Minería de Datos: Aprendizaje no Supervisado y Detección de Anomalías
Trabajo teórico de la asignatura

Cristian González Guerrero

7 de septiembre de 2017

Índice

1. Introducción

Este es el trabajo teórico de la asignatura *Minería de Datos: Aprendizaje no Supervisado y Detección de Anomalías*, impartida en el *Máster en Ciencia de Datos e Ingeniería de Computadores* (DATCOM). En el mismo, se desarrollan los contenidos de la asignatura, tomando como hilo conductor las diapositivas de la misma y completando con la bibliografía consultada. Este trabajo, realizado de forma original por Cristian González Guerrero, se presenta para su valoración en la evaluación extraordinara de Septiembre de 2017.

2. Clustering

2.1. Introducción

El clustering o agrupamiento es el proceso de clasificar en grupos un conjunto de items sin tener una información previa de su estructura. Se trata de un problema parecido al de clasificación, en cuanto a que se trata de clasificar los datos de entrata en grupos. No obstante, el clustering presenta una gran diferencia con respecto a la clasificación, y es que no se dispone de las clases, sino que se trata de buscar relaciones entre los datos y crear grupos de elementos según su parecido.

Para conseguirlo, se buscarán grupos de elementos que estén cerca entre sí, atendiendo a sus atributos y a alguna métrica que habrá que definir. Los grupos (*clusters*) obtenidos serán etiquetados, con el objetivo de obtener algún tipo de información útil a partir de estas etiquetas.

2.1.1. Clasificación de las técnicas de agrupamiento

Las distintas técnicas de agrupamiento pueden clasificarse en particionales y jerárquicas.

Agrupamiento Particional grupos disjuntos que cubren todo el conjunto de items

Modelos relacionales cada grupo debe verificar

1. los items del mismo grupo deben estar próximos entre sí
2. los distintos grupos deben estar a la mayor distancia posible

ejemplo: k-medias

Modelos de grafos

Mezcla de densidades un grupo es una región del espacio que

1. presenta una la densidad de items muy alta
2. está rodeada de una región de baja densidad

ejemplo: máxima esperanza

Búsqueda de modas

Agrupamiento Jerárquico jerarquía de agrupamientos particionales anidados

Enlace simple (single-link clustering)

Enlace ponderado

Enlace completo (complete-link clustering)

Otras formas de clasificar las distintas técnicas de agrupamiento incluyen la distinción entre agrupamientos *exclusivos*, es decir, aquellos que no permiten

solapamiento, forzando que cada item pertenezca a un único grupo, y agrupamientos *no exclusivos*, que son los que permiten solapamiento. Dentro de esta última distinción, deben mencionarse las técnicas de agrupamiento difuso, en las cuáles los items pertenecen a los distintos grupos con un nivel de pertenencia determinado.

Finalmente, los algoritmos de agrupamiento pueden ser *aglomerativos* o *divisivos*. En el caso de los algoritmos aglomerativos, se parte de una situación inicial en la que cada item es considerado un grupo, y en cada iteración se construyen grupos cada vez más grandes fusionando los grupos ya existentes, hasta tener un único cluster. Se trata por tanto de una visión Bottom-Up. Los algoritmos divisivos, en cambio, comienzan con un único cluster, consideran todas las posibles formas de dividir este cluster en dos y eligen la mejor división, operando de forma recursiva en ambas divisiones. Tienen por tanto una visión Top-Down del problema.

2.2. Proximidad, distancia y semejanzas

A la hora de plantear un problema de agrupamiento, se suele partir de un conjuto de datos (*dataset*) compuesto por una serie de documentos (*items*) de los que se tienen una serie de medidas (*variables*). Estas medidas o variables pueden ser de distintos tipos:

Cuantitativas si toman valores numéricos

- **Continuas** si los valores son continuos (por ejemplo, el peso de una persona)
- **Discretas** si los valores son discretos (por ejemplo, el número de hijos)

Cualitativas si no toman valores numéricos

- **Valores nominales (no ordenados)** si toman un valor de un conjunto no ordenado (por ejemplo, el color de pelo)
- **Valores ordinales** si toman un valor de un conjunto ordenado (por ejemplo, el grado de satisfacción)
  - **Variables binarias** son un subtipo de valor ordinal, que indica la presencia o la ausencia de una característica

En general, todos los algoritmos de agrupamiento parten de una *matriz de proximidad*, que se construye a partir de los items del conjunto de datos de trabajo (aunque en algunos problemas puede venir dada). Dicha matriz representa la proximidad (o la distancia) entre los distintos items del dataset.

Formalmente, la matriz de proximidad es una matriz $n \times n$, siendo n el número de items del dataset, donde la casilla ik representa la proximidad del elemento i al elemento k. Esta proximidad debe medirse con un *índice de proximidad* apropiado al tipo de datos del problema a resolver.

2.2.1. Índices de proximidad: distancias y semejanzas

Para resolver el problema de construir una matriz de proximidad a partir del conjunto de items I del dataset, usaremos un índice de proximidad $d : I \times I \to \mathbb{R}$ que deberá verificar:

1.

   a) Si d mide distancia, la distancia de un item consigo mismo es nula

$$d(i,i) = 0 \, \forall i \in I$$

   b) Si d mide similaridad, la similaridad de un item consigo mismo es mayor que la similaridad de este mismo item con cualquier otro

$$d(i,i) \geq \max_{k \in I} d(i,k) \; \forall i \in I$$

2. El índice de proximidad es una función simétrica

$$d(i,k) = d(k,i) \; \forall i,k \in I$$

3. El índice de proximidad es una medida no negativa

$$d(i,k) \geq 0$$

Estos índices de proximidad no son más que una generalización de otras medidas más conocidas, como las *funciones de distancia* o las *de similaridad*.

2.2.2. Funciones de distancia

Miden la distancia (o, equivalentemente, disimilaridad) de los items de un conjunto.

Además de las propiedades $1a$, 2 y 3, las funciones de distancia deben cumplir la desigualdad triangular

$$d(i,k) \leq d(i,l) + d(l,k) \; \forall i,k,l \in I$$

Algunas funciones de distancia usualmente utilizadas son las presentadas en table 1.

En general, las funciones de distancia son usadas cuando el set de datos presenta variables continuas o valores enteros. En algunos casos, las medidas de distancia también pueden ser útiles en el caso de variables ordinales.

2.2.3. Funciones de similaridad o semajanza

Miden el parecido o semejanza de los items de un conjunto.

Además de las propiedades $1b$, 2 y 3, las funciones de semejanza deben cumplir

$$d(i,i) = 1 \, \forall i \in I$$

Función de distancia	Notas
Distancia euclídea (norma-ℓ_2)	Es conveniente cuando pueden encontrarse grupos compactos y aislados. Conviene tener los datos normalizados.
Distancia Manhattan (norma-ℓ_1)	Conviene tener los datos normalizados.
Norma del supremo (norma-ℓ_∞)	Conviene tener los datos normalizados.
Distancia de Minkwski (norma-ℓ_p)	Es una generalización de las distancias euclídea, Manhattan y la norma del supremo. Es conveniente con variables normalizadas, ya que de lo contrario da mucho peso a los valores extremos.
Distancia de Mahalanobis	Es otra generalización de la distancia euclídea. Es conveniente en variables continuas con correlación entre ellas.

Cuadro 1: Funciones de distancia usuales.

Función de semejanza	Notas
Coeficiente de correlación de Pearson	Indica la dependencia lineal entre dos items, a través de su correlación.
Medida del coseno	Se basa en calcular el coseno del ángulo que forman los items cuando son representados como vectores en un espacio vectorial con métrica euclídea.
Índice de Jaccard	*Sólo para variables binarias.* Es la razón entre las ocurrencias conjuntas y cualquier tipo de ocurrencia de las dos variables.
Índice de acoplamiento simple	*Sólo para variables binarias.* Indica la probabilidad de encontrar los dos items a la vez o de no encontrar ninguno.
Índice de Dice	*Sólo para variables binarias.* Se trata de la media armónica que combina los valores de precisión y exhaustividad.

Cuadro 2: Funciones de distancia usuales.

Algunas funciones de semejanza usualmente utilizadas son las presentadas en 2

Las funciones de similaridad o semejanza son especialmente útiles cuando se trata de evaluar factores binarios, o bien variables nominales (no ordinales), que pueden ser transformadas en factores binarios.

Además de las funciones de distancia y las de semejanza, existen relaciones de semejanza difusas, que son especialmente útiles con variables nominales.

Cabe destacar que la preparación de los datos y la selección de distancias son etapas fundamentales a la hora de enfrentarse a un trabajo de agrupamiento.

2.3. Agrupamientos jerárquicos

El agrupamiento jerárquico (*hierarchical clustering*) produce una estructura jerárquica que es más informativa que el set de clusters desestructurados producido por los agrupamientos particionales. Esta estructura tiene una representación gráfica muy intuitiva denominada *dendrograma*. Además, el agrupamiento jerárquico no requiere especificar el número de clusters. No obstante, estas ventajas tienen como contrapunto una pérdida de eficiencia, traducida en un mayor coste computacional de los algoritmos, que tienen una complejidad de $O(n^2)$ o mayor.

La mayoría de los algoritmos de agrupamiento jerárquico son *aglomerativos*, y pueden clasificarse en función de la medida de proximidad que usan:

Enlace simple (single-link clustering) la proximidad viene dada por la mínima distancia entre nodos de distintos clusters. Esto tiene como consecuencia la minimización del enlace mínimo.

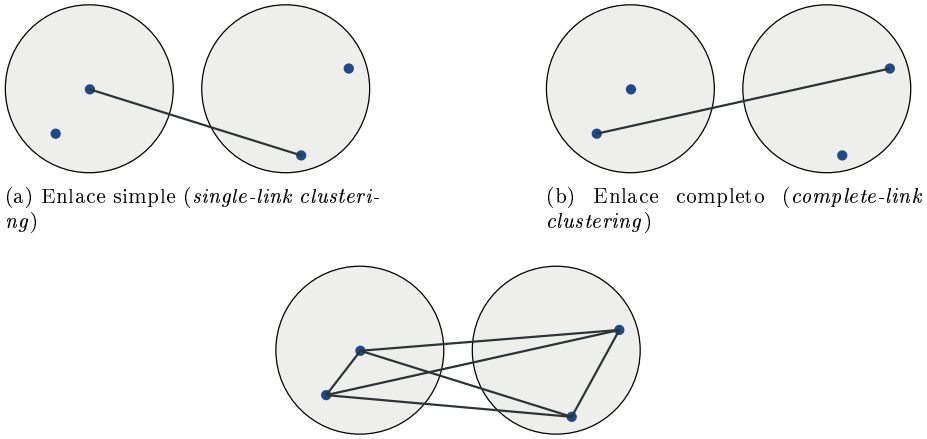
Enlace completo (complete-link clustering) la proximidad viene dada por la máxima distancia entre nodos de distintos clusters. Esto tiene como consecuencia la minimización del enlace máximo (o, equivalentemente, del diámetro del nuevo cluster).

Enlace ponderado (average-link clustering) la proximidad viene dada por la media de las distancias entre los nodos de distintos clusters. Esto tiene como consecuencia la minimización del enlace medio.

En cuanto a algoritmos de agrupamiento jerárquico, cabe destacar el algoritmo de Jhonson [1], que resuelve el problema de encontrar las distancias más cortas entre todos los items. Para ello, modifica la matriz de distancias en cada iteración, a partir de la sucesiva creación de grupos, que se van añadiendo a la misma.

2.4. Agrupamientos particionales

Las técnicas de agrupamiento particional pueden dividirse en tres aproximaciones distintas.



(a) Enlace simple (*single-link clustering*)

(b) Enlace completo (*complete-link clustering*)

(c) Enlace ponderado (*average-link clustering*)

Figura 1: Representación de los distintos tipos de enlace usados en el cálculo de la proximidad entre clusters en las técnicas de agrupamiento jerárquico.

2.4.1. Agrupamientos basados en centroide

Suponen que cada grupo está representado por un prototipo y asignan cada item al grupo cuyo prototipo esté más cercano. Este prototipo puede ser el item más representativo del grupo (*medoide*) o bien, cuando las variables son numéricas, un punto medio entre los items que lo componen (*centroide*). En el último caso, el cálculo del punto medio deberá atenerse a alguna métrica en concreto.

k-medias El algoritmo más representativo de este tipo de técnicas es el de las *k-medias* (*k-means*). Este algoritmo trata de dividir el espacio en los k grupos más representativos, de forma que las distancias de los items de cada grupo a su centroide sea minimizada.

Este algoritmo has sido ampliamente estudiado y presenta numerosas ventajas:

En primier lugar, las particiones forman una estructura conocida como diagrama de Voronoi, que tiene una representación muy intuitiva. También cabe destacar que conceptualmente se parece al también conocido algoritmo k-NN para clasificación.

No obstante, k-medias no está exento de inconvenientes:

El problema de optimización planteado es computacionalmente muy costoso, por lo que se suele atajar con aproximaciones iterativas que generalmente caen en mínimos locales. Esto hace que el algoritmo deba ejecutarse varias veces, usando distintos puntos de partida. Por otro lado, el hecho de tener que elegir de antemano el número de clusters k suele ser visto como un inconveniente.

Cabe destacar la importancia de este método, que ha inspirado numerosas variantes para tratar de solucionar sus puntos débiles. Entre los métodos desarrollados, debe mencionarse el *método de las k-medias difuso*, que aplica el concepto de conjuntos difusos para crear particiones difusas, asociando a cada item un grado de pertenencia a dichas particiones.

2.4.2. Agrupamientos basados en distribuciones

Supone que cada grupo es una distribución de probabilidad de la que se han extraído los items, que pasan a ser una muestra. Es el modelo más ligado a la estadística y tiene una teoría muy bien fundamentada, no obstante, suelen producir sobreajuste, a menos que se impongan restricciones sobre la complejidad del modelo.

Tal vez el método más prominente de este tipo de aproximaciones sea la mezcla Gaussiana, que modela el conjunto de datos con un número de distribuciones Gaussianas de las que se supone que se han extraído las muestras. El algoritmo trata de centrar las distribuciones y ajustar su varianza para aumentar la probabilidad de que las muestras hayan sido tomadas de estas distribuciones.

2.4.3. Agrupamientos basados en densidad

Supone que cada grupo está definido por una zona en la que hay una alta densidad de items y que está rodeado por una zona de baja densidad.

DBSCAN El algoritmo más representativo de este método es DBSCAN. Éste analiza la densidad basándose en el número de vecinos a cada punto dentro de una distancia ε del mismo. De esta forma, un punto puede ser

core point si dentro de la distancia ε hay al menos un número $minPts$ de puntos.

reachable si hay al menos un core point dentro de la distancia ε.

noise si no hay ningún core point dentro del radio ε.

Este comportamiento se ilustra en la figura 2. DBSCAN es especialmente bueno buscando clusters que tienen una densidad parecida, pero se reparten en el espacio siguiendo patrones irregulares.

2.5. Evaluación

La evaluación del agrupamiento es un problema difícil de plantear, pues no es fácil evaluar la bondad en un entorno no supervisado. Algunos de los planteamientos que podemos tratar de evaluar son los siguientes:

- Determinar la tendencia de agrupamiento de un conjunto, es decir, verificar que la estructura creada existe realmente en los datos.

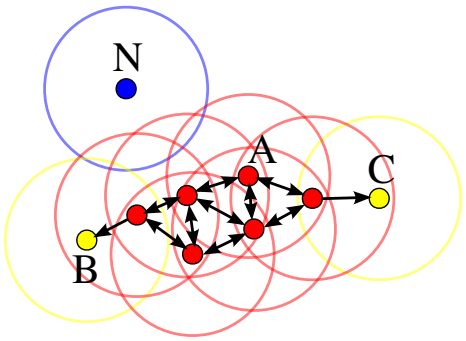


Figura 2: Ilustración del algoritmo DBSCAN. Los puntos rojos son *core points*; los amarillos, *reachable points*; y el azul, *noise* [2].

- Comparar los resultados de un agrupamiento con una partición conocida previamente.
- Comparar los resultados de dos agrupamientos diferentes, para ver cuál es mejor.
- Determinar el número correcto de grupos.

Atendiendo a si la calidad del agrupamiento se medirá con respecto a una partición conocida previamente o no, podemos distinguir entre dos tipos de medidas distintas: no supervisadas y supervisadas.

2.5.1. Medidas no supervisadas

Se utilizan para medir la bondad de un agrupamiento sin tener ninguna información adicional sobre el conjunto de datos. Podemos distinguir dos tipos principales:

Medidas de cohesión miden cómo de compactos son los grupos generados.

Medidas de separación miden cómo de separados están los grupos entre sí.

Este tipo de medidas son conocidas también como *índices internos*, ya que sólo utilizan los datos del problema, sin referenciar datos que no estén en el conjunto de entrenamiento. Como norma general, cuanto más cohesionados y separados estén los grupos obtenidos entre sí, mejor será la calidad del agrupamiento.

2.5.2. Medidas supervisadas

Miden la adecuación del agrupamiento obtenido a una partición ya existente. Este tipo de medidas son conocidas también como *índices externos*, ya que utilizan información que no se encuentra recogida en el conjunto de entrenamiento.

Las medidas supervisadas presentan dos enfoques diferentes:

Medidas orientadas a la clasificación miden cómo se ajustan las particiones obtenidas a una clasificación dada. De este modo, se definen algunas medidas de bondad que son usadas también en clasificación. Así, por ejemplo, podemos hablar de *precission*, *recall*, *F-medida*, etc.

Medidas orientadas a la similaridad se basan en construir matrices de incidencia del agrupamiento

$$IG_{ij} = \begin{cases} 1 & \text{si } i \text{ y } j \text{ pertenecen al mismo cluster} \\ 0 & \text{en caso contrario} \end{cases}$$

y de la clasificación

$$IC_{ij} = \begin{cases} 1 & \text{si } i \text{ y } j \text{ pertenecen a la misma clase} \\ 0 & \text{en caso contrario} \end{cases}$$

para establecer medidas de coincidencia entre ambas.

2.5.3. Medidas relativas

Sirven para comparar diferentes agrupamientos o grupos. Pueden ser supervisadas o no, pero siempre se formulan de forma relativa con el objetivo de comparación.

3. Detección de anomalías

3.1. Introducción y motivación

El análisis de datos es el proceso de inspeccionar, limpiar, transformar y modelar los datos con el fin de descubrir información útil, extraer conclusiones a partir de los datos o ayudar en el proceso de toma de decisiones.

Con frecuencia, el análisis de datos se enfoca desde una perspectiva en la que lo importante es obtener patrones o tendencias que se repiten o caracterizan los datos (*common oriented*). No obstante, también puede resultar útil identificar datos que no se ajustan a la norma o a la tendencia general del conjunto (*uncommon oriented*). Esta es la tarea de la que se encarga la *detección de anomalías*.

Una *anomalía* puede definirse como un punto (o item) del conjunto de datos que es considerablemente diferente a la mayoría de datos, que son considerados *normales*. Cabe destacar que esto implica que en el conjunto de datos existen muchas más *observaciones normales* que *observaciones anómalas*.

La detección de anomalías tiene numerosas aplicaciones en nuestros días, siendo especialmente útiles cuando se trata de *detectar errores en las medidas*, o bien para *detectar datos reales que presentan características extrañas*.

En el primero de los casos, las observaciones anómalas pueden ser eliminadas para evitar que interfieran con el resto de observaciones en nuestro estudio.

ID	$V_1, \ldots, V_n$	Is anomaly?
1	...	No
2	...	No
3	...	No
4	...	**Yes**
5	...	No
6	...	**Yes**

(a) Métodos supervisados

ID	$V_1, \ldots, V_n$	Is anomaly?
1	...	No
2	...	No
3	...	No
N/A		
5	...	No
N/A		

(b) Métodos semisupervisados

ID	$V_1, \ldots, V_n$	Is anomaly?
1	...	N/A
2	...	
3	...	
4	...	
5	...	
6	...	

(c) Métodos no supervisados

Cuadro 3: Training sets de partida para los distintos métodos de detección de anomalías.

Esta aproximación hace de la detección de anomalías una potente herramienta de preprocesamiento, casi obligatoria si en las etapas posteriores del trabajo se utilizan métodos no robustos, como aquellos basados en medias, como la regresión lineal.

En el segundo de los casos, las observaciones deberán ser minuciosamente analizadas, con el fin de detectar situaciones en las que se debe dar una respuesta distinta a la usual (por ejemplo, bloquear un ataque de denegación de servicio).

Los métodos de detección de anomalías pueden dividirse en tres grandes grupos, según la disponibilidad de obaservaciones anómalas identificadas en el dataset. Estos tipos de detección se presentan en los siguientes epígrafes.

3.2. Métodos supervisados

Los métodos supervisados (table 3a) parten de la existencia de anomalías conocidas en el conjunto de datos de entrenamiento (*training set*). Este hecho reduce el problema de la detección a una tarea de clasificación. No obstante, esta clasificación debe ser tratada de un modo especial, debido al hecho de que existen muchas más observaciones normales que anómalas, y que por tanto, un modelo que clasifique todas las observaciones como «normales» tendría una tasa de acierto muy elevada. Este problema en concreto recibe el nombre de clasificación no balanceada (*unbalanced classification*), y puede atajarse mediante las dos

aproximaciones tratadas a continuación.

3.2.1. Métodos basados en instancia (instance-based methods)

Estos métodos transforman el set de datos de entrenamiento para luego aplicar algoritmos de clasificación tradicionales. De esta forma, pretenden transformar el dataset original, que es no balanceado y tiene muchas más observaciones normales que anómalas, en un dataset con un número parecido de observaciones normales y anómalas. Existen dos formas principales de hacer esto.

Submuestreo de las clases mayoritarias Se trata de eliminar observaciones normales, aumentando así la proporción de observaciones anómalas sobre el total.

Para ello, puede eliminarse la observación más cercana a cada observación anómala (Tomek-links). De esta forma se crea una separación clara entre los items de las clases «normales» y los anómalos. También existen otros algoritmos más complejos, como *Condensed Nearest Neighbor (CNN)* o *Neighborhood Cleaning Rule (NCL)*, que están igualmente enfocados a resaltar las diferencias de las observaciones anómalas al posterior algoritmo de clasificación.

Sobremuestreo de las clases minoritarias Se trata de introducir observaciones anómalas artificiales, aumentando así la proporción de éstas.

La técnica más usada es probablemente SMOTE (Synthetic Minority Oversampling Technique) [3]. Esta técnica selecciona al azar una observación anómala y sus k vecinos anómalos más cercanos, para luego generar items sintéticos en la línea que une la observación inicial a sus vecinos.

3.2.2. Métodos basados en algoritmo (algorithm-based methods)

Estos métodos dejan intacto el dataset. A cambio, modifican los algoritmos de clasificación para dar más peso a las clases minoritarias. Existen varias formas de modificar los algoritmos:

Métodos sensitivos al coste Consisten en asignar un coste elevado a los datos de las clases minoritarias, haciéndolos más relevantes y consiguiendo una mejor clasificación.

Bagging Consiste en aplicar el algoritmo de clasificación varias veces, cada vez con una muestra del dataset diferente, que puede incluir elementos duplicados. Estas clasificaciones deberán producir modelos con sobreajuste, que luego serán promediados. De este modo, en cada paso se conseguirá un modelo más efectivo, que habrá contado con la presencia de más observaciones anómalas durante su construcción, mejorando así la calidad de la clasificación.

Boosting Consiste en aplicar el algoritmo de clasificación varias veces, dando cada vez más peso a los elementos que no fueron bien clasificados. De este modo, en cada paso del algoritmo se conseguirá un modelo más efectivo, que habrá sido entrenado dando más importancia a los elementos minoritarios.

Adaptaciones específicas Además, existen algoritmos de clasificación específicos que están adaptados al problema de clasificación no balanceada.

3.3. Métodos semisupervisados

Los métodos semisupervisados (table 3b) parten de la inexistencia de anomalías en el set de datos de entrenamiento. De este modo, el problema de detección de anomalías se basa en identificar futuras observaciones que no se ajustan al comportamiento normal previamente observado. Para ello, es necesario crear un *perfil de comportamiento normal*. Una vez hecho esto, se podrá considerar que una anomalía es una observación que no se ajusta a este perfil.

Los perfiles de comportamiento normal pueden ser modelado a través de diferentes métodos, que se tratan brevemente a continuación.

3.3.1. Métodos basados en clasificación

Se basan en proporcionar un valor numérico que indica cómo de anómala parece una observación, atendiendo a que se ajuste bien o no a las clases aprendidas durante el entrenamiento. Es decir, podremos calificar de outlier a aquellos datos que no parezcan estar en ninguna de las clases establecidas para el problema. Esta tarea puede llevarse a cabo con clasificadores bayesianos, árboles de decisión, reglas de asociación o incluso modelos de Markov [4, 5].

Este tipo de métodos asume que se dispone de un buen modelo para la clasificación, por lo que la etapa de modelado es fundamental en este caso.

3.3.2. Métodos basados en reglas de asociación

Se basan en generar reglas con una elevada confianza confimada, de forma que $A \rightarrow C$ sea una regla con mucha confianza, pero $A \rightarrow \neg C$ tenga muy poca confianza. De esta forma, un caso que no cumpla siga la regla es probablemente anómalo, ya que ésta se trata como un silogismo lógico [**?**].

3.3.3. Métodos basados en kernels

Se basan en el uso de máquinas de soporte vectorial (SVM) para describir la clase normal. Consiste en determinar el hiperplano que separa la clase normal del resto de puntos. Esto se consigue estableciendo una frontera alrededor de los datos que son considerados normales, con un kernel adecuado para el tipo de datos usado. A partir de aquí, los puntos que estén fuera del núcleo definido para la clase normal pueden ser considerados anómalos, o bien darles una puntuación según su lejanía a este núcleo.

3.3.4. Otros métodos

Otra forma de identificar una anomalía es detectar un evento o patrón que no ha sido observado en le pasado. En este sentido, la detección de anomalías puede verse como una comparación con registros históricos, en la que el problema se reduce a elegir atributos de interés en los datos observados y llevar a cabo un conteo del número de observaciones que tiene normalmente cada atributo.

3.4. Métodos no supervisados

Los métodos no supervisados (table 3c) parten de la ausencia de información acerca de la «normalidad» o «irregularidad» de las observaciones. De este modo, el set de datos puede contener anomalías que no han sido etiquetadas, no pudiendo diferenciar a priori si una observación es anómala o no.

En este caso, el problema se traduce a identificar qué observaciones presentan unos atributos muy alejados de lo normal, o bien una combinación extraña de valores. Por este hecho, las observaciones anómalas son también llamadas *valores atípicos* (o *outliers*) en este ámbito.

Al medir la lejanía o distancia de un outlier a la norma, siempre se dará un índice numérico para indicar cómo de anómalo parece el dato. Existen varias aproximaciones para medir esta distancia, que se presentan a continuación.

3.4.1. Aproximaciones gráficas y estadísticas

Las aproximaciones gráficas pasan por llevar a cabo una representación gráfica de las observaciones, pudiendo así identificar visualmente qué puntos están alejados del resto. Se trata de una aproximación útil cuando los registros no tienen demasiados atributos (o, equivalentemente, tienen un número limitado de dimensiones). No obstante, estos métodos son muy subjetivos y requieren de una inspección de los datos que difícilmente puede automatizarse.

Las aproximaciones estadísticas, en cambio, se basan en aplicar contraste de hipótesis para verificar si una observación puede ser calificada como anómala con un determinado nivel de significación (p-value). Estos tests deben ser elegidos en función de la distribución de datos, que se asume conocida, algunos parámetros de la propia distribución y el número esperado de outliers.

Además de estos tests estadísticos, existe una buena solución que se basa en la distancia de Mahalanobis. La distancia de Mahalanobis permite medir la distancia de un punto a una distribución normal multidimensional. Esta distancia tiene en cuenta tanto la media como la varianza de la distribución, por lo que no necesita la normalización de los datos. Además, existe una versión robusta de esta distancia, que tiene en cuenta el determinante de mínima covarianza (minimum covariance determinant) en lugar de la media.

Los estimadores que usan la distancia de Mahalonobis tienen la ventaja de ser paramétricos, por lo que tienen una eficiencia mayor y funcionan muy bien con distribuciones normales. No obstante, estas aproximaciones no son válidas cuando la distribución no es normal, o bien los datos pueden venir de una mezcla de distribuciones normales, como puede observarse en figure 3.

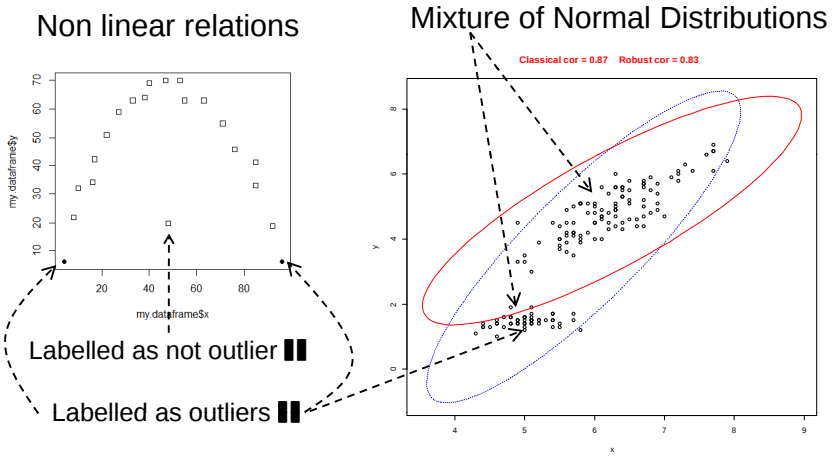


Figura 3: Distancia de Mahalanobis en casos desfavorables.

3.4.2. Aproximaciones basadas en la distancia

Las aproximaciones basadas en la distancia tratan de solventar los problemas de desconocer la distribución estadística de los datos, que no suele ser normal, sino más bien una combinación de varias distribuciones pseudo-normales. Para ello, se deberá calcular la distancia entre los puntos sospechosos de ser anómalos y el resto, comparándolos a su vez con la distancia que tienen la mayoría de los puntos entre sí.

En la práctica, esto suele llevarse a cabo usando una aproximación de los k vecinos más cercanos (*k Nearest Neighbors*).

En el caso más básico, se calcula la distancia de cada punto a su k-ésimo punto más cercano. Esta distancia se toma como índice de lejanía, que servirá para determinar si el punto en cuestión es o no un outlier al comparar la medida con la normal. Esta aproximación es simple, pero tiene numerosos problemas, como el coste computacional, la elección de k, o la incapacidad de detectar anomalías en el caso de tener distribuciones con densidades muy distintas, como se muestra en la figura 4.

Una versión un poco más sofisticada consiste en comparar la inversa de la distancia media de cada punto a sus k vecinos más cercanos (*k-density*) con la media de las inversas de las distancias entre estos k vecinos. El índice de lejanía (*anomaly score*) en este caso recibe el nombre de LOF, y es el resultado de dividir la primera medida entre la segunda (*k-relative density*). Este método, que se ilustra en la figura 5, produce muy buenos resultados en la mayoría de los casos.

3.4.3. Aproximaciones basadas en clustering

Este tipo de aproximaciones se basan en crear un modelo de clusters a través de agrupamiento, para luego tratar de detectar qué datos no pertenecen a ningún

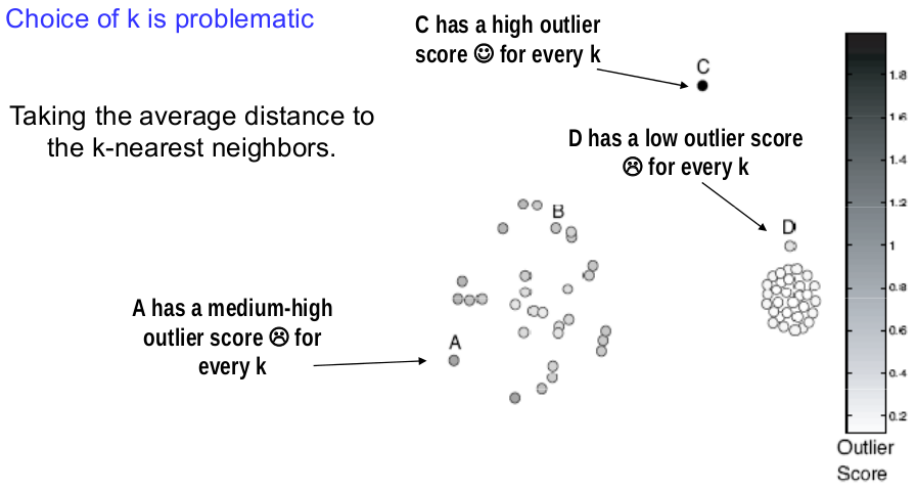


Figura 4: Ejemplo de detección de anomalías por métodos no supervisados basados en distancia. En este caso, el outlier score se ha obtenido usando la técnica de los k vecinos más cercanos.

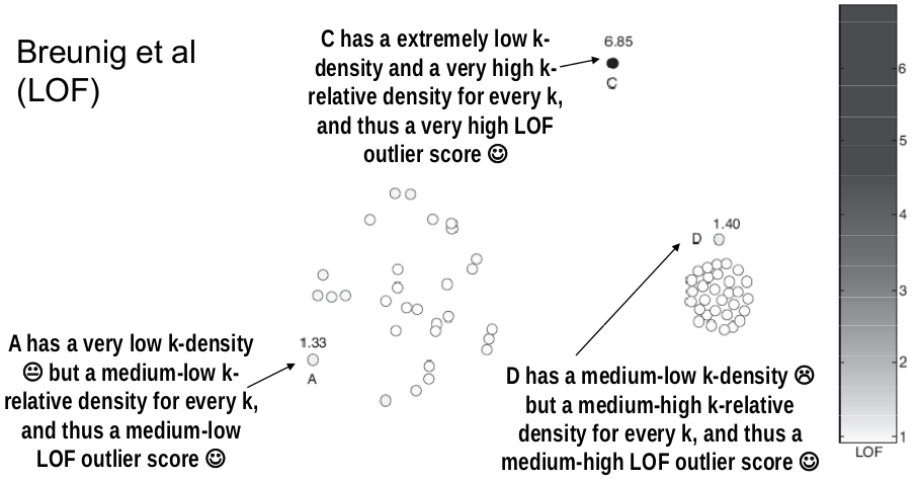


Figura 5: Ejemplo de detección de anomalías por métodos no supervisados basados en distancia. En este caso, se ha calculado la medida LOF para identificar los outliers.

cluster o se encuentran lejanos a los centroides de todos los clusters.

En el primero de los casos, los agrupamientos basados en densidad como DBSCAN u OPTICS pueden clasificar un punto como no perteneciente a ninguno de los clusters aprendidos. De este modo, una anomalía sería un punto que no pertenece a ningún cluster, por presentar unos atributos que lo hacen estar fuera de una zona de alta densidad.

La otra posibilidad es establecer un índice de lejanía, que puede definirse como la distancia al centroide más cercano. En este caso, existen distintas aproximaciones válidas para el cálculo de esta distancia:

1. la distancia euclídea entre el punto y el centroide más cercano, dividida por la distancia mediana entre todos los puntos del cluster en cuestión.

2. la distancia de Mahalanobis entre el punto y la distribución del cluster más cercano.

En cualquiera de estos casos, un modelo de k clusters debe ser elaborado. La elección del parámetro k es problemática. Cabe destacar que k debe ser un número relativamente grande, para tener así un buen número de pequeños clusters y que los outliers sean más facilmente detectados.

3.5. Evaluación

La evaluación de la detección de anomalías parte de las siguientes asunciones:

- La detección produce una salida binaria, del tipo *sí es anomalía* o *no es anomalía.*

- Existe un procedimiento o un experto capaz de determinar con certeza si la predicción de una anomalía es correcta o no. Esto es equivalente a tener un conjunto de test.

Partiendo de este punto, es posible crear una tabla de contingencia entre las predicciones y la realidad, usando las clases *anómalo* y *normal*. Teniendo esta tabla de contingencia es posible establecer numerosas medidas de calidad, como si se tratara de cualquier otro problema de clasificación.

No obstante, habrá que tener en cuenta que se trata de una clasificación no balanceada, puesto que la clase de *anómalo* será mucho menos frecuente que la clase *normal*. Por tanto, será necesario usar medidas adecuadas, evitando el *accuracy* para optar, por ejemplo, por el *recall* o *F-measure*. Una buena opción siempre es la curva ROC, y la medida que se deriva de ella, el área bajo la curva (AUC).

Otra medida útil, que debe ser tenida en cuenta, es el índice de falsa alarma (*false alarm rate*). Esta medida debe darse siempre junto a la precisión (*precision*), para ser capaces de discernir una anomalía real de una falsa alarma.

4. Reglas de asociación

4.1. Definición de regla de asociación

Las reglas de asociación son una técnica muy potente en el ámbido de aprendizaje no supervisado. Esta técnica permite obtener conocimiento interesante a partir de grandes bases de datos. Además, también permiten realizar tareas de clasificación, con modelos que pueden ajustarse tan bien como los árboles de decisión .

En sus orígenes, las reglas de asociación se usaron para extraer relaciones interesantes en compras de supermercado. De ahí que la terminología y los ejemplos clásicos tengan una gran relación con este ámbito.

Formalmente, una regla de asociación (*association rule*) puede definirse como una relación entre un antecedente X (*antecedent*) y un consecuente Y (*consequent*), de la forma

$$X \to Y$$

donde ambas partes están incluidas en un conjunto I de n elementos denominados *items*

$$I = \{i_1, i_2, ..., i_n\}$$

Además, ningún elemento del consecuente puede estar incluido en el antecedente

$$X, Y \subseteq I$$

$$X \cap Y = \emptyset$$

Estas reglas se obtienen a partir de relaciones encontradas en la base de datos de m transacciones D (*transaction database*), que es un conjunto de la forma

$$D = \{t_1, t_2, ..., t_m\}$$

cuyos elementos también son subconjutnos de I

$$t_i \subseteq I \forall i$$

Dentro de esta terminología, hay que distinguir dos tipos de items, y, por tanto, de transacciones:

1. Listado de elementos de longitud variable: no hay un número de atributos determinado, sino que pueden aparecer un número variable de elementos en cada transacción.

2. Número determinado de atributos, formando parejas *(atributo,valor)*.

En el primero de los casos, las transacciones y las reglas tienen una forma como la del siguiente ejemplo:

transacción_1: pan, leche, mantequilla

transacción_2: pan, mermelada, mantequilla, magdalenas

transacción_3: leche, cereales

regla_1: pan $\rightarrow$ mantequilla *y* mermelada

regla_2: leche$\rightarrow$ cereales

En el segundo de los casos, en cambio, los itemsets y las reglas tienen un formato más estricto:

transacción_1: (puesto,administrativo), (salario,alto), (estudios,medios)

transacción_2: (puesto,programador), (salario,medio), (estudios,medios)

transacción_3: (puesto,analista), (salario,alto), (estudios,superiores)

regla_1: (puesto,administrativo) $\rightarrow$ (estudios,medios)

regla_2: (puesto,analista) *y* (salario,alto)$\rightarrow$ (estudios,superiores)

En las siguientes secciones se expondrán los métodos para extraer estas reglas, así como algunas medidas que nos permiten conocer si las reglas encontradas son interesantes o no.

4.2. Medidas clásicas de las reglas de asociación

Existen múltitples medidas que nos permiten tener una idea del interés que pueden suscitar las reglas extraídas de una base de datos. En especial, hay dos medidas que son la base del estudio del interés de una regla. Éstas son el *soporte* y la *confianza*.

4.2.1. Soporte

El soporte (support) es una medida clásica de la importancia que tiene una regla, ya que nos indica la frecuencia relativa con la que ocurre esta regla. Esto es, una regla con un gran soporte es un hecho que ocurre en casi todas las transacciones de la base de datos. De forma equivalente, puede medirse el soporte sobre un itemset como la frecuencia relativa de este itemset en la base de datos. Esta última medida será muy útil a la hora de extraer las reglas.

Formalmente, el soporte del itemset X con respecto al conjunto de transacciones T puede definirse como

$$\text{supp}(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|}$$

esto es, el número de ocurrencias de X dividido entre el número total de transacciones en T.

En el caso de una regla, el soporte se define como el número de transacciones que incluyen tanto al antecedente como al consecuente en la base de datos. Formalmente,

$$\text{supp}(X \to Y) = \text{supp}(X \cup Y)$$

En términos probabilísticos, el soporte de una regla es la probabilidad de que se den el itemset X y el itemset Y en la misma transacción

$$\text{supp}(X \to Y) = P(X \wedge Y)$$

Al tratarse de una frecuencia relativa (o de una probabilidad), esta medida siempre toma valores entre 0 y 1, ambos incluidos,

$$\text{supp}(X) \in [0, 1] \, \forall X \subseteq I$$

4.2.2. Confianza

Se trata de una medida del cumplimiento de una regla. Dicho de otro modo, indica si el consecuente aparece cada vez que se da el antecedente. Formalmente,

$$\text{conf}(X \to Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}$$

En términos probabilísticos, la confianza es la probabilidad de X condicionada a Y, esto es,

$$\text{conf}(X \to Y) = P(X \mid Y) = \frac{P(X \wedge Y)}{P(X)}$$

Cabe mencionar que la dirección en la que se expresa la regla es importante, ya que las reglas $X \to Y$ y $Y \to X$ son distintas y pueden tener una confianza distinta.

$$\text{conf}(X \to Y) \neq \text{conf}(Y \to X)$$

Al igual que el soporte, la confianza toma valores entre 0 y 1,

$$\text{conf}(X \to Y) \in [0, 1] \, \forall X, Y \subseteq I$$

Hay dos casos en los que la confianza es una medida muy interesante. Cuando la confianza de una regla es máxima, $\text{conf}(X \to Y) = 1$, puede leerse que siempre que ocurre X también ocurre Y. Esto podría parecer una implicación lógicas, pero debemos tener cuidado. Para que una regla sea una implicación lógica, también debe cumplirse *modus tollens*, es decir, que $\text{conf}(\neg Y \to \neg X) = 1$.

Por otro lado, cuando la confianza de una regla es nula, $\text{conf}(X \to Y) = 0$, podemos leer que siempre que ocurre X, no ocurre Y.

4.3. Métodos clásicos de extracción de reglas

El objetivo de la extracción de reglas, en su versión más clásica, consiste en obtener un conjunto de reglas que tengan un mínimo soporte $minSup$ y una mínima confianza $minConf$. Estos umbrales deben ser dados por un experto en el problema.

4.3.1. Estrategias de extracción de reglas

En una primera aproximación, podemos plantear dos enfoques para la tarea de extacción de reglas.

El enfoque de fuerza bruta consistiría en crear una lista con todas las reglas de asociación posibles a partir de la base de datos. Una vez generada la lista, podríamos calcular el soporte y la confianza de cada una de ellas, para pasar a eliminar las reglas que no cumplen los umbrales $minSup$ y $minConf$. Este enfoque es demasiado simple y resulta computacionalmente prohibitivo, debido al inmenso número de itemsets que resultan. En concreto, dados d items, existen 2^ditemsets candidatos posibles.

El segundo enfoque, un poco más refinado, consiste en separar en dos el proceso de extracción. En primer lugar, se realiza una lista con todos los itemsets frecuentes, es decir, los que cumplen los umbrales de soporte y de confianza. A continuación, se generan las reglas a partir de la lista de itemsets generada. Aunque este enfoque supone una mejora con respecto al enfoque de fuerza bruta, sigue siendo computacionalmente muy costoso, principalmente debido al primero de los pasos.

Por este motivo, tendremos que usar técnicas para reducir el número de candidatos, el número de transacciones o bien el número de comparaciones.

A continuación se exponen métodos de extracción de reglas más refinados y que, a pesar de ser clásicos, son usados en nuestros días.

4.3.2. Método Apriori

Se basa en la propiedad antimonótona de la medida de soporte, que puede enunciarse como sigue.

> Dado un itemset X contenido en un itemset Y, el soporte de X no puede ser inferior al soporte de Y.

$$\text{supp}(X) \geq \text{supp}(Y)\, \forall X, Y / (X \subseteq Y)$$

Esto puede leerse del siguiente modo.

> Para que un itemset Y sea frecuente, todos sus subconjuntos X tienen también que serlos.

De este modo, Apriori genera una serie de conjuntos de k-itemsets frecuentes con una longitud k cada vez mayor, hasta que se llega a un itemset infrecuente, que no cumple los umbrales de soporte o confianza. En ese momento, Apriori deja de explorar todos los itemsets que contienen al itemset infrecuente encontrado [6].

La figura 6 ilustra la cantidad de itemsets no explorados gracias a Apriori.

Hay que destacar que el método Apriori es muy conocido y muy usado, aunque existen muchos factores que afectan a su rendimiento, como la elección del umbral de mínimo soporte, el número de items en la base de datos, el tamaño de la base de datos y la longitud media de las transacciones. Umbrales demasiado

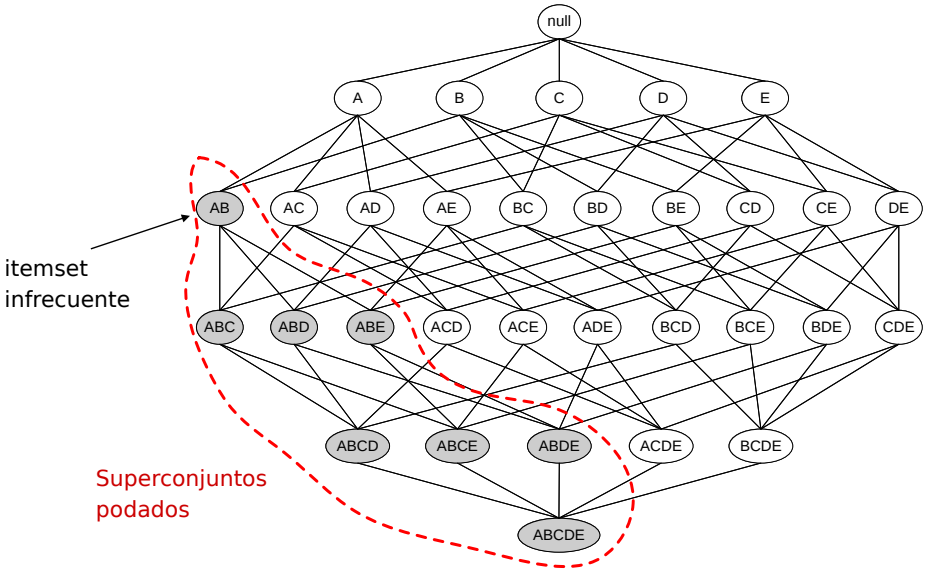


Figura 6: Ilustración de Apriori.

bajos en bases de datos grandes pueden hacer que Apriori requiera una cantidad de memoria muy grande, y que sus múltiples iteraciones le hangan tener un tiempo de ejecución bastante elevado.

4.3.3. Método Eclat

Se basa en el mismo principio que Apriori, pero almacena una lista (*tid-list*) con la que se acelera el cálculo del soporte. Presenta la desventaja de acelerar el proceso de extracción de reglas a costa de un uso de memoria muy elevado.

4.3.4. Método FP-growth

Consiste en extraer reglas a partir de una representación especial de la base de datos llamada FP-tree. Esta estructura consta de una tabla cabecera y de un grafo de transacciones, con las que se representan, de forma comprimida, todas las transacciones de la base de datos. Una vez que la estructura ha sido construida, se usa una aproximación recursiva para extraer los itemsets que cumplen los umbrales de soporte y confianza [7].

FP-growth presenta múltiple ventajas, como una reducción significativa tanto en la cantidad de memoria usada como en el tiempo de ejecución, produciendo resultados idénticos a Apriori.

4.4. Conjuntos maximales y cerrados

Cuando el tamaño de la base de datos es muy grande, el conjunto de itemsets frecuentes crece de forma exponencial. Esto supone problemas de almacenamiento, por lo que será necesario buscar representaciones alternativas que consigan reducir el conjunto inicial, sin perder la capacidad de generar todos los itemsets frecuentes. En este momento, resulta útil hablar de los itemsets maximales y cerrados.

Un *itemset frecuente maximal* es aquel itemset frecuente para el que ninguno de sus superconjuntos inmediatos son frecuentes. A partir de los itemsets maximales es posible obtener todos los itemsets frecuentes, pueto que serán todos los subconjuntos de items que puedan formarse a partir de ellos. No obstante, el soporte de cada itemset frecuente deberá ser calculado de nuevo.

Un *itemset frecuente cerrado* es aquel itemset frecuente para el que ninguno de sus superconjuntos inmediatos tiene un soporte igual al del suyo. A partir de los itemsets frecuentes cerrados es posible obtener todos los itemsets frecuentes. Además, cualquier subconjunto de ellos que no sea otro itemset cerrado tiene el mismo soporte que ellos. Por este motivo, no será necesario volver a calcular los soportes de cada itemset. Como contrapunto, los itemsets cerrados son más numerosos que los maximales, por lo que será necesario más espacio para almacenarlos.

La figura 7 muestra una representación gráfica de estos dos tipos de itemsets.

Como propiedad, cabe destacar que los itemsets maximales son un subconjunto de los itemsets cerrados, que a su vez son un subconjunto de los itemsets frecuentes.

Dependiendo de los recursos de memoria y tiempo, será conveniente usar una u otra representación. En concreto, si la eficiencia es más importante, podremos usar los itemsets cerrados, mientras que si la memoria no es demasiado elevada, será conveniente usar los itemsets frecuentes.

4.5. Generación de reglas

Una vez se han encontrado los itemsets frecuentes, se generan las reglas haciendo todas las posibles combinaciones de los elementos del itemset. Solo las reglas que superen el umbral $minConf$ son seleccionadas.

Para cada k-itemset frecuente, existen k reglas con un solo atributo en el consecuente, y $2^k - 2$ reglas en total, sin embargo, las primeras son más frecuentes. Por ejemplo, en el caso de $k = 3$, tenemos:

- Reglas con un solo atributo en el consecuente

$$AB \to C \qquad AC \to B \qquad BC \to A$$

- Todas las posibles reglas

$$\begin{array}{lll} AB \to C & BC \to A & B \to AC \\ AC \to B & A \to BC & C \to AB \end{array}$$

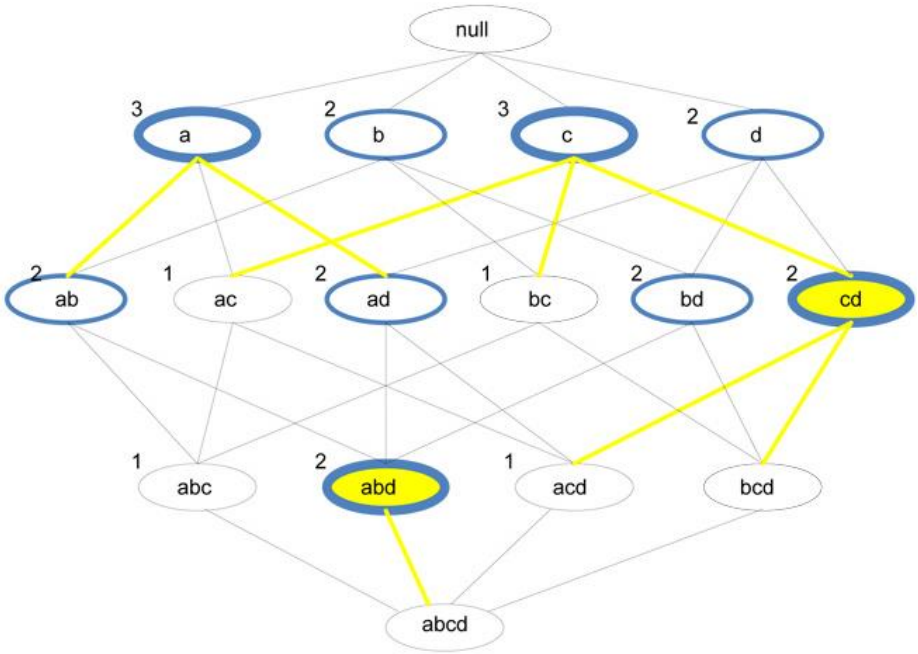


Figura 7: Itemsets frecuentes, maximales y cerrados. Las elipses con borde azul representan los itemsets frecuentes; el borde azul grueso indica que el itemset es, además, cerrado; el fondo amarillo representa que el itemset es maximal. Se considera un soporte mínimo $minSup = 2$.

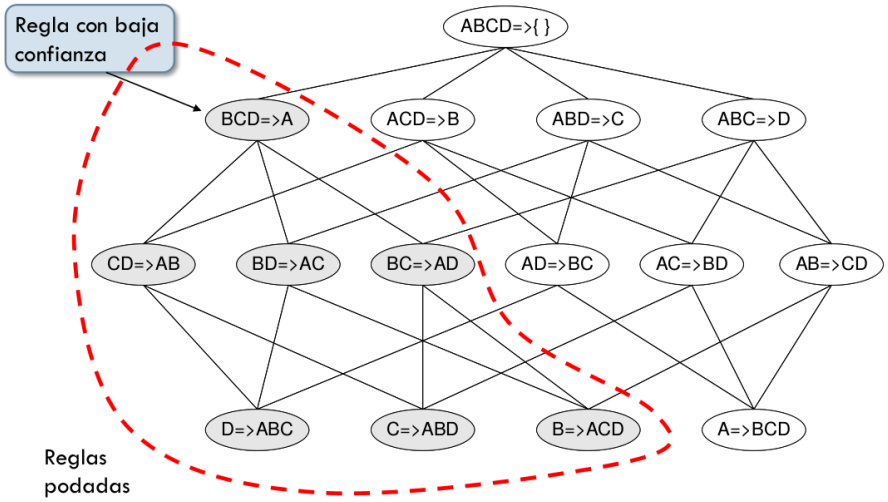


Figura 8: Reglas podadas por la propiedad antimonótona de la confianza de las reglas del mismo itemset.

Cabe mencionar que la confianza no tiene la propiedad antimonótona, por lo que no es posible saber si

$$\text{conf}(AB \to C) \gtrless \text{conf}(A \to C)$$

, sin embargo, las reglas generadas a partir del mimso itemset sí tienen la propiedad antimonótona

$$\text{conf}(AB \to C) \geq \text{conf}(A \to CB)$$

$$\text{conf}(AB \to C) \geq \text{conf}(B \to AC)$$

De este modo, habrá reglas que no necesiten ser exploradas por conocerse de antemano que tendrán una baja confianza, como se muestra en la figura 8.

4.6. Evaluación: Medidas de interés

La interpretación de las reglas obtenidas es un tema complejo, ya que las reglas de asociación representan tendencia y no implicación lógica, como se esbozó anteriormente. Los problemas de la interpretación pueden venir por no tener datos apropiados, por confusiones semánticas de los usuarios que tienen que interpretarlas o por no saber interpretar correctamente las medidas. De hecho, las medidas de soporte y confianza son, en la mayor parte de los casos, insuficientes, ya que no proporcionan una informacion completa sobre la regla.

La solución a este último problema es introducir nuevas medidas de calidad o interés, que, junto a las medidas clásicas, permitan una mejor evaluación

e interpretación de las reglas. Hay que tener en cuenta que ninguna de estas medidas es perfecta, y que por tanto será necesario prestar atención a varias de ellas cada vez que se desee evaluar una regla.

A continuación se presentan algunas de las medidas de interés más usadas.

4.6.1. Confianza confirmada

Es una medida que trata de acercar la semántica de las reglas de asociación a la de la lógica formal. Nos indica hasta qué punto es útil un item A para predecir otro item C. Se define como

$$\text{conf}(A \to C) - \text{conf}(A \to \neg C)$$

Su rango es $[-1, 1]$, donde 0 significa que es imposible predecir C a partir de A (independencia); 1 significa que A predice o implica C; -1 significa que A predice o implica $\neg C$.

4.6.2. Lift (interés)

Se trata de una medida simétrica, que mide la asociación entre dos itemsets A y C, sin tener en cuenta cuál está en el antecedente o cuál está en el consecuente. Se define como

$$\text{lift}(A \to C) = \frac{\text{conf}(A \to C)}{\text{supp}(C)} = \frac{\text{supp}(A \cup C)}{\text{supp}(A)\,\text{supp}(C)}$$

Su rango es $[0, +\infty)$, donde 1 significa independencia; valores menores que 1 significan dependencia negativa; valores altos implican dependencia cada vez más fuerte.

Al no estar acotado, el *lift* no permite comparar reglas de distintas bases de datos.

4.6.3. Convicción

Para una regla $A \to C$ se define como

$$\frac{\text{supp}(A)\,\text{supp}(C)}{\text{supp}(A \to \neg C)}$$

Su rango es $(0, +\infty)$, donde 1 significa independencia estadística; valores menores que 1, dependencia negativa. Valores entre 1,01 y 5 se consideran interesantes, mientras que valores superiores a 5 son reglas obvias.

Al considerarse $\text{supp}(\neg C)$ permite resolver el problema de itemsets muy frecuentes.

4.6.4. Otras medidas de interés

Existen muchas más medidas, como el factor de certeza, que es una medida de implicación relacionada con el lift y la convicción; Yule's Q, que representa la correlación entre eventos dicotómicos relacionados positivamente; o la diferencia absoluta de confianza, que es otra medida de implicación.

Otras medidas interesantes son la diferencia de información, que se basa en la entropía para conocer la ganancia de información que proporciona el antecedente sobre el consecuente.

4.6.5. Medidas subjetivas

Son medidas que tienen en cuenta el conocimiento previo del usuario. Puede medirse, por ejemplo, la utilidad de una regla, basándonos en criterios subjetivos sobre las acciones a realizar al descubrir una regla nueva. Podríamos considerar, por ejemplo, una acción como una campaña de márketing. Entonces, cabe cuestionarse la utilidad de la regla en función del tiempo de vida de la información, en cuanto a su interés para llevar a cabo la acción, el esfuerzo que requiere la actuación, el impacto o los efectos laterales.

De especial interés son las reglas inesperadas, que son aquellas que contradicen las creencias del usuario. Para descubrir reglas inesperadas, será necesario definir cómo representar las creencias del usuario y cómo medir lo inesperado. Los enfoques principales incluyen medidas probabilísticas, de distancia sintáctica y de contradicción lógica.

Un tipo de contradicciones son las paradojas, que son reglas de asociación que revelan tendencias contradictorias. Estas paradojas son útiles para buscar planteamientos distintos del problema.

4.7. Interpretaciones

El objetivo último de las reglas de asociación, al igual que el del aprendizaje no supervisado en general, es conseguir conocimiento novedoso y potencialmente útil. Para ello, es necesario establecer una estructura en los datos, que se corresponda con el marco formal.

En este contexto, podemos definir una interpretación como una correspondencia que se establece entre elementos de la estructura de datos y elementos del marco formal. Una interpretación específica define, por tanto, qué entendemos por item y transacciones en el contexto de datos que tenemos. De esta forma, la interpretación contribuye a definir la semántica de las reglas.

4.7.1. Interpretación tabular común

En una base de datos con estructura de tabla, los items pueden ser interpretados como parejas (atributo,valor) y las transacciones como registros de la tabla. En este caso, una regla como
(puesto,administrativo) $\rightarrow$ (estudios,medios)
puede leerse como «todo el que tiene un puesto de administrativo tiene estudios

medios», o, equivalentemente, «puesto de administrativo implica tener estudios medios».

4.7.2. Reglas jerárquicas

En el caso en que tengamos una estructura jerárquica de items (por ejemplo, varios tipos de refresco dentro del item genérico «refresco»), podemos establecer reglas jerárquicas. En este caso, los datos consisten en un conjunto de transacciones que contienen items básicos y una jerarquía de categorías que agrupa los items en varios niveles. De esta forma, podemos dar la interpretación de que los items son la unión de los items básicos y sus categorías de la jerarquía, y las transacciones se forman tomando cada transacción de items básicos y añadiendo los ancestros en la jerarquía de todos los items presentes.

Esto tiene una gran utilidad, ya que en el caso de que los items de niveles bajos no tengan soporte para generar reglas, pueden tomarse items superiores en la jerarquía, mientras que si el soporte es excesivo, puede bajarse en el nivel jerárquico.

4.7.3. Reglas secuenciales

Un patrón secuencial se construye a partir de una secuencia de items básicos que tienden a aparecer en un orden prefijado. Este tipo de reglas es muy útil en la minería de texto, ya que se evalúa la confianza de que aparezcan sucesiones de palabras. De este modo, pueden establecerse relaciones entre secuencias en el antecedente y secuencias en el consecuente.

4.7.4. Reglas cuantitativas

Utilizadas para extraer reglas cuando tenemos datos estructurados con variables que pueden tomar valores muy variados y que pueden ser ordenados. Es el caso de las parejas (atributo,valor), donde *valor* es un número real.

La solución pasa por dividir el dominio de la variable valor en distintos intervalos, que deberán ser definidos por un experto o, en su defecto, por métodos automáticos. Las ventajas de la primera opción es la riqueza semántica que se aporta al problema, haciendo más fácil la interpretación. Sin embargo, la división en intervalos automática puede aportar información nueva, siempre y cuando se ajuste bien el método de división.

Referencias

[1] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[2] Wikimedia Commons. File:dbscan-illustration.svg — wikimedia commons, the free media repository, 2016. [Online; accessed 6-September-2017].

[3] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[4] Juan M Estevez-Tapiador, Pedro Garcia-Teodoro, and Jesus E Diaz-Verdejo. Stochastic protocol modeling for anomaly based network intrusion detection. In *Information Assurance, 2003. IWIAS 2003. Proceedings. First IEEE International Workshop on*, pages 3–12. IEEE, 2003.

[5] William W Cohen. Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning*, pages 115–123, 1995.

[6] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, A Inkeri Verkamo, et al. Fast discovery of association rules. *Advances in knowledge discovery and data mining*, 12(1):307–328, 1996.

[7] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *ACM sigmod record*, volume 29, pages 1–12. ACM, 2000.

[8] Christopher D. Manning, Prabhakar Raghavan, and Hinrich SchÃŒtze. *Introduction to Information Retrieval*. Cambridge University Press, 2012.

[9] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*. The MIT Press, 2001.

[10] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York Inc., 2006.

[11] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[12] Charu C. Aggarwal. *Outlier Analysis*. Springer, 2016.

[13] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, Sept 2009.

[14] Earl Harris. Information gain versus gain ratio: A study of split method biases. *AMAI*, 2002.

[15] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 2(3):408–421, 1972.

[16] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993.

[17] Kenneth Lai and Narciso Cerpa. Support vs. confidence in association rule algorithms. In *Proceedings of the OPTIMA Conference, Curicó*, 2001.

[18] N Marín, MD Ruiz, and D Sánchez. Fuzzy frameworks for mining data associations: fuzzy association rules and beyond. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(2):50–69, 2016.