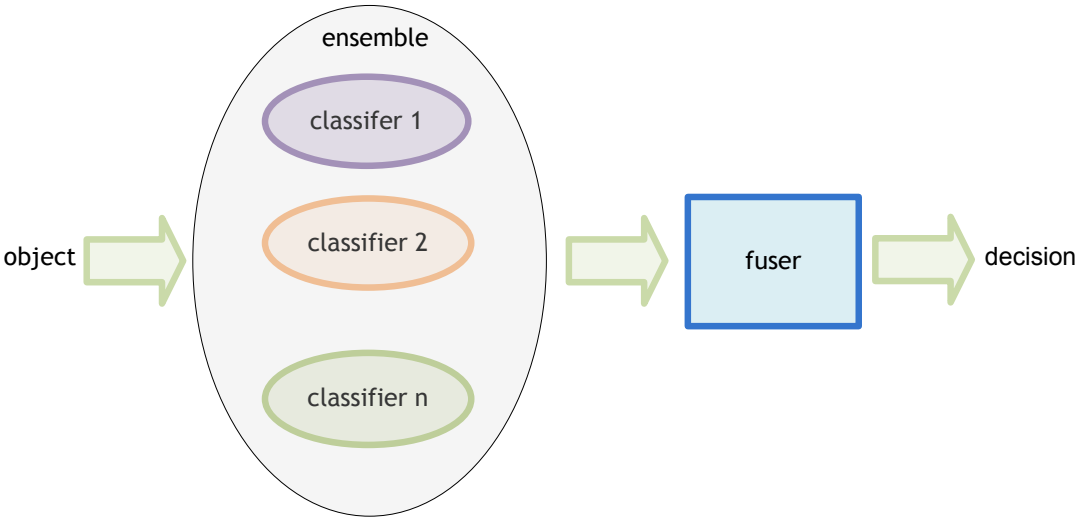

Figura B.7: Jerarquía de un ensemble (I)

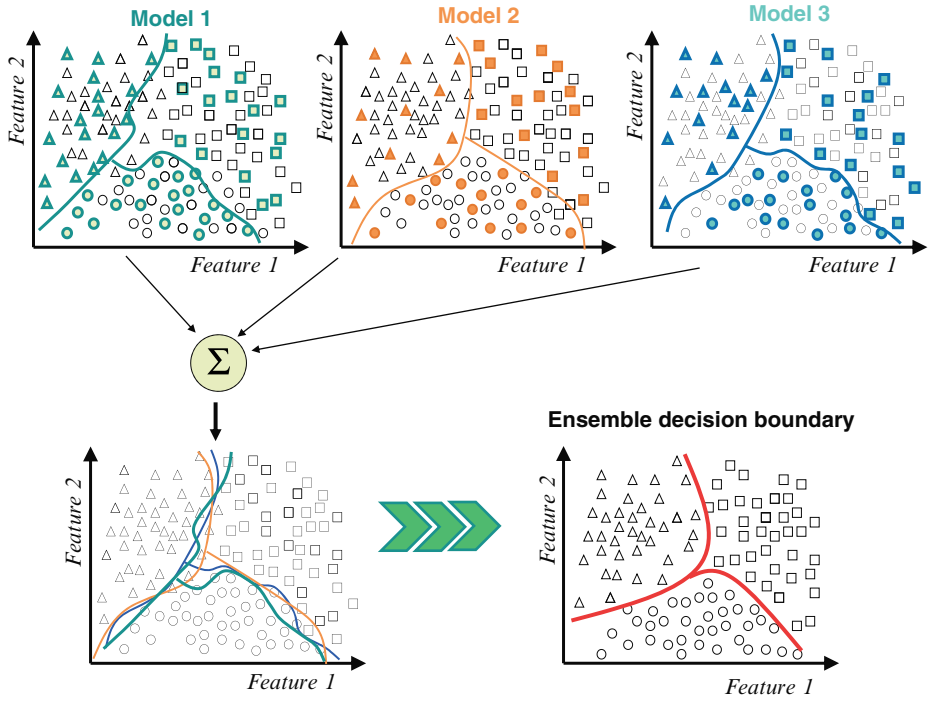

Figura B.8: Jerarquía de un ensemble (II)

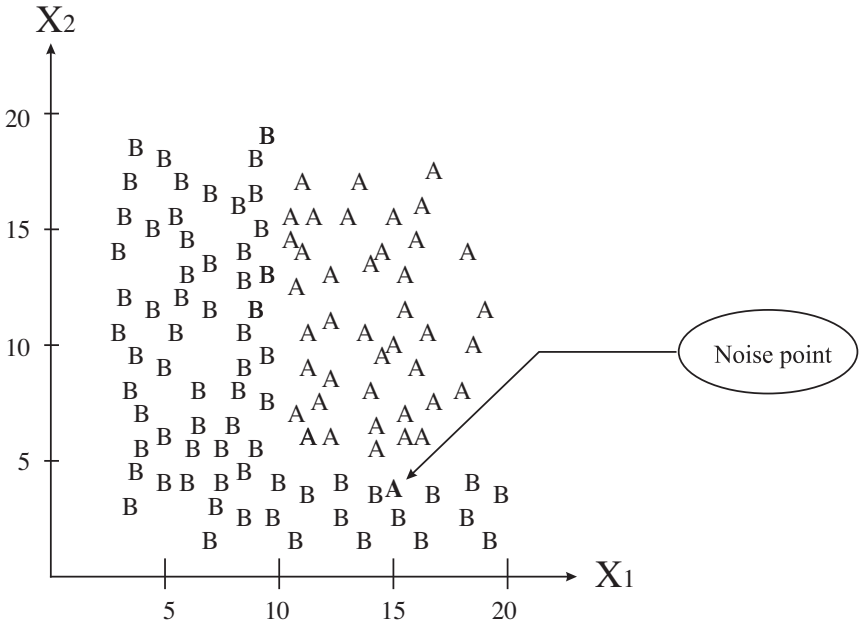

Figura B.9: Ruido

Bibliografía

[1] Joaquín Abellán and Serafín Moral. Upper entropy of credal sets. applications to credal classification. *Int. J. Approx. Reasoning*, 39(2-3):235–255, June 2005. ISSN 0888-613X.

[2] J. Abellán. Uncertainty measures on probability intervals from the imprecise dirichlet model. *International Journal of General Systems*, 35(5):509–528, 2006.

[3] J. Abellán and S. Moral. Difference of entropies as a non-specificity function on credal sets. *International Journal of General Systems*, 201–214, 2005.

[4] J. Abellán and A.R. Masegosa. An ensemble method using credal decision trees. *European Journal of Operational Research*, 205(1):218–226, 2010.

[5] J. Abellán and A.R. Masegosa. A Filter-Wrapper Method to Select Variables for the Naive Bayes Classifier Based on Credal Decision Trees. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 205(1):833–854, 2009.

[6] J. Abellán and S. Moral. A non-specificity measure for convex sets of probability distributions. *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, 8 (3):357–367, 2000.

[7] J. Abellán and S. Moral. Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems*, 18(12):1215–1225, 2003.

[8] J. Abellán and S. Moral. Maximum of entropy for credal sets. *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, 11(5):587–597, 2003.

[9] J. Abellán and S. Moral. An algorithm to compute the upper entropy for order-2 capacities. *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, 14 (2):141–154, 2006.

[10] J. Abellán, G.J. Klir, and S. Moral. Disaggregated total uncertainty measure for credal sets. *International Journal of General Systems*, 35(1):29–44, 2006.

[11] Joaquín Abellán and Andrés R. Masegosa. An experimental study about simple decision trees for bagging ensemble on datasets with classification noise. 5590:446–456, 2009.

[12] Joaquín Abellán and Andrés R. Masegosa. Bagging decision trees on data sets with classification noise. 5956:248–265, 2010.

[13] Charu C. Aggarwal. *Data Mining - The Textbook*. Springer, 2015. ISBN 978-3-319-14141-1.

[14] Amit, Yali and Geman, Donald. Shape Quantization and Recognition with Randomized Trees. *Neural Comput.*, 9(1):1545–1588, 1997. ISSN 0899-7667.

[15] J. E. Cano and S. Moral and J. F. Verdegay-Lopez. Combination of Upper and Lower Probabilities. *Uncertainty in Artificial Intelligence: Proc. of the Seventh Conference*, 61–68, 1991.

[16] Couso, Inés and Moral, Serafín and Walley, Peter. Examples of Independence for Imprecise Probabilities. *ISIPTA*, 121–130, 1999.

[17] V. Barnett and T. Lewis. Outliers in statistical data. *John Wiley & Sons*, 1994.

[18] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[19] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125.

[20] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[21] L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. ISBN 9780412048418.

[22] Gustave Choquet. Theory of capacities. *Actas del Instituto de Fourier*, 5:131–295, 1954.

[23] Luis M. De Campos, Juan F. Huete, and Serafin Moral. Probability intervals: A tool for uncertain reasoning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 02(02):167–196, 1994.

[24] A.P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Studies in Fuzziness and Soft Computing*, 219:57–72, 2008.

[25] Thomas G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000. ISSN 0885-6125.

[26] Drucker, Harris and Schapire, Robert and Simard, Patrice. Boosting Performance in Neural Networks. *International Journal of Pattern Recognition and Artificial Intelligence*.

[27] Dubois, Didier and Prade, Henri. A survey of belief revision and updating rules in various uncertainty models. *International Journal of Intelligent Systems*, 61–100, 1994.

[28] T.L. Fine. Foundations of probability. *Basics Problems in Methodolody and Linguistics*.

[29] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. 1996.

[30] B. Frénay and M. Verleysen. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2014.

[31] Yoav M. Friedman. A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, 11:86–92, 1940.

[32] K. Grabczewski. *Meta-Learning in Decision Tree Induction*. Studies in Computational Intelligence. Springer International Publishing, 2013. ISBN 9783319009605.

[33] Hartley, R. V. L.. *Transmission of Information*. *Bell System Technical Journal*, 7(3): 535–563, 1928.

[34] Ray J. Hickey. Noise modelling and evaluating learning from examples. *Artificial Intelligence*, 82(1–2):157 – 179, 1996. ISSN 0004-3702.

[35] Jaynes, E.T. *On the rationale of maximum-entropy methods*.

[36] G.J. Klir. *Uncertainty and Information: Foundations of Generalized Information Theory*. Wiley, 2005. ISBN 9780471755562.

[37] Klir, George J. and Smith, Richard M. *On Measuring Uncertainty and Uncertainty-Based Information: Recent Developments*. Annals of Mathematics and Artificial Intelligence. 5–33.

[38] Miroslav Kubat. *An Introduction to Machine Learning*. 2015.

[39] S. Link and H. Prade. *Foundations of Information and Knowledge Systems: 6th International Symposium, FoIKS 2010, Sofia, Bulgaria, February 15-19, 2009. Proceedings*. LNCS sublibrary. SL 3, Information systems and applications, incl. Internet/Web, and HCI. Springer, 2010. ISBN 9783642118289.

[40] Yutaka Maeda and Hidetomo Ichihashi. An uncertainty measure with monotonicity under the random set inclusion. *International Journal of General Systems*, 21(4):379–392, 1993.

[41] Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387244352, 9780387244358.

[42] Carlos J. Mantas and Joaquín Abellán. Analysis and extension of decision trees based on imprecise probabilities: Application on noisy data. *Expert Systems with Applications*, 41 (5):2514 – 2525, 2014. ISSN 0957-4174.

[43] Carlos J. Mantas and Joaquín Abellán. Credal-C4.5: Decision tree based on imprecise probabilities to classify noisy data. *Expert Systems with Applications*, 41:4625 – 4637, 2014.

[44] Héctor Quintián Emilio Corchado Marios Polycarpou André C.P.L.F. de Carvalho, Jeng-Shyang Pan Michał Wozniak. *Hybrid Artificial Intelligence Systems*. Springer, 2014.

[45] Robert A Meyers and Peter Kokol. *Computational complexity: theory, techniques, and applications*. Springer, 2012.

[46] S. Moral and L. M. de Campos. Updating Uncertain Information. In Robert A. Meyers, editor, *Uncertainty in Knowledge Bases: Proc. of the 3rd International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'90*, pages 58–67. Springer, 1991.

[47] P. B. Nemenyi. *Distribution-free multiple comparison*. Princenton University. 1963.

[48] Vili Podgorelec and Milan Zorman. Decision tree learning. In Robert A. Meyers, editor, *Encyclopedia of Complexity and Systems Science*, pages 1–28. Springer New York, 2015.

[49] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986. ISSN 0885-6125.

[50] Steven L. Salzberg. C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, 16(3):235–240, 1994. ISSN 0885-6125.

[51] José A. Sáez, Mikel Galar, Julián Luengo, and Francisco Herrera. Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition. *Knowledge and Information Systems*, 38(1):179–206, 2014. ISSN 0219-1377.

[52] Schapire, Robert E. The Strength of Weak Learnability. *Kluwer Academic Publishers*. ISSN 0885-6125.

[53] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976.

[54] Cosma Shalizi. Classification and regression trees. 2009.

[55] C. E. Shannon. Communication in the presence of noise. *Proc. Institute of Radio Engineers*, 37(1):10–21, 1949.

[56] P. Suppes. The measurement of belief (with discussion). *The Royal Statistical Society*, 36:160–191, 1974.

[57] Roman Timofeev. Classification and regression trees(cart) - theory and applications. 2004.

[58] P. Walley. *Statistical reasoning with imprecise probabilities*. Monographs on statistics and applied probability. Chapman and Hall, 1991. ISBN 9780412286605.

[59] P. Walley. Inferences from multinomial data: learning about a bag of marbles. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):3–57, 1996.

[60] Peter Walley. Inferences from multinomial data: Learning about a bag of marbles. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):pp. 3–57, 1996. ISSN 00359246.

[61] Yan Wang. Imprecise probabilities based on generalised intervals for system reliability assessment. *International Journal of Reliability and Safety*, pages 319–342, 2010.

[62] K. Weichselberger. The theory of interval-probability as a unifying concept for uncertainty. *International Journal of Approximate Reasoning*, 24(2–3):149 – 170, 2000. ISSN 0888-613X.

[63] Yoav Freund and Robert E. Schapire. A short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771-780, September, 1999.

[64] Cha Zhang and Yunqian Ma. Ensemble Machine Learning - Methods and Applications. ISBN 978-1-4419-9325-0.

[65] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210, 2004. ISSN 0269-2821.