TRABAJO TEÓRICO FINAL

MÁSTER EN CIENCIA DE DATOS

Minería de Datos: Aprendizaje no supervisado y detección de anomalías.

Autor
José Ángel Díaz García

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Granada, Enero de 2018

Índice general

1. Introducción **5**

1.1. Técnicas supervisadas y no supervisadas 6

1.2. Organización del trabajo 6

2. Clustering **7**

2.1. Introducción 7

2.2. Medidas de similitud 8

2.3. Similitud en atributos continuos 8

2.4. Similitud en atributos no continuos 9

2.5. Métodos 9

2.5.1. Particionales 10

2.5.2. Jerárquicos 12

2.6. Validación 13

2.7. Extensiones del Clustering 14

2.8. Aplicaciones 14

3. Detección de anomalías **16**

3.1. Introducción 16

3.2. Métodos 17

3.2.1. Métodos Supervisados 17

3.2.2. Métodos Semi-Supervisados 18

3.2.3. Métodos No Supervisados 18
3.3. Validación 18
3.4. Aplicaciones 20

4. Reglas de Asociación **22**
4.1. Introducción 22
4.2. Validación 23
4.3. Obtención de reglas 24
4.4. Principales algoritmos 25
4.4.1. Apriori 25
4.4.2. Eclat 26
4.4.3. FP-Growth 26
4.5. Aspectos Avanzados 27
4.6. Aplicaciones 27

Índice de figuras

2.1. Ejemplo del algoritmo kmedias. 11
2.2. Ejemplo de dendograma y clusters asociados. 12

4.1. Distintas medidas aplicables a reglas de asociación. 29

Índice de tablas

2.1. Medias de similitud en atributos no continuos. 10

3.1. Matriz de confusión para dos clases. 18

Capítulo 1

Introducción

Actualmente nadie debería sorprenderse cuando escuche que vivimos en la *sociedad de la información*, concepto acuñado para referenciar a una sociedad cambiante y donde la manipulación de datos e información juega un papel más que relevante en las actividades sociales, culturales y sobre todo, económicas. El tratamiento de estos datos puede suponer una ardua labor, más aún cuando el volumen de estos es tan grande que los paradigmas para su procesado deben migrar hacia nuevas vertientes y aún más cuando estos datos provienen de fuentes tan dispares como nuestras tendencias en la compra diaria, el uso que le damos a una tarjeta de crédito o a una red social... Es por ello, que fruto de la necesidad del análisis y la obtención de información de estos datos en especie des-estructurados y aparentemente carentes de significado surgen técnicas y herramientas capaces de procesar y obtener información útil y relevante.

Una de estas técnicas es la minería de datos, que podría ser definida como el proceso de obtención de información relevante y no trivial sobre conjuntos de datos, de manera que esta puede ser utilizada en los procesos de toma de decisiones de empresas o entidades, sin olvidar el papel académico e investigador donde el uso de estas técnicas es innumerable.

Dentro del área de la minería de datos, encontramos además distintos enfoques. Estos enfoques pueden ir en función de diversos factores, pero sin duda la división de técnicas de minería de datos más extendida, es la que las divide entre técnicas dirigidas o aprendizaje supervisado y técnicas no dirigidas, o aprendizaje no supervisado. El presente trabajo, se centra en

un estudio de estas últimas técnicas, desde un enfoque de resumen y que pretende condensar los conceptos más relevantes de cada una de las mismas, no sin antes, diferenciarlas de las técnicas de aprendizaje supervisado, algo necesario para su correcto estudio posterior.

1.1. Técnicas supervisadas y no supervisadas

Las técnicas de minería de datos podrían ser divididas en dos vertientes, **aprendizaje supervisado** y **aprendizaje no supervisado**. Pese que nos centraremos en estudiar las técnicas no supervisadas, es necesario comprender la diferencia entre ambas y en eso se centrará este capítulo.

Las técnicas de aprendizaje supervisado, se presentan como un problema de elección de la clase o categoría que le será asignada a una nueva muestra u observación. Los algoritmos consiguen esto basando su predicción en ciertos parámetros y un conjunto de muestras ya clasificadas *apriori* conocido como conjunto de entrenamiento o *training-set*. Esta es la principal diferencia de estas técnicas con las técnicas no dirigidas donde no disponemos de este conocimiento previo sobre los datos y por tanto estos algoritmos se centrarán en gran medida en obtener relaciones entre los mismos.

Dentro de las técnicas de supervisadas, las más famosas son la regresión y la clasificación, por otro lado, si nos centramos en las técnicas no supervisadas encontramos como principales enfoques, el clustering, las reglas de asociación y la detección de anomalías.

1.2. Organización del trabajo

El trabajo está organizado siguiendo como hilo conductor las transparencias de la asignatura '*Minería de datos: Aprendizaje no supervisado y detección de anomalías*', del máster en Ciencia de Datos siguiendo como referencia las transparencias [1] [2] [3] de los distintos profesores de la asignatura. Tras este capítulo de introducción donde se introduce el tema y la diferencia entre las técnicas dirigidas y no dirigidas, se ilustran cada una de las técnicas vistas en el transcurso de la asignatura, comenzando por el clustering y las anomalías para finalizar con las reglas de asociación.

Capítulo 2

Clustering

En este segundo capítulo, estudiaremos las técnicas de agrupamiento o clustering, desde un enfoque en profundidad que nos llevará desde una introducción *grosso modo* del problema (sección 2.1) al estudio de técnicas extendidas de clustering (sección 2.7) o sus aplicaciones (sección 2.8) , con las que se dará por terminado este capítulo.

2.1. Introducción

El clustering, se enmarca dentro del aprendizaje no supervisado y es una técnica de minería de datos descriptiva. Estas técnicas, a diferencia de las predictivas, no se usan para predecir una salida sino que nos ofrecen herramientas (gráficos, reglas, agrupamientos) para entender y describir de una mejor manera que está ocurriendo con unos determinados datos de entrada, de los que no disponemos información previa acerca de su estructura. En el caso del clustering, **tratamos de encontrar agrupaciones de los datos de entrada, representados por un vector de atributos, en función de distintas medidas de similitud**, este concepto, será estudiado en detalle en la siguiente sección.

2.2. Medidas de similitud

Para poder discernir entre si una determinada muestra es similar a otra, se usan las denominadas **medidas de similitud**. Antes de entrar en detalle en la definición de estas medias, es necesario destacar la **naturaleza subjetiva** del clustering, o lo que es lo mismo, que en función del problema, los datos y las preguntas a las que se intentan dar respuesta puede haber varias soluciones apropiadas. Por otro lado, cabe esperar una fase previa de pre-procesado de datos que puede incluir filtrado de variables (generalmente guiadas por un experto) o normalizaciones, para poder obtener estas distancias o similitudes apropiadamente.

Es menester mencionar que los datos de partida, podrán darse en forma de dataset (Items - Variables) o por medio de una matriz de proximidad, que habitualmente será obtenida del dataset pero que en ciertas aplicaciones puede generarse directamente.

2.3. Similitud en atributos continuos

Estas métricas, se usan para medir la distancia entre dos individuos x e y, se usan en atributos continuos y estos deberán estar normalizados en la mayoría de los casos. Ademas, deberán satisfacer las propiedades reflexiva, simétrica y desigualdad triangular. Algunas de las medidas más famosas son:

- Distancia Minkowsky: Es una medida que agrupa la distancia manhattan y euclídea. Puede expresarse con la siguiente fórmula.

$$d_r(x,y) = (\sum_{j=1}^{J} |x_j - y_j|^r)^{\frac{1}{r}}, r \geq 1 \tag{2.1}$$

- Distancia Euclídea:

  Es la medida más usada y la que mejor se adapta a atributos continuos, aunque puede verse afectada por outliers. Quedaría definida con la expresión matemática:

$$d_2(x,y) = \sqrt{\sum_{j=1}^{J}(x_j - y_j)^2} \tag{2.2}$$

- Distancia Manhattan:

  Esta métrica también es conocida como métrica del taxista, y su nombre viene dado por el recorrido que un coche debería de hacer por Manhattan para ir de un punto A al B, es decir, con líneas rectas que son la suma de las diferencias absolutas de sus coordenadas. Su fórmula sería:

$$d_1(x,y) = \sum_{j=1}^{J}|x_j - y_j| \tag{2.3}$$

- Distancia de Chebyshev:

  Esta medida es menos conocida, y representa la distancia con un símil del mundo del ajedrez, en el que la distancia entre dos muestras vendrá dada por el número movimientos que el rey tendría que hacer para llegar de uno a otro. Podíamos definirlo matemáticamente de la siguiente manera:

$$d_\infty(x,y) = max_{j=1\ldots J}\,|x_j - y_j| \tag{2.4}$$

2.4. Similitud en atributos no continuos

En este punto encontramos multitud de medidas en función del dominio del problema. Aunque las medidas para atributos no continuos son muy variadas se han recogido algunas de las más famosas en la tabla 2.1.

2.5. Métodos

En esta sección veremos los distintos métodos o enfoques de agrupamiento, así como introduciremos a grandes rasgos algunos de los principales algoritmos de cada vertiente.

Medida	***Idea***	***Uso***
Levenshtein	Nº de operaciones para transformar una cadena en otra	Se usa en correctores ortográficos, sistemas de reconocimiento de voz o plagios entre otros.
Jaccard	Basada en teoría de conjuntos	Su principal uso está en el campo de la Recuperación de Información
Datos Binarios	Se basa en la diferencia entre dos cadenas de números binarios	Biología y estudio de comunidades ecológicas
Coseno	Se basa en la similitud coseno sobre un Document Term Matrix	Su principal uso está en el campo de la Recuperación de Información en buscadores como Google.

Tabla 2.1: Medias de similitud en atributos no continuos.

2.5.1. Particionales

La principal carácteristicas de los métodos de clustering particionales reside en parámetro ***k***, que podrá estar definido o no. Este valor, es un número entero que determinará el número de particiones (*k=2 ->2 particiones*) a realizar del conjunto global. Los elementos de cada uno de los grupos 'se parecerán' más entre sí, que entre cualquier miembro de otro grupo distinto. Las particiones, podrán atender a criterios locales, los cuales suponen que cada grupo está representado por un un elemento prototipo, o globales, basados en la estructura local de los datos, como la densidad. En función de cada uno de estos, encontramos distintos algoritmos, como puede ser en algoritmo de las K-medias (global) [4] o el DBSCAN (local) [5], los cuales definiremos a continuación.

K-Medias

El método de las k-Medias es bastante sencillo. Se parte de un valor de K, que indicará el número de clusters finales, y el número de centroides aleatorios iniciales: Los pasos del algoritmo serían los siguientes:

1. Se obtienen k muestras aleatorios sobre la muestra, serán nuestros centroides de partida.

2. Cada elemento en la muestra, se asigna al centroide de partida más cercano obteniendo k grupos.

3. Dentro de cada grupo, se calculan sus centroides y se vuelven a asignar los elementos más cercanos, refinando los grupos iniciales.

4. Mientras el proceso no converja, se continúa.

Estos pasos, pueden verses ilustrados en la figura 2.1.

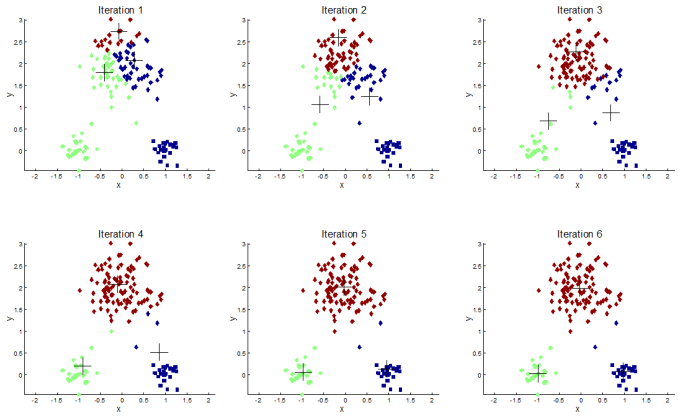

Figura 2.1: Ejemplo del algoritmo kmedias.

DBSCAN

Este algoritmo es un método basado en densidad (criterio local), la idea principal de estos métodos reside en identificar regiones en el espacio del problema cuya densidad de muestras difiera notablemente de otras, identificando los grupos en función de estas regiones y diferencias. El algoritmo DBSCAN usa densidad basada en centros, donde estimaremos la densidad de una región contando el número de muestras que residen dentro de un radio fijado como parámetro y que se denomina *eps*. Una vez fijado este parámetro, se centrará en obtener iterativamente **puntos núcleo**, (serán aquellos centrales a una región de gran densidad) y **puntos frontera** (aquellos que delimitan una región de alta densidad). Los pasos serían:

1. Se analiza punto por punto y se comprueba si para un valor de *eps* ese punto es un punto núcleo.

2. Si el punto es núcleo se crea un grupo y se buscan otros núcleos alcanzables a partir de él. Si se localiza alguno, se fusionan los grupos.

3. Terminaremos cuando no se pueda añadir ya ningún punto a ningún grupo.

2.5.2. Jerárquicos

La principal idea del agrupamiento jerárquico reside en una sucesión de particiones que se anidan una continuación de la otra, de manera que determinados ejemplos pertenecientes a una partición n están totalmente incluidos en una partición $n+1$. Este tipo de clusters, se representa mediante dendogramas (figura 2.2) y no necesitan el parámetro k que vimos en la sección anterior.

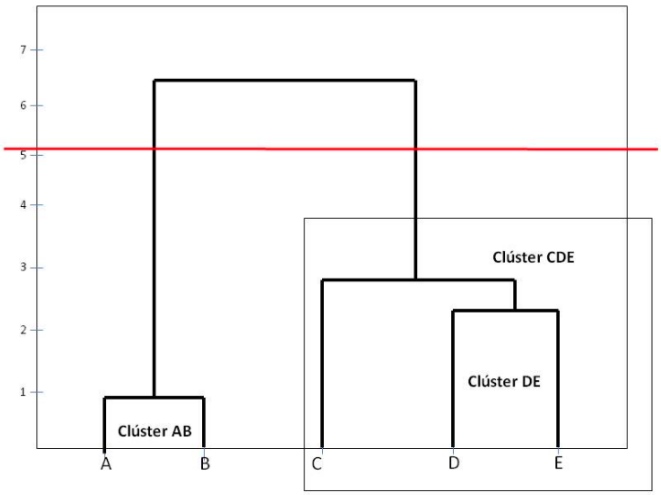

Figura 2.2: Ejemplo de dendograma y clusters asociados.

Los enfoques de agrupamiento jerárquico son en su mayoría aglomerativos, es decir, considerando que cada ítem representa un grupo, avanzan en altura agregando ítems entre si que formarán los grupos finales. Dos de las técnicas más famosas son el **enfoque basado en grafos** y el **algoritmo de Jhonson**.

Enfoque basado en Grafos

Se considera cada ítem como vértice de un grafo a raíz del cual, por medio de conexión de vértices, se generan las particiones. Estas conexiones entre vértices pueden generarse de dos formas:

1. Enlace Simple: Obtendremos los grupos mediante la obtención de las componentes conexas del grafo.
2. Enlace Completo: Obtendremos los grupos, al identificar los subgrafos completamente conectados.

Algoritmo de Jhonson

Este algoritmo se basa en la transformación de la matriz de distancia, que será reducida cada vez que el algoritmo consiga identificar un nuevo grupo. Este proceso es iterativo y se basa en distintas formas de calcular la proximidad entre grupos que pueden ser tales como, el mínimo, el máximo, la media de grupos o la distancia entre centroides.

2.6. Validación

La validación, es uno de las etapas más delicadas en cualquier proceso de ciencia de datos, ya que con ella, podremos discernir si nuestros modelos se comportan adecuadamente y se amoldan a la realidad. Para más inri, en enfoques no supervisados como es el caso del clustering, donde no conocemos nada *a priori* sobre la estructura de los datos, el proceso de validación puede suponer una ardua labor. Pese a su dificultad en algunos casos, la validación de un proceso de agrupamiento es muy interesante ya que nos permitirá discernir entre agrupamientos y ruido o comparar técnicas de agrupamiento.

La evaluación de resultados podrá hacerse siguiendo dos criterios:

1. Criterios Externos: Se apoyan en información adicional, como es un conjunto de entrenamiento típico y una validación donde suprimimos el valor del cluster (clase).

2. Criterios Internos: Se obtienen a partir de los propios datos, y responden preguntas como: ¿Qué valor de k usar?, cuya respuesta vendrá dada por el valor de la suma del error cuadrado, o ¿Cómo de buenos son mis cluster?, pregunta que hallará la respuesta en las medidas de **cohesión** y **separación**.

2.7. Extensiones del Clustering

Pese a que los métodos estudiados anteriormente son los más extendidos, la potencia y la utilidad de las técnicas de clustering hacen que cada vez sean más las extensiones de los métodos de agrupamiento que tratan de mejorar los métodos clásicos o de solventar problemas de eficiencia de algunos métodos como por ejemplo, el agrupamientos jerárquico.

Algunas de estas técnicas pueden ser la técnica BIRCH [6], CURE [7] o ROCK [8] usadas todas para aumentar la eficiencia de las técnicas de clustering jerárquico y por otro lado, el método de las **k-medias difuso**, que hace uso de lógica difusa para mejorar los resultados del algoritmo k-medias; sobre el cual, además, encontramos en la literatura distintas aproximaciones que ilustran el uso de *medoides* frente a las medias. Un algoritmos de esta vertiente es el algoritmo CLARANS [9].

2.8. Aplicaciones

Desde su primera incursión allá por finales de los años 60 en el campo del análisis de datos, las técnicas de clustering han sido aplicadas a distintos problemas dentro de la informática además de otras áreas como la biología, la medicina o el marketing. Acorde a Kumar [10], algunas de las áreas y aplicaciones más famosas o más extendidas dentro del clustering podrían ser:

- Psicología y medicina: Una enfermedad podrá tener distintos síntomas o variaciones en la presentación de los mismos, el clustering, puede ser usado en estas áreas para identificar estas variaciones y agrupar en subcategorias.

- Marketing y negocios: El clustering en marketing tiene infinitud de aplicaciones desde ser utilizado para segmentar clientes a la detección de comunidades en redes sociales para aplicar una determinada promoción.
- Meteorología: Entender el clima de nuestro planeta requiere el estudio y representación de patrones, las técnicas de agrupamiento pueden ser utilizadas para la búsqueda de estos.

Para ilustrar ejemplos reales de aplicación de las técnicas de clustering y remarcar su importancia en el ámbito de investigación, se ha indagado acerca de estudios recientes que utilicen métodos de agrupamiento, algunos de estos estudios pueden ser el artículo de Moosavi et al.[11], donde se proponen técnicas de clustering para agrupar usuarios en redes sociales en función de sus acciones, o el artículo de Baier et al. [12] donde se proponen clustering de imágenes con fines enfocados al marketing.

Capítulo 3

Detección de anomalías

Aunque se suele estudiar dentro del campo del aprendizaje no supervisado, la detección de anomalías es un campo a caballo entre este y el aprendizaje supervisado, habiendo sido estudiada desde ambos enfoques y con técnicas propias de cada uno de ellos en innumerables ocasiones. En este capítulo, se introducirá el concepto de anomalía y detección de las mismas (sección 4.1), los métodos aplicados en su detección (sección 3.2) y se finalizará puntualizando sus métodos de validación (sección 3.3) y sus aplicaciones e implicaciones en problemas reales (sección 4.6).

3.1. Introducción

Antes de comenzar a definir el proceso de la detección de anomalías, cabría la necesidad de preguntarnos, ¿Qué es una anomalía? Acorde con Aggarwal [13], un *outlier* o anomalía podría definirse como un ejemplo cuyas características son significativamente diferentes del resto de los datos. Localizar las anomalías será por tanto la meta final del proceso de detección de anomalías, teniendo en cuenta que estas pueden estar debidas a dos motivos:

1. Errores: Son ejemplos que no pertenecen al dominio real del problema sino que se han debido a un error en la captura de los datos o incluso a procesos de pre-procesado previos al proceso de detección de anomalías.

2. Datos reales: Son datos que a diferencia del caso anterior no presentan error en su captura o procesado, pero en los que por una o varias de sus variables tienen valores que los convierten en anomalías. Estos datos, pueden llegar a ser muy interesantes y deben ser analizados de manera concienzuda.

De esta pequeña introducción podríamos desgranar que la mayor dificultad en el proceso de detección de anomalías está en que desconocemos por completo la naturaleza o dominio de lo que buscamos, por lo que el proceso es complicado, por lo que habrá ciertas ocasiones en las que podríamos prescindir del proceso de detección de anomalías como al usar arboles como el C4.5 o los métodos basados en reglas de asociación cuyos valores de soporte y confianza dejarán fuera del proceso de minería de datos a los outliers, sin tener que tratar estos previamente. Por otro lado, habrá otras ocasiones donde si que habrá que aplicar el proceso previo de detección de outliers, como al usar métodos de regresión, por naturaleza muy sensibles a las anomalías. Llegado por tanto el caso de necesitar la detección de anomalías, dispondremos de varias técnicas para dicha tarea, sobre la discusión de éstas será de lo que verse la siguiente sección.

3.2. Métodos

En esta sección trataremos sobre los distintos enfoques desde los que puede ser estudiada la detección de anomalías.

3.2.1. Métodos Supervisados

Estos métodos se basan en la construcción de un modelo de clasificación, basado en un conjunto de datos de entrenamiento donde tenemos anomalías etiquetadas como tal. Debemos tener en cuenta que la proporción de muestras que correspondan a una anomalía en relación con las muestras que no lo son, estarán en una gran minoría por lo que estaremos ante un problema de clases no balanceadas. Estos problemas son muy delicados y podrán afrontarse desde dos enfoques, según si nos centramos en las instancias o en el algoritmo.

Métodos basados en instancia

Estos métodos mutan los datos antes de aplicar la clasificación. Generalmente, se centran en aumentar las muestras de la clase minoritaria (oversampling) o disminuir las de la clase mayoritaria (undersampling).

Métodos basados en algoritmos

Estos métodos mantienen el conjunto de datos de entradas inamovible y se centran en mejorar el proceso de entrenamiento mediante el uso de pesos.

3.2.2. Métodos Semi-Supervisados

3.2.3. Métodos No Supervisados

3.3. Validación

La salida de un método de detección de anomalías puede ser si o no, dependiendo de si es o no es una anomalía. Acode a esta salida podremos construir con ayuda de un experto la matriz de confusión que podemos ver en la (Tabla 3.1).

	Positive prediction	*Negative prediction*
Positive class	True positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

Tabla 3.1: Matriz de confusión para dos clases.

Esta matriz (Tabla 3.1) almacena los ejemplos cuya clasificación se acierta y aquellos cuya clasificación es errónea. De estas medidas podemos obtener el ***accuracy rate*** cuya formula podemos verla en la ecuación 3.1.

$$Acc = \frac{TP + TN}{TP + FN + TN + FP} \tag{3.1}$$

Este método es uno de los métodos de evaluación más usado, aunque su comportamiento no es del todo apropiado con problemas no balanceados, como el caso de outliers, ya que no tiene en cuenta la distribución de los ejemplos de cada clase. Es por ello, que han surgido propuestas para evaluación que mejoran los resultados del *accuracy rate*. Y que se basan en medidas que tienen en consideración el porcentaje de cada una de las muestras de las clases estudiadas frente a las demás lo que facilita y mejora el comportamiento de los métodos de evaluación frente a problemas con representaciones de clase en clara minoría, como los outliers. Algunas de las medidas más extendidas son:

- ***True positive rate***: Porcentaje de instancias positivas correctamente clasificadas. Este valor también es conocido como sensibilidad o recall y juega un papel muy importante en algunos de los métodos de evaluación que veremos al finalizar esta sección.

$$TPR = \frac{TP}{TP + FN} \tag{3.2}$$

- ***True negative rate***: Porcentaje de instancias negativas correctamente clasificadas. Este valor también se le conoce como especifidad, y al igual que el anterior tiene un papel relevante en métodos de evaluación que veremos en este capítulo.

$$TNR = \frac{TN}{TN + FP} \tag{3.3}$$

- ***False positive rate***: Porcentaje de instancias positivas mal clasificadas.

$$FPR = \frac{FP}{FP + TN} \tag{3.4}$$

- ***False negative rate***: Porcentaje de instancias negativas mal clasificadas.

$$FNR = \frac{FN}{TP + FN} \tag{3.5}$$

- ***Precisión***: Es el ratio de todos los ejemplos que son realmente positivos, frente a todas las observaciones cuya predicción es positiva y puede ser calculada fácilmente con los valores estudiados hasta el momento con la siguiente ecuación.

$$precision = \frac{TP}{TP + FP} \tag{3.6}$$

- ***F-measure*** [14]: Se basa en una correlación de la sensibilidad o recall y la precisión.El cálculo del F-Measure es sencillo y sofisticado al mismo tiempo ya que añade un valor x que permite al usuario ponderar una parte u otra en función de diversos factores. Los valores típicos de x suelen ser 0.5, 1 y 2 lo que significaría respectivamente doble peso para precisión, mismo peso para ambas y doble peso para el recall.

$$F - Measure = (1 + x) * \frac{precision * recall}{(x * precision) + recall} \tag{3.7}$$

3.4. Aplicaciones

A lo largo de este capítulo hemos introducido y estudiado en cierta medida el concepto de anomalía y su proceso de detección, por tanto ahora estamos en posición de introducir algunas de sus aplicaciones y campos donde este proceso, toma especial relevancia, como podrían ser las siguientes:

- **Detección de intrusiones**: Siempre que un sistema esté conectado en red, corremos el riesgo de que un intruso acceda y realice acciones maliciosas. Actualmente se recuperan datos de los accesos y tráfico de red, a los que se aplican técnicas de detección de anomalías para detectar intrusos o comportamientos no comunes, para por ejemplo cerrar puertos tras su detección o bloquear IPs maliciosas.

- **Fraude en tarjetas de crédito**: La detección de anomalías se puede adaptar a este campo para detectar cuando el uso que se hace de una tarjeta de crédito no es real. Por ejemplo, no suelo pagar viajes con tarjeta y en un día realizo 3 compras de vuelos, a muy seguro, mi banco detectaría este comportamiento anómalo y me lo notificaría.

- **Internet de las cosas**: En el mundo del internet de las cosas, el principal componente son los sensores, que en algunos casos generan ingentes cantidades de datos por segundo de los cuales solo en raras ocasiones merecerán atención. En detectar estos momentos, es donde se centra la detección de anomalías en este área.

- **Ciencias de la naturaleza**: En lo que meteorología, climatología y estudios ecológicos respecta, el estudio de las anomalías se centra en revelar por ejemplo acciones humanas. Por ejemplo, la aparición de deforestación en imágenes por satélite donde debería haber árboles.

Al igual que se hizo con el clustering, se ha investigado acerca de artículos recientes sobre la detección de anomalías con el fin de comprender e ilustrar aplicaciones actuales de la materia. El primero interesante que hemos recuperado, dada la novedad de su aplicación, es el propuesto por Prado-Romero et al. [15] donde se aplican técnicas de detección de anomalías para determinar movimientos anómalos en el **bitcoin** entre distintas cuentas, intentando descubrir posibles acciones criminales, como la financiación del terrorismo. En temáticas más amables, encontramos el estudio de Zacher y Ryba [16], donde se aplican la detección de anomalías a detectar usos indebidos y problemas en el servidor de correo de una compañía.

Capítulo 4

Reglas de Asociación

Las reglas de asociación han sido una de las técnicas más estudiadas en el campo de la minería de datos. En este capítulo, se verá el concepto de regla de asociación y su trasfondo (sección 4.1), las medidas clásicas para su validación (sección 4.2), algunos de los principales algoritmos (sección 4.4) y finalizaremos estudiando algunas de sus aplicaciones (sección 4.6).

4.1. Introducción

La reglas de asociación dentro del ámbito de la informática no son muy distintas, al menos en el concepto general, de la búsqueda de relaciones en cualquier ámbito. Las reglas de asociación se enmarcan dentro del aprendizaje automático o minería de datos y no es algo nuevo sino que llevan siendo usadas y estudiadas desde mucho tiempo atrás, datando una de las primeras referencias a estas, del año 1993 [17]. Su utilidad es la de obtener conocimiento relevante de grandes bases de datos y se representan según la forma **X->Y** donde **X**, es un conjunto de ítems que representa el antecedente e **Y** un ítem consecuente, por ende, podemos concluir que los ítems **consecuentes** guardan una relación de co-ocurrencia con los ítems **antecedentes**. Esta relación puede ser obvia en algunos casos, pero en otros necesitará del uso de algoritmos de extracción de reglas de asociación que podrán desvelar relaciones no triviales y que puedan ser de mucho valor. Podremos presentar por tanto a las reglas de asociación, como un método de extracción de relaciones aparentemente ocultas entre ítems o elementos dentro de bases de

datos transaccionales, *datawarehouses* u otros tipos de almacenes de datos de los que es interesante extraer información de ayuda en el proceso de toma de decisiones de las organizaciones.

4.2. Validación

La forma clásica de medir la bondad o ajuste de las reglas de asociación a un determinado problema, vendrá dada por las medidas del soporte, la confianza y el lift, que podremos definir de la siguiente manera:

- Soporte: Se representa como *supp (X→Y)*, y representa la fracción de las transacciones que contiene tanto a X como a Y respecto al total de transacciones. Quedaría definido por la siguiente ecuación:

$$supp(X \rightarrow Y) = \frac{supp(X \cup Y)}{totaltransacciones} \tag{4.1}$$

- Confianza: Se representa como *conf (X→Y)*, y representa la fracción de transacciones en las que aparece el ítem Y, de entre aquellas transacciones donde aparece el ítem X. Su ecuación sería:

$$conf(X \rightarrow Y) = \frac{supp(X \rightarrow Y)}{supp(X)} \tag{4.2}$$

- Lift: El *lift*, es una medida útil para evaluar la independencia entre los ítems de una determinada regla de asociación. En una regla del tipo *conf (X→Y)*, esta medida representa el grado en que X tiende a ser frecuente cuando A está presente en la regla, o viceversa. El lift, quedará definido matemáticamente de la siguiente manera:

$$lift(X \rightarrow Y) = \frac{conf(X \rightarrow Y)}{supp(Y)} \tag{4.3}$$

Pese a que estas medidas son las más comunes y extendidas, hay innumerables propuestas de medias complementarias en la literatura, tales como la **convicción, factor de certeza, diferencia absoluta de confianza** entre otras muchas (figura 4.1).

#	Measure	Formula
1	ϕ-coefficient	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_k P(A_j,B_k)+\sum_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-\max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{AB})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{AB})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i)\log P(A_i), -\sum_j P(B_j)\log P(B_j))}$
8	J-Measure (J)	$\max\left(P(A,B)\log(\frac{P(B\|A)}{P(B)}) + P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}),\right.$ $\left. P(A,B)\log(\frac{P(A\|B)}{P(A)}) + P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})\right)$
9	Gini index (G)	$\max\Big(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A})[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2]$ $-P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B})[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2]$ $-P(A)^2-P(\overline{A})^2\Big)$
10	Support (s)	$P(A,B)$
11	Confidence (c)	$\max(P(B\|A), P(A\|B))$
12	Laplace (L)	$\max\left(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2}\right)$
13	Conviction (V)	$\max\left(\frac{P(A)P(\overline{B})}{P(A\overline{B})}, \frac{P(B)P(\overline{A})}{P(B\overline{A})}\right)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B)-P(A)P(B)$
17	Certainty factor (F)	$\max\left(\frac{P(B\|A)-P(B)}{1-P(B)}, \frac{P(A\|B)-P(A)}{1-P(A)}\right)$
18	Added Value (AV)	$\max(P(B\|A)-P(B), P(A\|B)-P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\overline{A}\,\overline{B})}{P(A)P(B)+P(A)P(\overline{B})} \times \frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{A}\,\overline{B})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\|A)-P(B), P(A\|B)-P(A))$

Figura 4.1: Distintas medidas aplicables a reglas de asociación.

4.3. Obtención de reglas

Si nos centramos en la manera de obtener las reglas, estas pueden abordarse desde dos perspectivas, solución por fuerza bruta (prohibitivo) o desde un enfoque basado en dos etapas. La primera de estas etapas es la generación de itemsets frecuentes, a partir de los cuales, en la segunda etapa se obtienen las reglas de asociación, que tendrán si todo ha ido correctamente un valor de confianza aceptable o elevado. La primera etapa de obtención de itemsets

frecuentes puede conllevar problemas de memoria ya que en una base de datos con muchos ítems o transacciones el número de estos será muy elevado, es por ello que surgen aproximaciones en el proceso de representación de itemsets frecuentes que nos permitirán obtener estos en bases de datos de gran tamaño. Estas aproximaciones son:

- Itemsets maximales: Son aquellos itemsets frecuentes para los que ninguno de los superconjuntos inmediatos al itemsets en cuestión, son frecuentes. A partir de estos podremos recuperar todos los itemsets frecuentes de manera sencilla sin tener que mantenerlos todos en memoria.
- Itemsets cerrados: Son aquellos itemsets frecuentes para los que ninguno de los superconjuntos inmediatos al itemsets en cuestión, tienen un soporte igual. Con esta aproximación, tendremos soportes e itemsets frecuentes que podremos recuperar fácilmente, aunque al ser más numerosos que los maximales mantenerlos en memoria puede llegar a ser complicado.

En resumen usaremos itemsets cerrados cuando la eficiencia sea un factor a tener en cuenta o prohibitivo, frente al tamaño de la base de datos. Si estuviéramos en el caso contrario, los itemsets maximales serán nuestra opción ganadora al ser más compactos. Sea como sea, una vez obtenidos los itemsets frecuentes podemos centrarnos en la obtención de las reglas para ello, se crean tordas las posibles combinaciones de regla con el itemset y se seleccionan solo aquellas que superen el umbral de confianza definido por el experto del problema en cuestión.

4.4. Principales algoritmos

En esta sección veremos una introducción a los principales algoritmos empleados en problemas de obtención de reglas de asociación.

4.4.1. Apriori

El algoritmo **Apriori**, fue propuesto por Agrawal y Srikant en 1994 [18] y desde entonces sigue siendo el algoritmo más extendido para la obtención

de itemsets frecuentes, con los que construiremos en una segunda etapa las reglas de asociación. Se basa en el principio de que si un itemset es frecuente, entonces todos sus subconjuntos también lo son por lo que al encontrar uno de estos, podremos podar el árbol de búsqueda evitando hacer comprobaciones y aumentando la eficiencia. Para obtener los itemsets frecuentes, el algoritmo en base a un valor mínimo de soporte fijado por el experto en la materia, generará todas las posibles combinaciones de itemsets y comprobará si son o no frecuentes. En cada iteración, se generan todos los posibles itemsets distintos que se pueden formar combinando los de la anterior, por lo que los itemsets irán creciendo de tamaño.

Apriori tiene bastantes factores o limitaciones relacionados con la eficiencia del algoritmo y que pueden afectar en gran medida al proceso de minería de datos que en algunos problemas específicos podría incluso resultar prohibitivo por tiempos o espacio. Algunas de estas limitaciones serían:

1. Soporte: Umbrales demasiado bajos conllevarán a una explosión del número de itemsets frecuentes lo que está directamente relacionado con una mayor necesidad de memoria y tiempo.

2. Número de ítems distintos: Esta limitación, está ligada a la necesidad del algoritmo apriori de almacenar el soporte de cada uno de éstos, lo que puede conllevar problemas de memoria.

3. Tamaño de la base de datos: Este punto está ligado, al anterior, pero en lugar de tener en cuenta los ítems individuales se tienen en cuenta el número de transacciones. Apriori al ser exhaustivo realiza múltiples pasadas por toda la base de datos por lo que el tiempo de ejecución puede ser muy elevado o incluso no llegar a acabar en varios días o semanas.

4. Longitud de las transacciones: Ligado al problema anterior, si las transacciones a su vez están formadas por muchos ítems, almacenar esto en memoria puede llegar a ser privativo e incluso imposible.

4.4.2. Eclat

El algoritmo Eclat [19], se basa en una estructura de datos denominada tid-list, que será generada para cada ítem en la base de datos y que almacena

los id de las distintas transacciones de la base de datos que contienen al ítem en cuestión. Este enfoque nos permite obtener el soporte de un k-ítemset de manera muy rápida realizando la intersección de sus subconjuntos. Por otro lado, mantener estas estructuras en memoria, puede llegar a ser imposible si la base de datos contiene muchas transacciones.

Las limitaciones de los algoritmos tradicionales han llevado a el estudio de otros método menos sensibles a los requisitos temporales o de espacio, de cara a poder aplicar estas técnicas a mayores cantidades de datos aún. Este método es el algoritmo FP-Growth y lo estudiaremos en el siguiente punto.

4.4.3. FP-Growth

El algoritmo **FP-Growth** [20] fue propuesto en el año 2000, como una solución a los problemas de memoria generados por los métodos típicos como el Apriori, visto anteriormente. Es un algoritmo muy eficiente y ampliamente extendido en problemas y soluciones que podrían ser enmarcados bajo el nombre de Big Data.

FP-Growth, crea un modelo comprimido de la base datos original utilizando una estructura de datos que denomina como ***FP-tree*** que está formada por dos elementos esenciales:

- Grafo de transacciones: Gracias a este grafo la base de datos completa puede abreviarse. En cada nodo, se describe un itemsets y su soporte que se calcula siguiendo el camino que va desde la raíz hasta el nodo en cuestión.
- Tabla cabecera: Es una tabla de listas de ítems. Es decir, para cada ítem, se crea una lista que enlaza nodos del grafo donde aparece.

Una vez se construye el árbol, utilizando un enfoque recursivo basado en divide y vencerás, se extraen los itemsets frecuentes. Para ello primero se obtienen el soporte de cada uno de los ítems que aparecen en la tabla de cabecera, tras lo cual, para cada uno de los ítems que superan el soporte mínimo se realizan los siguientes pasos:

1. Se extrae la sección del árbol donde aparece el ítem reajustando los valores de soporte de los ítems que aparecen en esa sección.

2. Considerando esa sección extraída, se crea un nuevo ***FP-tree***.

3. Se extraen los itemsets que superen el mínimo soporte de este último ***FP-tree*** creado.

En función a lo estudiado, es obvio ver que la memoria que ocupa es mucho menor que la generada por Apriori, así como al generar itemsets por medio del principio divide y vencerás, **FP-Growth** se presta a ser usado en entornos distribuidos como por ejemplo el entorno de Big Data, Apache Spark, aumentando sus prestaciones de manera notable.

4.5. Aspectos Avanzados

4.6. Aplicaciones

Las reglas de asociación son muy conocidas por sus aplicaciones en problemas como, el del análisis de la 'cesta de la compra'. Si bien, es verdad que esta puede ser su aplicación más extendida, las reglas de asociación tienen infinitud de aplicaciones en campos tan dispares como la obtención de información a partir de los datos recopilados por aerogeneradores, datos bancarios o logísticos. Dentro de la propia ciencia de datos, las reglas de asociación pueden usarse para extender otras vertientes, como el de la minería social o la minería de textos donde se usan para asociar la presencia de términos en ciertos documentos.

Al igual que se hizo en los capítulos anteriores de clustering y detección de anomalías, se han recopilado algunos artículos científicos recientes que tratan sobre reglas de asociación, para ilustrar ejemplos reales de aplicación de las mismas. El primer estudio es propuesto por Hu y Guo [21], donde usan un enfoque basado en reglas de asociación para la obtención de información relevante sobre el estado de la polución en el centro urbano de la ciudad de Lan-Xi-Yin. El segundo trabajo, es propuesto por Zhong [22] y está enfocado al ámbito del deporte donde por medio del algoritmo apriori, se analizan técnicas y tácticas en el baloncesto.

Bibliografía

[1] Juan Carlos Cubero y Amparo Vila Miranda. Clustering. *Transparencias de clase de teoría*. 2017-2018.

[2] Juan Carlos Cubero. Detección de anomalías. *Transparencias de clase de teoría*. 2017-2018.

[3] Jesús Alcalá Fernández. Reglas de Asociación: Introducción. *Transparencias de clase de teoría*. 2017-2018.

[4] MacQueen, J. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Volume 1: Statistics, 281-297

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)* .226-231.

[6] Zhang, T.; Ramakrishnan, R.; Livny, M. (1996). "BIRCH: an efficient data clustering method for very large databases". *Proceedings of the 1996 ACM SIGMOD international conference on Management of data - SIGMOD '96*. pp. 103-114

[7] Guha, Sudipto; Rastogi, Rajeev; Shim, Kyuseok (2001). ÇURE: An Efficient Clustering Algorithm for Large Databases". *Information Systems. 26*: 35?58

[8] S. Guha, R. Rastogi and K. Shim, ROCK: a robust clustering algorithm for categorical attributes,"*Proceedings 15th International Conference on Data Engineering* , Sydney, NSW, 1999, pp. 512-521.

[9] R. T. Ng and Jiawei Han, ÇLARANS: a method for clustering objects for spatial data mining,ïn *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 1003-1016, Sep/Oct 2002.

[10] Tan, Steinbach, Kumar. Introduction to Data Mining, chapter 8: Cluster Analysis: Basic Concepts and Algorithms.

[11] Moosavi, S.A. and Jalali, M. Community detection in online social networks using actions of users. 2014 *Iranian Conference on Intelligent Systems, ICIS*

[12] Baier D., Daniel I. Image Clustering for Marketing Purposes. In: Gaul W., Geyer-Schulz A., Schmidt-Thieme L., Kunze J. *Studies in Classification, Data Analysis, and Knowledge Organization*. Springer, Berlin, Heidelberg. 2012.

[13] Charu C. Aggarwal. 2013. Outlier Analysis. Springer Publishing Company, Incorporated.

[14] Kaggle Wiki https://www.kaggle.com/wiki/MeanFScore

[15] Prado-Romero M.A., Doerr C., Gago-Alonso A. (2018) Discovering Bitcoin Mixing Using Anomaly Detection. In: Mendoza M., Velastín S. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2017*. Lecture Notes in Computer Science, vol 10657

[16] Zacher, S. & Ryba, P. (2018). Anomaly detection in server metrics with use of one-sided median algorithm. *Journal of Applied Computer Science Methods*, 9(1), pp. 5-2

[17] Rakesh Agrawal, Tomasz Imieliski, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD* Rec. 22, 1993, 207-216.

[18] R. Agrawal and R. Srikant Fast algorithms for mining association rules in large databases. 1994. *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB, pp. 487-499.

[19] Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1997, August). New Algorithms for Fast Discovery of Association Rules. In *KDD* (Vol. 97, pp. 283-286).

[20] Han, J., Pei, H., Yin, Y.: Mining Frequent Patterns without Candidate Generation. 2000. *Proc. Conf. on the Management of Data* (SIGMOD 2000), Dallas, TX, pp. 1?12.

[21] HU, Q. L., & GUO, S. (2017). Mining on the Air Pollutants Association Rules of Lan-Xi-Yin Urban Agglomeration. DEStech Transactions on Environment, Energy and Earth Sciences, (epee)

[22] Zhong, X. A Study on Basketball Techniques and Tactics Based on Apriori Algorithm. Wireless Personal Communications, 1-10.