

Aprendizaje No Supervisado y Detección De Anomalías

Máster de Ciencia de datos e Ing. De Computadores - UGR

M^a Cristina Heredia Gómez

Índice general

I

Clustering y detección de anomalías

1	Clustering	6
1.1	Definición y tipos de clustering	6
1.1.1	Definición	6
1.1.2	Tipos	7
1.2	Aplicaciones	9
1.3	Algoritmos de Clustering	10
1.3.1	K-Medias	10
1.3.2	DBSCAN	12
1.4	Evaluación de Resultados	13
2	Detección de anomalías en datos	17
2.1	Introducción	17
2.1.1	Aplicaciones	17
2.1.2	Magnitud del problema	18
2.2	Métodos de detección de outliers	18
2.2.1	Salidas de los algoritmos	18
2.2.2	Métodos supervisados	19
2.2.3	Métodos semisupervisados	21
2.2.4	Métodos no supervisados	23
2.2.5	Evaluación	25

3	Reglas de Asociación: Aspectos básicos	27
3.1	Introducción	27
3.2	Aplicaciones	28
3.3	Medidas de las reglas de asociación	28
3.4	Métodos de extracción de reglas	30
3.4.1	Apriori	30
3.4.2	Eclat	31
3.4.3	FP-growth	31
3.5	Conjuntos maximales y cerrados	32
3.5.1	Itemsets maximales	32
3.5.2	Itemsets cerrados	33
3.6	Problemas	33
4	Reglas de asociación: Aspectos avanzados	34
4.1	Problemas de interpretabilidad	34
4.2	Medidas de calidad	34
4.2.1	Medidas objetivas	35
4.2.2	Medidas subjetivas	36
4.3	Interpretación	37
4.4	Reglas de asociación difusas	39
4.5	Reglas Jerárquicas	40
4.6	Análisis de las reglas por grupos	41
	Bibliografía	42
	Artículos	42
	Libros	42



Clustering y detección de anomalías

1	Clustering	6
1.1	Definición y tipos de clustering	
1.2	Aplicaciones	
1.3	Algoritmos de Clustering	
1.4	Evaluación de Resultados	
2	Detección de anomalías en datos	17
2.1	Introducción	
2.2	Métodos de detección de outliers	



1. Clustering

1.1 Definición y tipos de clustering

El clustering es el problema de agrupar un conjunto de objetos en grupos llamados clusters, según las similitudes que presenten esos objetos entre sí. Es una técnica empleada en muchos campos como el aprendizaje automático, recuperación de información en imágenes, bioinformática, estadística o sociología, entre otros. Si bien en los 70 el clustering se incluyó dentro del área de la inteligencia artificial dentro del aprendizaje no supervisado (datos cuya etiqueta de clase se desconoce), inicialmente en los 60 se incluyó dentro del ámbito del análisis de datos y la taxonomía numérica, aunque su origen se remonta al 1932 en el campo de la antropología. Para resolver el problema del clustering se han propuesto múltiples algoritmos en la literatura. Entre los enfoques más estudiados se encuentran buscar grupos tal que la distancia entre los miembros del cluster sea mínima, enfoques basados en encontrar las áreas más densas en el espacio de los datos, modelos de grafos o distribuciones estadísticas. A pesar de todas las propuestas, el problema de encontrar qué determina un cluster y cómo encontrar los mejores clusters sigue siendo un proceso difícil e iterativo, y actualmente no existe ningún algoritmo que obtenga un rendimiento mejor para todos los tipos de problemas ("no free lunch"), pues depende altamente de la naturaleza de los datos que se estén considerando.

1.1.1 Definición

Definición 1.1.1 — Clustering. Clasificación no supervisada (sin tener información previa acerca de la clase) de observaciones, datos o vectores de características en grupos denominados clusters, tal que los elementos de un mismo cluster sean similares entre sí y diferentes de los elementos de otros clusters.

Definición 1.1.2 — Centroide. En clustering, se denomina centroide al punto medio de cada cluster de elementos.

1.1.2 Tipos

La clasificación se divide principalmente en clasificación supervisada o no supervisada. Dentro de la segunda se engloban las técnicas de clustering. Las metodologías propuestas hasta ahora para el clustering pueden clasificarse en distintos tipos, como se puede observar en la siguiente figura:

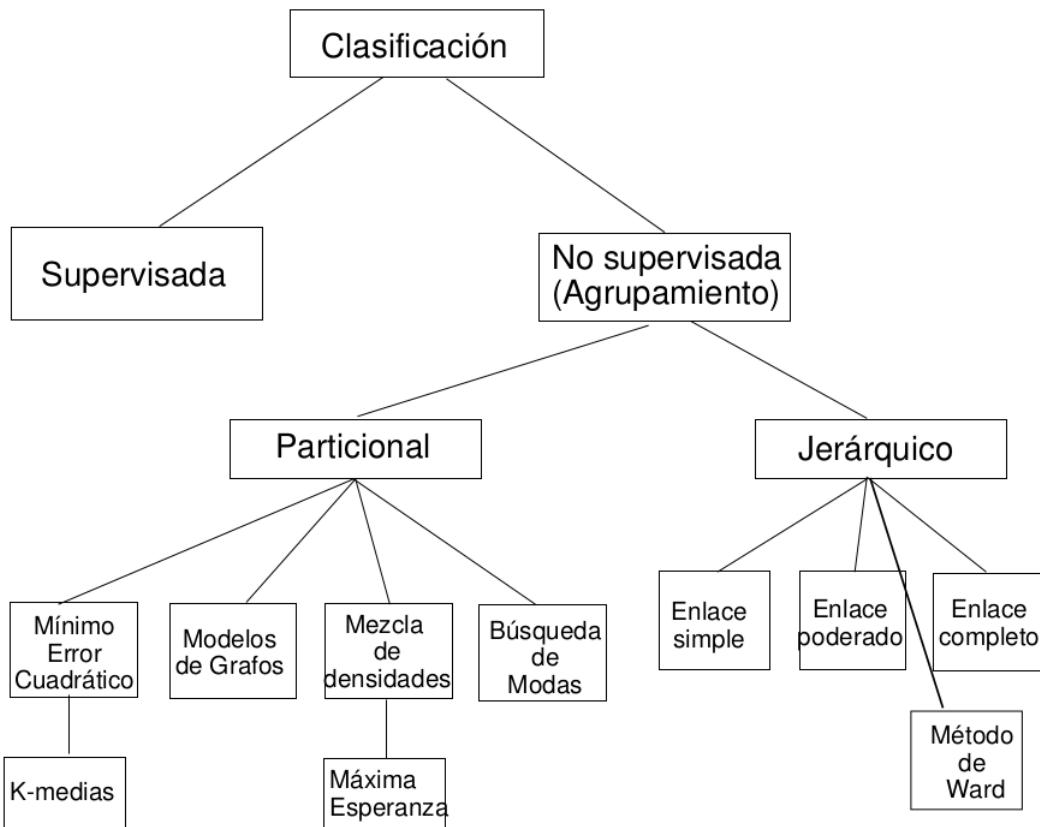


Figura 1.1: Tipos de clasificaciones de clasificación y de clustering

Supervisada y no supervisada en clasificación supervisada el problema a resolver es clasificar o agrupar un elemento etiquetado en función de sus características. Como los datos están etiquetados, se conoce a priori información sobre la clase o grupo al que pertenecen. Por el contrario, en un problema de clasificación no supervisada no se tiene información de las clases o grupos de los elementos, por lo que la tarea de clasificar resulta más compleja ya que hay que resolver problemas adicionales de la clasificación no supervisada, como qué número de clases es el más óptimo o qué elementos meter en cada clase.

Agrupamiento particional el agrupamiento particional se da cuando los clusters a obtener son disjuntos y cubren todo el conjunto de elementos. Se usan fundamentalmente en aplicaciones de ingeniería donde se necesitan particiones simples, dado que son muy útiles para trabajar con datasets de gran tamaño, donde la practicidad de los dendogramas se ve limitada. Dentro del agrupamiento particional, se han propuesto técnicas de naturaleza variada, en función de como se interpretan los elementos y los clusters.

Mínimo error cuadrático: las técnicas que aplican este enfoque, por ejemplo K-medias, emplean la suma total de la distancia de cada punto a su centroide, pretendiendo así

minimizar la distancia entre elementos de un mismo grupo y maximizar la distancia entre grupos.

Modelos de grafos: este enfoque representa los datos en forma de grafo, tal que los vértices son elementos y las aristas son las conexiones entre los mismos. Las aristas se definen por criterios de semejanza. Es una filosofía similar a la del KNN (K nearest neighbors).

Mezcla de densidades estas técnicas consideran un cluster como las regiones más densas del espacio de N-dimensiones separadas por regiones de poca densidad. La técnica de la máxima esperanza, sigue este enfoque a través de la búsqueda de la máxima similitud mediante un proceso iterativo que alterna el cálculo de la esperanza a través de una función y maximización de la misma.

Búsqueda de modas estos métodos siguen una filosofía similar al grupo anterior, ya que hacen uso de estimadores estadísticos de la densidad de probabilidad de cada grupo para formar los clusters.

Agrupamiento Jerárquico Como se define en [2], si consideramos que un cluster puede tener subclusters entonces el clustering jerárquico es un conjunto de cluster anidados en una estructura con forma de árbol llamada dendograma, en donde cada nodo del árbol representa un cluster y puede obtenerse como unión de sus hijos, que serán los subclusters. La raíz del árbol es el cluster que contiene todos los elementos. Dentro de las técnicas de agrupamiento jerárquico también hay enfoques basados en grafos, como los métodos de enlace simple, enlace ponderado y enlace complejo, y el método de Ward que se basa en la mínima varianza.

Enlace simple La proximidad entre clusters se define como la distancia mínima entre dos puntos cualesquiera de cada uno de los cluster. Los clusters se obtienen buscando las componentes conexas del grafo, para ello se comienza considerando cada punto como un cluster y se van añadiendo enlaces de uno en uno entre los puntos más cercanos, combinandolos luego en un cluster y finalizando cuando todos los vértices (elementos) están conectados.

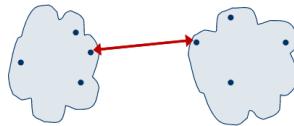


Figura 1.2: medida de distancia en clustering de enlace simple

Enlace complejo La proximidad entre clusters se define como la distancia máxima entre dos puntos cualesquiera de cada uno de los cluster. Los clusters se obtienen de nuevo considerando cada punto como un cluster y se van añadiendo enlaces de uno en uno, los más cortos primero, pero en este caso un grupo de puntos no es considerado un cluster hasta que todos los puntos de ese grupo están totalmente unidos (forman un clique).

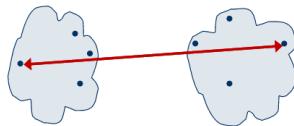


Figura 1.3: medida de distancia en clustering de enlace complejo

Enlace ponderado Es una técnica comprendida entre las técnicas de enlace simple y enlace complejo donde la proximidad entre clusters se define como:

$$\text{proximidad}(C_i, C_j) = \frac{\sum_{x \in C_i, y \in C_j} \text{proximidad}(x, y)}{m_i \cdot m_j}$$

Es decir, la distancia promedio de todos los puntos de un cluster con todos los puntos del otro cluster, donde m_i es el tamaño el cluster C_i y m_j es el tamaño del cluster C_j .

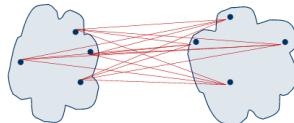


Figura 1.4: medida de distancia en clustering de enlace ponderado

Método de Ward Usa la misma función objetivo que el clustering con K-Medias, dado que se define la proximidad entre clusters como el incremento en el error cuadrático cuando se unen dos clusters. Las distancias iniciales entre dos clusters C_i, C_j se definen usando la distancia euclídea al cuadrado entre los puntos:

$$d_{ij} = d(\{X_i\}, \{X_j\}) = \|X_i - X_j\|^2$$

El método de Ward se puede implementar recursivamente usando algoritmos de **Lance–Williams**, representados por una fórmula recursiva que actualiza las distancias entre clusters en cada iteración, donde cada vez se une una pareja de clusters. Es decir, sean Q y R dos clusters, donde R se ha obtenido por unión de dos clusters A y B , se define la proximidad entre Q y R como:

$$p(R, Q) = \alpha_A p(A, Q) + \alpha_B p(B, Q) + \beta p(A, B) + \gamma |p(A, Q) - p(B, Q)|$$

Es decir, después de unir los clusters A y B formando otro cluster R , la proximidad de R a un cluster ya existente Q , es una función lineal de las proximidades de Q a los clusters originales A y B .

Basados en centroides Los métodos basados en centroides calculan la proximidad entre clusters calculando la distancia entre los centroides de los mismos. Los métodos basados en centroides tienen una peculiaridad llamada **inversión**, a diferencia de otros métodos de clustering.

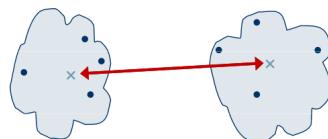


Figura 1.5: medida de distancia en clustering basado en centroides

Definición 1.1.3 — Inversión. Propiedad presente en los métodos de clustering basados en centroide donde dos clusters que han sido unidos serán menos distantes que el par de clusters unidos en la iteración anterior. En otros métodos la distancia entre clusters unidos se incrementa a medida que vamos pasando de clusters formados por un solo elemento a un cluster con todos los elementos.

1.2 Aplicaciones

Los humanos tenemos de forma inherente y natural la capacidad de dividir objetos en grupos y clasificarlos, por ejemplo, antes una foto de la calle sabríamos clasificar perfectamente cualquier objeto reflejado, como coches, árboles, casas, animales... Los algoritmos de clustering nos ayudan a entender mejor los datos y detectar las clases para poder hacer agrupaciones de nuestro interés.

En esta sección se revisan algunas de las aplicaciones reales de técnicas de clustering en áreas multidisciplinares.

- **Recuperación de Información:** para agrupar los resultados de búsqueda en varios clusters donde cada cluster capture un aspecto concreto de la búsqueda. Por ejemplo, en búsquedas de películas agrupar webs por valoraciones, trailers...
- **Predicción del clima:** para encontrar patrones en la presión atmosférica de regiones polares y áreas cuya influencia en el clima es alta.
- **Negocios:** cada vez más empresas, especialmente las relacionadas con la banca, usan los datos que poseen de sus clientes para hacer segmentación de clientes y orientar en base a los resultados sus campañas de marketing.
- **Psicología:** se ha usado clustering para identificar diferentes tipos de depresión en subcategorías.
- **Medicina:** se usa clustering para detectar patrones espaciales o temporales en cómo se distribuye o propaga una enfermedad. Por ejemplo, google usó sus búsquedas para estudiar la propagación de la gripe en EEUU.
- **Biología:** para analizar información genética a gran escala. Por ejemplo, se ha aplicado para agrupar genes diferentes con funciones similares.
- **Redes sociales:** para detectar comunidades de amigos y sugerir amigos y páginas de interés.
- **Química:** para encontrar similitudes en la estructura interna de los compuestos químicos.
- **Estudio de poblaciones:** por ejemplo, se puede usar clustering para identificar las áreas de una ciudad donde hay más incidencia de un tipo de crimen.
- **Robótica:** para rastrear objetos y detectar valores atípicos en los datos obtenidos de los sensores.
- **Actividad microbiana:** para analizar la resistencia a los antibióticos de distintos grupos de bacterias.
- **Buscadores:** para agrupar los resultados de búsqueda por temática.
- **Mapas temáticos:** para agrupar el consumo de la gente por zonas y orientar las ventas.

1.3 Algoritmos de Clustering

En la sección 1.1.2 se presentó una clasificación de las técnicas de clustering. Dentro de las técnicas de agrupamiento por particiones se encuentran algoritmos como k-Medias, PAM, CLARA/CLARANS y BFR. Entre las técnicas de clustering jerárquico se encuentran Diana/Agnes, BIRCH, CURE, ROCK, Chamaleon y ROCK entre otras, y en agrupamiento por densidades se encuentran los algoritmos DBSCAN, Optics y DenClue entre los más conocidos. En esta sección se explicarán algunas de las técnicas de mayor relevancia de entre las ya mencionadas.

1.3.1 K-Medias

Es un algoritmo de agrupamiento por particiones, donde dado un número de clusters dado previamente (K), los puntos se asignan al cluster cuyo centroide esté más cerca, usando una métrica de distancia. Es un problema computacionalmente complejo (NP-duro), que consta de un proceso de inicialización y un proceso iterativo mediante el cual va recalculando el centroide de cada cluster en función de los puntos que se vayan asignando a cada cluster. Cuando los centroides no cambian, el proceso se detiene.

Procedimiento

El algoritmo consta de dos etapas que se repiten hasta que el algoritmo se detiene: la etapa de asignación y la etapa de actualización. Durante la **etapa de asignación** se calculan las distancias de todos los puntos a los K centroides, y se asigna cada observación al cluster más cercano, que será aquel cuya distancia media sea la más cercana (tenga la distancia euclídea al mínimo cuadrado).

Durante la **etapa de actualización** se recalcula el centroide de cada cluster, obteniendo K nuevos centroides. Para recalcular los centroides (m_i) se buscan seleccionan los valores m_i que minimizan el SSE:

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} d^2(m_i, x)$$

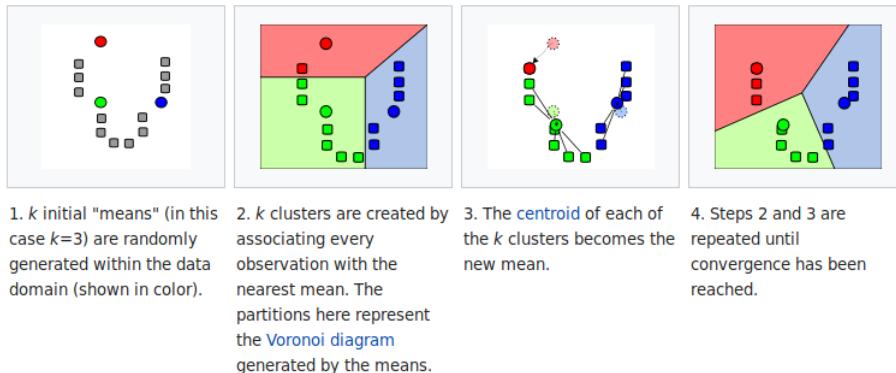


Figura 1.6: Procedimiento estándar del K-Medias, foto tomada de [K-means wiki](#)

Inicialización

El primer paso del algoritmo consiste en asignar los puntos a los centroides iniciales, para lo que hay que seleccionar K centroides previamente. El procedimiento menos sofisticado consiste en hacer la selección inicial de los centroides de forma aleatoria, pero existen otros procedimientos de selección inicial de centroides como el método de la partición aleatoria o K-means++.

Definición 1.3.1 — Método de la partición aleatoria. Comienza asignando aleatoriamente un cluster a cada observación y luego pasa a la **etapa de actualización** calculando la media inicial de los puntos para recolocar el centroide del cluster en ese lugar.

Definición 1.3.2 — Método K-means++. Se selecciona el centroide del primer cluster de forma aleatoria sobre los puntos a agrupar, y los demás centroides se seleccionan de entre los puntos restantes con probabilidad proporcional a su distancia al cuadrado al centroide (ya existente) más cercano.

Medidas de distancia

Los puntos se asignan al cluster más cercano, para lo que hay que emplear alguna medida de distancia. Las medidas de distancia más comunes empleadas en K-medias son:

- Distancia euclídea: cuando se utiliza esta distancia el SSE se minimiza utilizando la media aritmética por cada variable.
- Distancia de Manhattan: cuando se utiliza esta distancia el SSE se minimiza usando la mediana.

Problemas

El algoritmo de las K-Medias es un algoritmo de clustering sencillo, sin embargo cuenta con varios problemas:

- P Los resultados de K-Medias dependen mucho de la elección inicial que se haga de los centroides.

- P** K-Medias no garantiza encontrar la mejor solución, es decir, el algoritmo no garantiza que los centroides obtenidos sean los que minimizan globalmente al SSE, dado que los calcula en cada iteración usando la media aritmética.

Elegir el nivel de clusters (K) a priori, sin conocer cuantas agrupaciones pueden existir.

- P** Es sensible a outliers, dado que usa la media para calcular los centroides.

Definición 1.3.3 — Outliers. Los outliers son valores anómalos en los datos introducidos por algún error o variabilidad.

- P** El manejo de atributos no numérico también supone un problema en K-Medias, dado que se basa en distancias.

- P** K-Medias no funciona bien cuando los clusters son de diferente tamaño, no convexos o con diferente densidad.

1.3.2 DBSCAN

Clustering espacial basado en la densidad de aplicaciones con ruido (DBSCAN) es un algoritmo que detecta regiones densas de puntos separadas por regiones poco densas. Precisa de dos parámetros:

- ε : que denota la densidad, es decir, el número de puntos en un radio específico.
- **minPts**: denota el mínimo número de puntos que determinan una región densa.

El algoritmo comienza desde un punto arbitrario que no haya sido visitado previamente, se obtienen sus ε vecinos y se comprueba si dicha vecindad contiene suficientes puntos para formar un cluster. Si los contiene, se forma un cluster, sino el punto en cuestión se etiqueta como ruido, aunque luego si está dentro del vecindario de otro punto que cumpla el criterio podrá formar parte de un cluster. Si un punto pertenece a la parte densa de un cluster, entonces su vecindario también será parte de ese cluster. Este proceso iterativo se repite hasta que el cluster con alta densidad de puntos se encuentra completamente.

Procedimiento

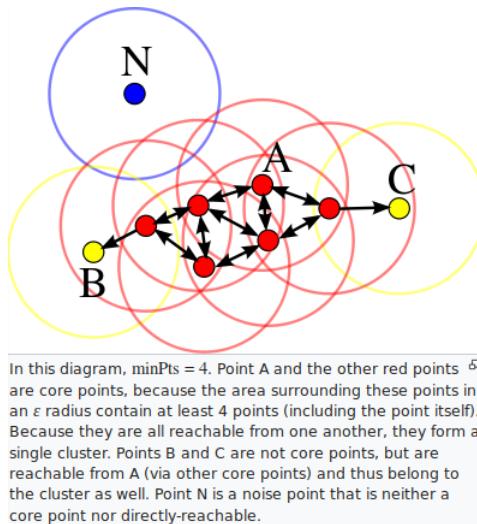
El algoritmo de DBSCAN se compone de tres pasos: el primer paso consiste en **encontrar los ε vecinos de cada punto** e identificar los puntos core que tengan más de **minPts** vecinos. El segundo paso consiste en **encontrar las componentes conexas de puntos core en el vecindario**, buscando las componentes conexas en el grafo del vecindario, ignorando puntos que no sean core. El último paso consiste en **asignar los puntos que no son core a algún cluster**. Para ello, para cada punto no core se le asigna a un cluster si éste está en el radio ε y sino se etiqueta como ruido.

Definición 1.3.4 — Puntos core. Los puntos core son aquellos puntos que tienen al menos **minPts** vecinos en un radio de ε , y por tanto serán puntos contenidos en un cluster.

Definición 1.3.5 — Puntos frontera. Los puntos frontera son aquellos puntos que pertenecen al vecindario de algún punto core pero que tienen menos de **minPts** vecinos en un radio de ε .

Inicialización

El punto por el que se comienza a explorar el vecindario se selecciona de forma arbitraria.

Figura 1.7: Procedimiento BDSCAN, foto tomada de [DBSCAN wiki](#)

Medidas de distancia

DBSCAN se puede usar con cualquier medida de distancia o función de similaridad, siendo la más común la distancia euclídea.

Problemas

Los principales problemas de DBSCAN son:

- P No es un algoritmo completamente determinístico, ya que a veces se da que algunos puntos frontera son accesibles desde más de un cluster, pudiendo ser parte de más de un cluster.
- P La calidad de los resultados del algoritmo depende de la medida de distancia que se use, pudiendo truncar completamente los resultados si la medida elegida no es la más adecuada cuando los datos presentan alta dimensionalidad.
- P El parámetro **minPts** no puede ser bien estimado para todos los clusters cuando los datos presentan grandes diferencias de densidades.
- P Es difícil estimar correctamente el parámetro ϵ .

1.4 Evaluación de Resultados

Las medidas de evaluación de clusters que se aplican para validar los resultados de clustering se clasifican según [2] en los siguientes tipos:

No supervisado Las medidas para validar clustering no supervisado se conocen con el nombre de índices internos, dado que solo pueden usar información del dataset. Estos índices tratan de evaluar factores como cuál es el mejor número de clusters (K), para lo que se puede usar o bien la suma de mínimos cuadrados (SSE) para estimar K o bien hacer clustering jerárquico sobre una muestra. Para evaluar como de buenos son los clusters se emplean dos medidas:

- medidas de cohesión: determinan como de cerca están los elementos de un cluster.

■ medidas de separación: determinan como de bien separados están los clusters entre sí. Tanto la cohesión como la separación se pueden medir o bien sin usar centroides (enfoque basado en grafos) o bien usando centroides (enfoque basado en prototipo).

1. Enfoque basado en grafos: No se emplean centroides para medir la cohesión ni la separación.

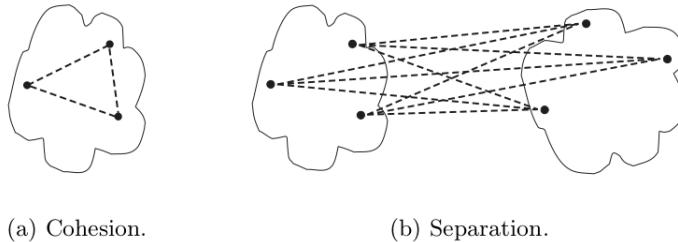


Figura 1.8: (a) Cohesión, (b) Separación sin centroides, foto de [2]

Definición 1.4.1 — Cohesión (enfoque basado en grafos). Se define la cohesión de un cluster como la suma de los pesos de los enlaces que conectan puntos con el cluster considerado.

$$\text{cohesion}(C_i) = \sum_{x \in C_i, y \in C_i} \text{proximidad}(x, y)$$

Definición 1.4.2 — Separación (enfoque basado en grafos). Se define la separación entre dos clusters como la suma de los pesos de los enlaces de los puntos de un cluster a los puntos del otro cluster.

$$\text{separacion}(C_i, C_j) = \sum_{x \in C_i, y \in C_j} \text{proximidad}(x, y)$$

2. Enfoque basado en prototipo: Se emplean centroides para medir la cohesión y la separación.

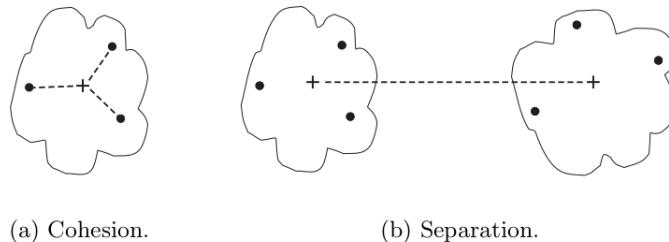


Figura 1.9: (a) Cohesión, (b) Separación usando centroides, foto de [2]

Definición 1.4.3 — Cohesión (enfoque basado en prototipo). Se define la cohesión de un cluster como la suma de las proximidades con respecto al centroide del cluster (prototipo).

$$\text{cohesion}(C_i) = \sum_{x \in C_i} \text{proximidad}(x, c_i)$$

Definición 1.4.4 — Separación (enfoque basado en prototipo). Se define la separación entre dos clusters como la proximidad entre los prototipos (centroideos) de los cluster.

$$\text{separacion}(C_i, C_j) = \text{proximidad}(c_i, c_j)$$

$$\text{separacion}(C_i) = \text{proximidad}(c_i, c)$$

Hay dos medidas de separación dado que la separación del prototipo de un cluster a un prototipo general está relacionada directamente con la separación entre prototipos de un cluster a otro.

Hasta ahora se ha mencionado como evaluar un conjunto de clusters, sin embargo, existen formas de validar los clusters de forma individual, sin considerar todo el grupo. Para ello se emplean técnicas como el coeficiente de silueta o la matriz de proximidad.

- **Coeficiente de silueta** Combina cohesión y separación. Se puede calcular el coeficiente de silueta medio de un cluster haciendo la media de los coeficientes de silueta de los puntos de dicho cluster. El coeficiente de silueta tiene un valor entre [-1,1], aunque el cluster será bueno solo si el valor del coeficiente es positivo y cercano a 0. El proceso para calcular el coeficiente de silueta consta de tres pasos:

1. para cada objeto i -ésimo calcula su distancia media a todos los demás objetos del cluster (a_i).
2. para cada objeto i -ésimo y cualquier cluster que no contenga el objeto, se calcula la distancia media del objeto a todos los objetos del cluster dado. Encuentra así el mínimo con respecto a todos los cluster (b_i).
3. para cada objeto i -ésimo se calcula el coeficiente de silueta como:

$$s_i = (b_i - a_i) / \max(a_i, b_i)$$

- **Matriz de proximidad** Un cluster ideal es aquel cuyos puntos tienen una similitud de 1 con puntos de su mismo cluster y 0 con puntos de otros clusters. Si ordenamos las filas y columnas de la matriz de similitud tal que todos los elementos de una misma clase estén juntos, una matriz de similitud ideal sería aquella que tenga una diagonal con estructuras de bloques. Una alta correlación entre la matriz ideal y la matriz actual de similitud indica que los puntos que pertenecen al mismo cluster están cercanos unos a otros, mientras que una correlación baja indica lo contrario. No es buena medida para

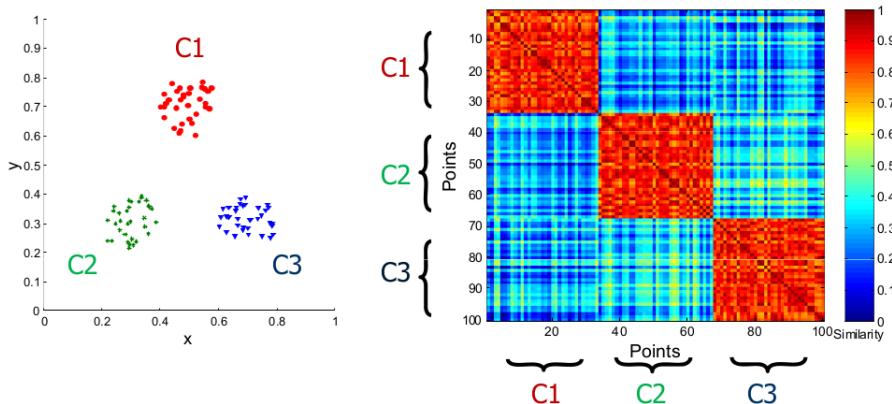


Figura 1.10: Índice interno: matriz de similitud

algunos clusters densos o contiguos, dado que no son globulares y algunos están muy entrelazados con otros grupos.

Supervisado Se conocen como índices externos dado que usan información no presente en el dataset, como la información de la clase. Para ello se procede a hacer un clustering eliminando el atributo de la clase, y luego se comparan los resultados de los clusters obtenidos, comprobando si en cada cluster hay una mayoría de valores que coinciden con los valores de clase que ya se conocían. La entropía y la precisión son ejemplos de índices externos.

Relativo Comparan diferentes clusterings. No son una forma particular de evaluación de clusters, sino un uso específico de medidas ya conocidas. Por ejemplo, usar la entropía para comparar dos K-Medias.



2. Detección de anomalías en datos

2.1 Introducción

La detección de anomalías en datos consiste en analizar aquellos datos de un conjunto de datos que sean considerablemente distintos del resto de datos. La fase de análisis es crucial dado que una anomalía puede deberse a un error en la lectura de los datos, por ejemplo un error introducido por los sensores, o pueden representar datos reales, por ejemplo un valor anómalo en datos médicos. En el primer caso, al ser la anomalía un error introducido por un sensor, bastaría con eliminar ese punto del conjunto de datos, pues estaría introduciendo ruido y ninguna información útil. Sin embargo, en el segundo caso, una anomalía en datos médicos puede ser muy relevante, por ejemplo, la aparición de un tumor en una foto es un valor anómalo que, sin embargo, es muy importante y no conviene eliminar. Es por esto que un buen análisis de detección de anomalías que nos permita discernir entre ruido o valor anómalo es fundamental.

2.1.1 Aplicaciones

Algunas de las aplicaciones reales de la detección de anomalías mencionadas en [1] son:

Sistemas de detección de intrusos analizan el tráfico de red y otros datos de monitoreo del sistema para detectar comportamientos inusuales y maliciosos, como por ejemplo, un ataque DDoS.

Fraude de tarjetas de crédito Los usos no autorizados de tarjetas de crédito muestran patrones diferentes a los usuales, que pueden ser usados para detectar anomalías en las transacciones con tarjeta.

Diagnóstico médico Se pueden aplicar técnicas de detección de outliers a los datos recogidos de escáneres y sensores para encontrar enfermedades.

Cumplimiento de la ley Determinar fraudes en transacciones financieras, inversiones en bolsa, reclamos a seguros etc requiere encontrar patrones inusuales entre los datos.

Ciencias de la tierra Desde los satélites se recogen miles de datos espaciotemporales de la tierra como los cambios de clima. Encontrar anomalías en esos datos dan mucha información acerca de tendencias humanas medioambientales o humanas que se estén generando y estén llevando a dichos cambios.

2.1.2 Magnitud del problema

En todas las aplicaciones mencionadas anteriormente, los datos presentan un modelo "normal", por lo que las anomalías se reconocen como desviaciones de dicho modelo. En muchos casos, las anomalías se descubren como múltiples puntos en lugar de un solo punto anómalo. Algunas técnicas de aprendizaje son menos sensibles a valores anómalos que otras, por ejemplo, algoritmos de clasificación como C4.5 clasifican muy bien las clases mayoritarias, por lo que la clasificación de puntos de las clases mayoritarias no se verá afectada por otros valores poco frecuentes, como los valores anómalos. Otro ejemplo de técnicas robustas son las reglas de asociación, ya que al basarse en medidas de confianza son muy robustas a excepciones (anomalías).

Sin embargo, algunas técnicas de regresión, clustering, test de hipótesis y cualquier otro método basado en medias será muy sensible a valores anormales. Por ejemplo, en clustering, los centroides de los clusters pueden variar mucho a causa de englobar valores anómalos en un determinado cluster. Por lo tanto, se puede concluir la importancia de un análisis detallado antes de eliminar las anomalías.

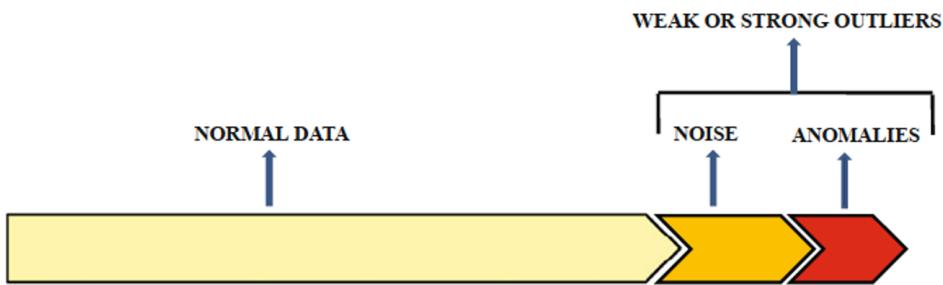


Figura 2.1: Espectro de datos normales a outliers, figura sacada de [1]

2.2 Métodos de detección de outliers

2.2.1 Salidas de los algoritmos

Las salidas o resultados de los algoritmos de detección de outliers pueden ser de dos tipos:

Puntuación Es una forma muy general de salida, donde el algoritmo devuelve como salida una puntuación sobre el nivel de anomalía de un punto del conjunto de datos. Esto es usado para determinar un ranking de los puntos en función de su nivel de anomalía.

Etiqueta El algoritmo devuelve como salida una etiqueta binaria indicando si el punto del conjunto de datos es o no un outlier. Las puntuaciones mencionadas arriba también se pueden convertir en etiquetas binarias mediante el uso de umbrales que se imponen en ciertas puntuaciones. La etiqueta binaria contiene menos información que el mecanismo de puntuación pero da más información útil.

Incluyen las etiquetas de clase para entrenar el modelo que determine cierto tipo de anomalías. Están especialmente diseñados para detección de anomalías, más que para eliminación de ruido, dado que asumen que las etiquetas representan la información que se está buscando, en lugar de ruido o errores.

En la bibliografía, algunos autores se refieren al ruido y a las anomalías como **outliers débiles** y **outliers fuertes**. La detección y eliminación de ruido es necesaria dado que conllevará la obtención de un conjunto de datos más limpio para posteriormente aplicarle algoritmos de aprendizaje automático. Para encontrar las anomalías y diferenciarlas del ruido en los datos, es necesario aprender un modelo que distinga los datos normales de los datos anómalos. En los siguientes apartados se revisarán técnicas supervisadas, no supervisadas y semi-supervisadas para detección de anomalías.

2.2.2 Métodos supervisados

Las técnicas de detección de outliers supervisadas son más eficientes en varias aplicaciones específicas, dado que las características de muestras previas de los datos pueden usarse para afinar el proceso de búsqueda hacia los valores atípicos más relevantes, construyendo un modelo de clasificación que incluye información de la clase de la anomalía. En los métodos supervisados, los datos de entrenamiento están etiquetados indicando, para cada instancia, si esta es una anomalía o no.

Tid	SrcIP	Start time	Dest IP	Dest Port	Number of bytes	Attack
1	206.135.38.95	11:07:20	160.94.179.223	139	192	No
2	206.163.37.95	11:13:56	160.94.179.219	139	195	No
3	206.163.37.95	11:14:29	160.94.179.217	139	180	No
4	206.163.37.95	11:14:30	160.94.179.255	139	199	No
5	206.163.37.95	11:14:32	160.94.179.254	139	19	Yes
6	206.163.37.95	11:14:35	160.94.179.253	139	177	No
7	206.163.37.95	11:14:36	160.94.179.252	139	172	No
8	206.163.37.95	11:14:38	160.94.179.251	139	285	Yes
9	206.163.37.95	11:14:41	160.94.179.250	139	195	No
10	206.163.37.95	11:14:44	160.94.179.249	139	163	Yes

Figura 2.2: Aprendizaje Supervisado: los datos de train incluyen etiquetas de clase

En el ejemplo anterior la última columna representa las etiquetas de clase, que dan información sobre si los accesos registrados a un sistema fueron ataques (valores anómalos) o no.

Algunos modelos de clasificación supervisados son los conocidos árboles de decisión, redes bayesianas o SVMs.

Clasificación desbalanceada la clasificación desbalanceada es un problema muy común en análisis de outliers, pues normalmente los outliers que realmente representan datos anómalos en lugar de ruido son datos con etiquetas de clase poco frecuentes. Esto supone un problema, dado que algunos clasificadores se centran en clasificar correctamente la clase mayoritaria, mientras que ignoran a la minoritaria, a pesar de la relevancia de esta última. Por ejemplo: en un sistema de análisis de imágenes médicas de pacientes tenemos dos clases normal (99.9 %) y anómalo (0.1 %). Un clasificador como C4.5 por ejemplo que etiquete cada valor nuevo de entrada como "normal" tendrá un acierto del 99.9 %, sin embargo no será fiable pues estará obviando información muy importante, pues una anomalía en esas imágenes puede suponer la existencia de un nódulo maligno o tumor. Existen dos enfoques para manejar el problema de la clasificación supervisada cuando existen clases minoritarias:

- **Métodos basados en instancia** Estos métodos modifican el conjunto de datos de entrenamiento de forma previa a entrenar el algoritmo de clasificación para intentar solucionar el problema del desbalanceo. Para ello aplican técnicas de Undersampling u Oversampling.

Definición 2.2.1 — Undersampling. técnica mediante la cual se eliminan instancias de las clases mayoritarias para igualar o aproximar el número de instancias que

contiene la clase minoritaria.

Definición 2.2.2 — Oversampling. técnica mediante la cual se generan instancias de la clase minoritaria para igualar o aproximar el número de instancias que contienen las clases mayoritarias.

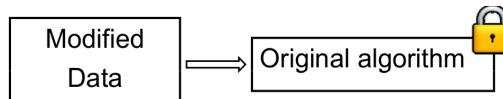


Figura 2.3: Métodos basados en instancia

Algunas de las técnicas más famosas de undersampling son:

- **Tomek-links:** Se generan los pares de instancias positivas-negativas de distancia mínima, y se eliminan las instancias de la clase mayoritaria.

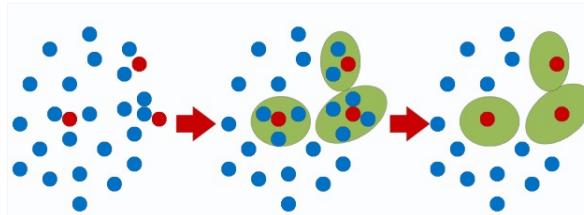


Figura 2.4: Pasos del algoritmo Tomek-links, foto de [slidsharecdn](#)

- **CNN (vecino más cercano condensado):** Selecciona todas las instancias con clase positiva y sólo una instancia de clase negativa de forma aleatoria, generando un nuevo conjunto de datos D' . Luego usa el conjunto de datos original como datos de test y D' como conjunto de datos de entrenamiento, clasificando las instancias del conjunto original usando un 1NN. Finalmente, añade a D' las instancias que no hayan sido clasificadas correctamente.
- **NCL (regla de limpieza del vecindario):** método híbrido donde el conjunto de datos final se obtiene mediante un undersampling que combina Tomek links y CNN.

Algunas de las técnicas más conocidas de oversampling, son SMOTE y sus variantes:

- **SMOTE:** Para una instancia de clase minoritaria se seleccionan k vecinos minoritarios de forma aleatoria, y se genera un nuevo conjunto de datos en la línea entre la instancia seleccionada y los vecinos seleccionados. SMOTE también realiza un

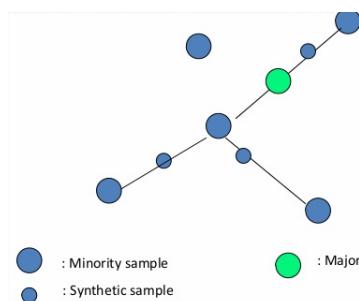


Figura 2.5: Algoritmo SMOTE, foto de [slidsharecdn](#)

proceso de undersampling sobre la clase mayoritaria, además del oversampling

sobre la minoritaria.

- **Métodos basados en algoritmos** Estos métodos no alteran el conjunto de datos sobre el que se va a aprender el modelo, pero en su lugar van asignando pesos a las instancias dentro del modelo de aprendizaje.

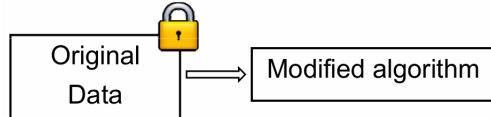


Figura 2.6: Métodos basados en algoritmo

Algunos métodos basados en algoritmo, son:

- Métodos sensibles al coste: que asignan costes más altos a los valores de clase minoritarios, para darles más peso y evitar que el algoritmo de aprendizaje las ignore.
 - Bagging: Incluye más instancias de clase minoritaria en cada paso del algoritmo de bagging.
 - Boosting: Asigna más peso a la clase minoritaria en cada iteración del algoritmo de boosting.
 - Adaptaciones específicas de métodos basados en reglas, redes neuronales o SVMs.
 - Métodos híbridos: como SMOTE+Boosting o SMOTE+Bagging.

2.2.3 M tododos semisupervisados

En muchos casos reales, los métodos supervisados no son aplicables dado que no se dispone de información sobre la etiqueta de los datos anómalos.

Tid	SrcIP	Start time	Dest IP	Dest Port	Number of bytes	Attack
1	206.135.38.95	11:07:20	160.94.179.223	139	192	No
2	206.163.37.95	11:13:56	160.94.179.219	139	195	No
3	206.163.37.95	11:14:29	160.94.179.217	139	180	No
4	206.163.37.95	11:14:30	160.94.179.255	139	199	No
5	206.163.37.95	11:14:32	160.94.179.254	139	19	Yes
6	206.163.37.95	11:14:35	160.94.179.253	139	177	No
7	206.163.37.95	11:14:36	160.94.179.252	139	172	No
8	206.163.37.95	11:14:38	160.94.179.251	139	285	Yes
9	206.163.37.95	11:14:41	160.94.179.250	139	195	No
10	206.163.37.95	11:14:44	160.94.179.249	139	163	Yes

Figura 2.7: Aprendizaje semi-supervisado: los datos de train no incluyen etiquetas de clase sobre las anomalías

En estos casos, se construye un modelo para los datos normales (los que no son anomalías) ya que de ellos sí se conoce su clase, y se considerará un nuevo dato entrante anómalo si no coincide

con el perfil aprendido (si es cualquier otra cosa). Para modelar los datos normales, se usan métodos de clasificación, reglas de asociación, SVMs entre otros métodos.

- **Basados en clasificación:** Hay un modelo de clasificación disponible pero no se conoce la clase de los atributos anómalos. Así que, cuando un nuevo punto llega al modelo, si el modelo se equivoca clasificándolo, el punto se considera una anomalía. Por ejemplo, un sistema de reglas de clasificación, por ejemplo, RIPPER, devolvería un valor de anomalía para cada nuevo punto entrante. El principal problema de este enfoque es que se generan muchos falsos positivos (considerar anomalías a puntos que no lo son). Otros métodos de clasificación están basados en métodos bayesianos o modelos de markov, que tratan de modelar la clasificación de los datos a través de probabilidades dadas por redes bayesianas y máquinas de estado finitas.
- **Basados en reglas:** tratan de encontrar patrones frecuentes en los datos modelando reglas de asociación.

Definición 2.2.3 — Soporte. El soporte de un conjunto de ítems X en un conjunto de datos D es la proporción de transacciones en la base de datos que contiene dicho conjunto de ítems.

$$Sop(X) = \frac{|X|}{|D|}$$

Definición 2.2.4 — Confianza. La confianza de una regla se define como:

$$Conf(X \Rightarrow Y) = \frac{sop(X \cup Y)}{sop(X)}$$

es decir, el soporte de que el antecedente y consecuente ocurran juntos entre el soporte del antecedente.

Un método basado en reglas para detección de anomalías es **LERAD**, que genera reglas del tipo $U \rightarrow W$ con $P(\neg W|U)$ baja. Para estimar la probabilidad ($P(\neg W|U)$) se usa $p = r/n$, donde r es el soporte y n es el número de consecuentes.

- **Basados en núcleo:** transforman los datos a un espacio de dimensión mayor, para así encontrar una función que haga los datos separables. Se pueden usar núcleos lineales, pero también radiales o con otras formas para obtener ajustes más flexibles. Se asume que solo hay una clase normal (de datos no anómalos), y se construye un modelo para esos datos determinando una región de separación en el plano. Un punto será considerado anomalía si queda fuera de la región determinada por el plano.

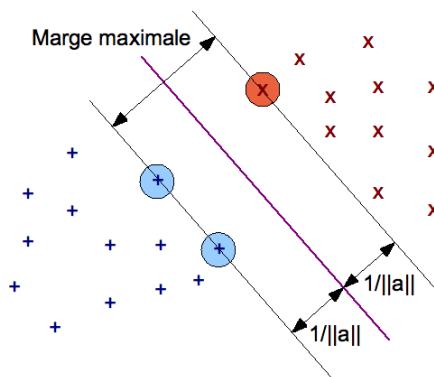


Figura 2.8: SVM con kernel lineal,foto de strasbg.fr

- **Basados en datos históricos:** Consideran las anomalías como eventos que no han ocurrido o no han sido observados en el pasado, por lo que compara la cuenta de eventos ocurridos actualmente con los ocurridos en el pasado para detectar las anomalías. Para esto se necesitan tres etapas:

1. Seleccionar las características más prometedoras
2. Contar las observaciones de cada atributo en un momento concreto del tiempo
3. Comparar el conteo realizado con datos históricos

2.2.4 Métodos no supervisados

A diferencia de los supervisados y semi-supervisados, estos métodos se aplican cuando los datos de entrada contienen valores anómalos pero estos no están etiquetados.

Tid	SrcIP	Start time	Dest IP	Dest Port	Number of bytes
1	206.135.38.95	11:07:20	160.94.179.223	139	192
2	206.163.37.95	11:13:56	160.94.179.219	139	195
3	206.163.37.95	11:14:29	160.94.179.217	139	180
4	206.163.37.95	11:14:30	160.94.179.255	139	199
5	206.163.37.95	11:14:32	160.94.179.254	139	19
6	206.163.37.95	11:14:35	160.94.179.253	139	177
7	206.163.37.95	11:14:36	160.94.179.252	139	172
8	206.163.37.95	11:14:38	160.94.179.251	139	285
9	206.163.37.95	11:14:41	160.94.179.250	139	195
10	206.163.37.95	11:14:44	160.94.179.249	139	163

Figura 2.9: Aprendizaje no supervisado: los datos de entrenamiento contienen anomalías y no están etiquetados

El procedimiento para clasificar es el siguiente: dado un nuevo punto de entrada, se le considerará anomalía o no según la relación que guarde con el resto de datos. Algunos enfoques para detección de anomalías no supervisada, son:

- Enfoques gráficos: Se trata de inspeccionar gráficamente un conjunto de datos para determinar que puntos son anomalías, por ejemplo mediante Boxplots, biplots, etc. Presenta el inconveniente de que puede demandar mucho tiempo y la interpretación del gráfico quedará sujeto a sugerión.
- Enfoques basados en estadísticos: Se aplican test estadísticos para decidir si una muestra es o no una anomalía. Existen modelos no paramétricos, que aplican propiedades de una distribución normal a una distribución no normal, y modelos paramétricos, que son los que asumen que existe un modelo paramétrico que describe los datos, por ejemplo, una distribución normal. En función del número de valores anómalos, si la distribución normal es univariante se pueden usar los test de Grubb, Moore o Rosner. El último es especialmente útil cuando hay menos de K outliers, dado que maneja los errores acumulados penalizando el α (0.05) y ajustando el FWER. Si la distribución es normal multivariante, se pueden usar métodos poco robustos, como la distancia de Mahalanobis, o métodos más robustos, como la distancia de Mahalanobis con MCD y la mediana, en lugar de las medidas de media y

covarianza, dado que la media tira mucho si hay valores extremos (es poco robusta) y la covarianza no funciona bien cuando hay outliers presentes.

Sin embargo, para controlar el error FWER es importante aplicar test estadísticos como el de Bonferroni o el test de Holm. Este tipo de enfoque está muy limitado dado que en la mayoría de los datos reales, éstos no presentan una distribución normal o una baja dimensionalidad.

- Enfoques basados en distancia: Por ejemplo, enfoques basados en la técnica del vecino más cercano o clustering, que utilizan una medida de distancia concreta a cada par de instancias de datos para discriminar entre anomalías y datos normales.

Vecinos cercanos Dado un conjunto de datos, un punto de dicho conjunto (t), y una medida de distancia, el método asignará un valor de anomalía (z) al punto (t) según cual sea la distancia de (t) a otros puntos del conjunto. Los dos enfoques principales son:

1. Basado en el vecino más cercano: la dinámica del algoritmo consiste en calcular la distancia entre cada par de puntos, encontrar un número (K) que sea el k -ésimo punto más cercano a otro punto, para cada punto (P) calcular su valor de outlier como la distancia del punto (P) a sus K vecinos más cercanos, y considerar como outliers los puntos con mayor puntuación de outlier obtenida.

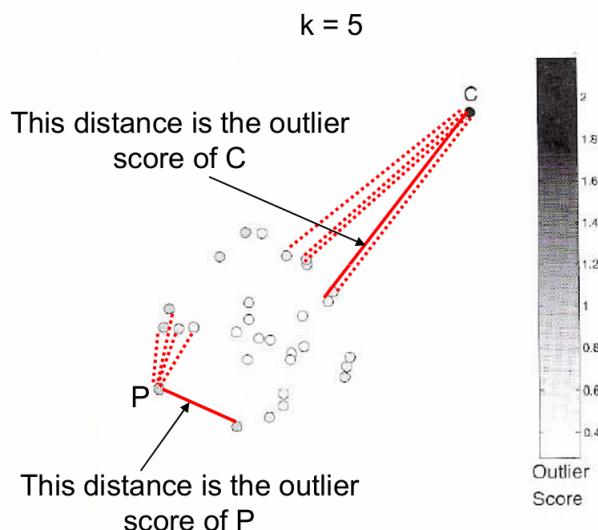


Figura 2.10: No supervisado basado en distancias: Vecino más cercano

2. Basado en el vecino más cercano por densidad: Define la densidad de un punto como la inversa de las sumas de las distancias a sus (K) vecinos más cercanos. Luego se define la densidad relativa de un punto (P) como el radio entre su K -densidad y la media de las K -densidades de sus K vecinos. La densidad relativa de un punto (P) será su valor de outlier.

Basados en clustering Suponiendo que existen clusters construidos por algún método de clustering, se podrá decir que un elemento nuevo es una anomalía si no pertenece a ningún cluster, o bien basándonos en la distancia al centroide más cercano. Tres posibles maneras de determinar la distancia al centroide más cercano, son:

1. usando la distancia euclídea al centroide más cercano, pero tiene el problema de que una anomalía puede estar cercana a un centroide y, por tanto, dar un bajo valor de outlier cuando realmente es un outlier.
2. medir la distancia relativa al centroide más cercano: la distancia relativa será el radio de la distancia del punto del centroide a la mediana de la distancia de todos los puntos en el cluster de ese centroide.

3. medir la distancia de Mahalanobis (lo cual implica el cálculo d la matriz de covarianza) de cada punto a la distribución del cluster.

2.2.5 Evaluación

Sólo la precisión (accuracy) del modelo no es una buena medida de evaluación del mismo, como se ha ejemplificado al principio de este capítulo, donde se mostró un clasificador de datos médicos que ignoraba los valores anómalos cuando estos eran poco representativos en el total de muestras del conjunto. Por tanto, para evaluar métodos supervisados de detección de anomalías será necesario emplear otras medidas adicionales, como el Recall, la precisión, el F-score o el área bajo la curva ROC.

Cabe destacar que es imposible mejorar todas las medidas simultáneamente, pues algunas de ellas presentan una relación inversa, por ejemplo, si la precisión crece, el recall decrece.

Definición 2.2.5 — Precisión. Evalúa el porcentaje de predicciones de una anomalía que se etiquetaron correctamente.

$$\frac{TP}{(TP + FP)} = \text{Prob}(\text{Real} = A | \text{Pred} = A)$$

Donde TP son los verdaderos positivos y FP son los falsos positivos.

Definición 2.2.6 — Recall. Mide la sensibilidad, o la tasa de verdaderos positivos, es decir, el porcentaje de anomalías reales de clase (A) que se consiguieron capturar.

$$\frac{TP}{(TP + FN)} = \text{Prob}(\text{Pred} = A | \text{Real} = A)$$

Donde TP son los verdaderos positivos y FP son los falsos positivos.

Definición 2.2.7 — Tasa de falsos positivos. Mide el porcentaje de casos normales que fueron predecidos erroneamente como anomalías.

$$FPR = \frac{FP}{(FP + TN)} = \text{Prob}(\text{Pred} = A | \text{Real} = NC)$$

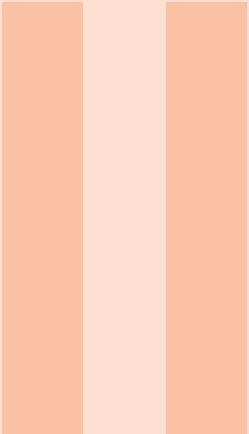
Donde FP son los falsos positivos y TN son los verdaderos negativos.

Definición 2.2.8 — Tasa de verdaderos negativos. Mide el porcentaje de casos normales que fueron predecidos correctamente.

$$TNR = \frac{TN}{(TN + FP)} = \text{Prob}(\text{Pred} = NC | \text{Real} = NC) = 1 - FPR$$

Definición 2.2.9 — F_1 – Score. Mide el equilibrio entre la precisión y el recall, por lo que un valor bajo de F_1 denota un desequilibrio entre precisión y recall.

$$F_1 = 2 \cdot \frac{\text{precisin} \cdot \text{recall}}{(\text{precisin} + \text{recall})}$$



Reglas de Asociación

3 Reglas de Asociación: Aspectos básicos 27

- 3.1 Introducción
- 3.2 Aplicaciones
- 3.3 Medidas de las reglas de asociación
- 3.4 Métodos de extracción de reglas
- 3.5 Conjuntos maximales y cerrados
- 3.6 Problemas

4 Reglas de asociación: Aspectos avanzados 34

- 4.1 Problemas de interpretabilidad
- 4.2 Medidas de calidad
- 4.3 Interpretación
- 4.4 Reglas de asociación difusas
- 4.5 Reglas Jerárquicas
- 4.6 Análisis de las reglas por grupos

Bibliografía 42

- Artículos
- Libros



3. Reglas de Asociación: Aspectos básicos

3.1 Introducción

La minería de reglas de asociación consiste en la aplicación de técnicas a bases de datos para extraer conocimiento de interés social o comercial, mayoritariamente, como dependencias entre ciertos elementos de la base de datos, donde la clase a la que pertenecen los items es desconocida.

Definición 3.1.1 — itemset. Las reglas de asociación son del tipo $X \rightarrow Y$, donde X se denomina el antecedente de la regla, e Y se denomina el consecuente. Ambos, X e Y son conjuntos de elementos (items) que cumplen que $X \cap Y = \emptyset$.

Los itemsets pueden contener 1, 2, 3, o N items, denominándose 1-itemset, 2-itemenset, o N -itemset, según el número de elementos que contenga. Por ejemplo, una regla de asociación sería:

$$\text{queso} \rightarrow \text{vino}$$

que se interpretaría como que la gente que compra queso (antecedente) compra vino (consecuente). La aplicación de las reglas de asociación a bases de datos empezó en el ámbito de los supermercados, para extraer información sobre los productos comprados por los clientes. En este caso, los **items** serían los artículos del supermercado que queremos asociar, y las **transacciones** serían itemsets, que en este caso serían un conjunto de ventas. Es decir, las transacciones definen casos concretos de una relación entre items.

P En una regla no pueden aparecer los mismos items en el consecuente y antecedente de la regla, pues sería redundante. Por ejemplo, si tenemos la regla:

$$\text{queso y vino} \rightarrow \text{mermelada}$$

no tendría sentido la regla:

$$\text{queso y vino} \rightarrow \text{vino y mermelada}$$

pues ya se encuentra esa relación reflejada en la regla anterior.

Los ítems de una base de datos pueden ser de dos tipos:

1. Si cada registro de la base de datos es un listado de elementos, no hay variables. Por tanto se define un ítem como cada uno de los posibles elementos.
2. La base de datos contiene un número conocido de características, y cada registro de la base de datos contiene un valor para cada variable, por tanto, un ítem será un par (atributo, valor) en este caso.

3.2 Aplicaciones

Las reglas de asociación no sólo se aplican para obtener conocimiento sobre datos pasados, sino que también se han aplicado para predecir en términos de futuro y presente. Algunas de sus aplicaciones más comunes, son:

- **Análisis de mercado:** como el ya mencionado ejemplo del análisis de las cestas de la compra en los supermercados para tomar decisiones acerca de posicionamiento de los productos, ofertar promociones, etc.
- **Extracción de información a partir de transacciones bancarias:** y otros datos sobre los clientes, como por ejemplo, predecir la probabilidad de impago de un cliente antes de conceder un préstamo.
- **Minería web:** para descubrir patrones presentes en la web acerca de navegadores, logs, contenidos de las páginas, etc.
- **Análisis en redes sociales:** para extraer asociaciones a partir de la información que se publica en las redes sociales, que se puede utilizar para sugerir amigos o páginas de interés.
- **Detección de intrusiones:** para detectar patrones cuando se está atacando la seguridad de un sistema informático.
- **Producción continua:** en procesos de manufacturación donde los materiales con procesados bajo reacciones químicas o calentamientos.
- **Bioinformática:** para encontrar relaciones existentes entre ciertos genes o bacterias.
- **Minería de textos:** para asociar la presencia de ciertas palabras en los documentos.

3.3 Medidas de las reglas de asociación

Dos medidas clásicas para evaluar las reglas de asociación son el soporte y la confianza de la regla.

Sopor te

Es una medida que toma valores entre 0 y 1, donde $\text{soporte}=1$ indica que aparece en todas las transacciones de la base de datos y 0 lo contrario, que no aparece en ninguna. El soporte de una regla y de un ítemset se definen de forma distinta:

Definición 3.3.1 — Soporte de un ítemset. Sea (X) un ítemset, se define su soporte como:

$$\text{Sop}(X) = \frac{\text{Num ocurrencias de } X}{\text{total de transacciones en la BD}}$$

(probabilidad de X en el conjunto de transacciones $p(X)$). Su soporte será la frecuencia con la que el ítemset ocurre en la base de datos.

Definición 3.3.2 — Soporte de una regla. Sea una regla $(X \rightarrow Y)$, se define su soporte como:

$$\text{Sop}(X \rightarrow Y) = \text{Sop}(X \cup Y) = \frac{\text{Num ocurrencias de } X \cup Y}{\text{total de transacciones en la BD}}$$

(probabilidad del itemset $X \cup Y$ en el conjunto de transacciones $p(X \wedge Y)$). Su soporte será la frecuencia con la que ocurre el itemset $X \cup Y$.

Confianza

Esta medida también toma valores entre 0 y 1, una confianza alta indica que siempre que ocurre un ítem (X) ocurre (Y), al contrario de un nivel de confianza bajo, que indica que cuando ocurre (X) no ocurre (Y).

Definición 3.3.3 — Confianza de una regla. Sea una regla $(X \rightarrow Y)$, se define su confianza como:

$$Conf(X \rightarrow Y) = \frac{Sop(X \rightarrow Y)}{Sop(X)}$$

Es decir, como la $p(X|Y) = p(X \wedge Y)/p(X)$.

A diferencia del soporte (definición 3.4.1), la confianza no tiene la propiedad anti-monótona, por lo que la confianza de un subconjunto de ítems sí puede ser mayor que la del conjunto completo. Sin embargo, en las reglas generadas a partir del mismo itemset la confianza sí cumple la propiedad de anti-monotonía.

Proceso de extracción de reglas

Ante un conjunto de transacciones de una base de datos, el objetivo es encontrar aquellas reglas que tengan un soporte igual o superior a un soporte mínimo prefijado ($minSop$), y una confianza igual o mayor a una mínima confianza ya prefijada ($minCon$), ambos dados por el experto del problema. Existen varios enfoques de extracción de reglas de asociación. Como por ejemplo:

- Por fuerza bruta: tratar de extraer las reglas más prometedoras generando todas las reglas posibles, calculando la confianza y soporte para cada una y eliminando las menos prometedoras. Es computacionalmente inasequible para hardware estándar.
- En dos pasos: Primero generar todos los itemsets frecuentes y luego generar las reglas con alta confianza a partir de ellos. El problema es que generar los itemsets frecuentes también es computacionalmente costoso.

Definición 3.3.4 — Itemset frecuente. Es aquel cuyo soporte es mayor o igual al umbral establecido por el experto ($minSop$).

Existen diversas estrategias para generar los itemsets frecuentes:

1. Reducir el número de candidatos, a través de una búsqueda completa sobre el espacio y reducciones a través de técnicas de poda.
2. Reducir el número de transacciones cuando aumenta el tamaño del dataset, esta técnica la usan métodos de extracción de enfoque vertical.
3. Reducir el número de comparaciones, no es necesario comprobar todas las transacciones para cada candidato. Es importante el uso de estructuras de datos eficientes para almacenar los candidatos y transacciones.

Proceso de generación de reglas

Dado un itemset frecuente (S), se generan las reglas haciendo todas las posibles combinaciones y filtrando sólo las que tengan una confianza $\geq minConf$. Hay dos formas:

- Generar reglas con un solo atributo en el consecuente
- Generar reglas con más de un atributo en el consecuente. Es decir, sea s el número de ítems del itemset $\Rightarrow \exists 2^s - 2$ reglas candidatas ($S \rightarrow \emptyset$ y $\emptyset \rightarrow L$ se ignoran).

3.4 Métodos de extracción de reglas

3.4.1 Apriori

Definición 3.4.1 — Anti-monotonía del Soporte. $\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$ es decir, el soporte de un subconjunto de ítems (X) es mayor o igual que el soporte del conjunto (Y).

Dada la propiedad antimonotónica del soporte, se mantiene que si un ítemset es frecuente, entonces todos sus subconjuntos también lo serán.

Procedimiento

El algoritmo Apriori está diseñado para trabajar con bases de datos de transacciones. Su procedimiento se basa en cuatro pasos iterativos. Antes de comenzar la fase iterativa, dado un valor de K, por ejemplo, K=1, se generan los ítemsets frecuentes de longitud=K, en este caso, de longitud = 1. Luego comienza el proceso iterativo siguiente:

1. Se genera el conjunto de (K+1) candidatos a ítemsets frecuentes, a partir del conjunto de ítemset frecuentes, a partir de combinar los ítemset frecuentes que solo se diferencian en el último ítem.
2. Calcular el soporte para cada candidato
3. Eliminar los candidatos que son infrecuentes.
4. Incrementar K en 1.

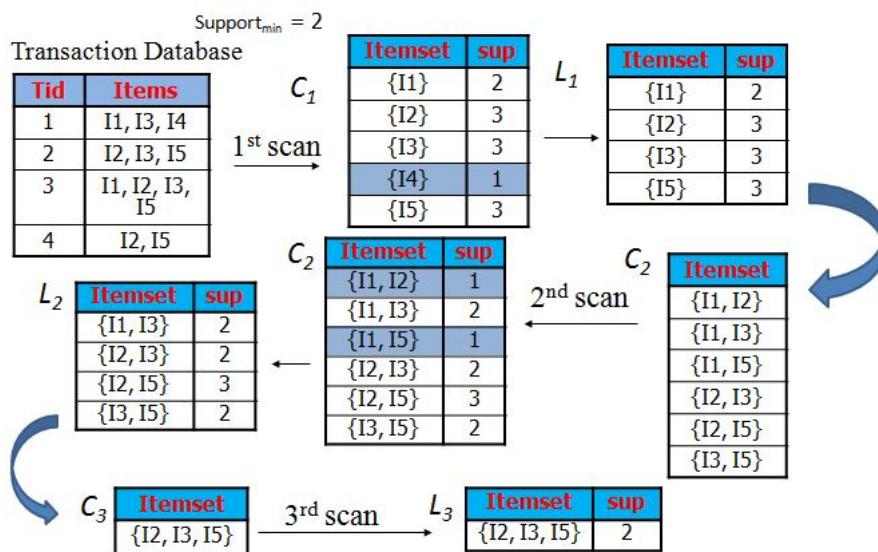


Figura 3.1: Ejemplo del algoritmo Apriori, imagen de [lessons2all](#)

Los ítemset frecuentes se pueden generar siguiendo una estrategia de búsqueda en anchura o en profundidad. Apriori emplea una búsqueda en anchura y un árbol hash para encontrar los ítemsets.

Eficiencia

Hay varios elementos que afectan en gran medida a la eficiencia del algoritmo Apriori. El umbral de mínimo soporte es uno de ellos, por lo que es importante elegirlo adecuadamente. Umbrales muy bajos implicarán generar muchos ítemsets frecuentes y mucho más largos que umbrales más altos. Por otra parte, el número de ítems en la base de datos también tiene un impacto directo en la eficiencia, ya que a más ítems, se necesitará más espacio para almacenar más ítemsets y más tiempo de cómputo.

Relacionada con la anterior, puesto que Apriori realiza varias pasadas por la base de datos completa, el tamaño de la misma tendrá un impacto directo en el tiempo de ejecución del algoritmo. Por

último, la longitud medida de las transacciones también tendrá un impacto grande en el rendimiento del algoritmo, dado que puede aumentar la longitud de los itemset frecuentes, demandando más espacio de almacenamiento y tiempo de cómputo.

3.4.2 Eclat

El procedimiento para este algoritmo es el mismo que para Apriori, pero a diferencia de que en Eclat, para cada itemset se almacena una lista con la posición en la base de datos de las transacciones en las que aparece ese ítem (tid-list).

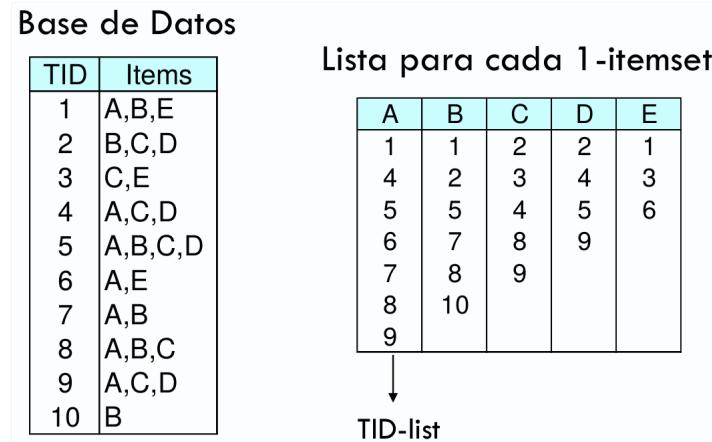


Figura 3.2: Lista generada para cada ítemset en Eclat

Procedimiento

Como ya se ha mencionado, el procedimiento es el mismo que Apriori, pero en este algoritmo el soporte de los ítemsets se calcula haciendo la intersección de las listas (tis-list) de sus ($K-1$) subconjuntos.

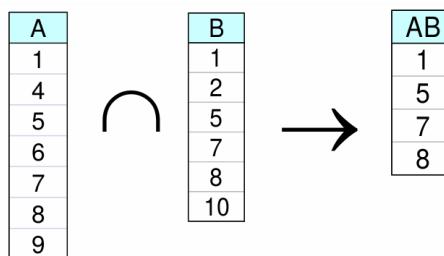


Figura 3.3: cálculo del soporte de los ítemsets en Eclat

De esta forma, el cálculo del soporte es muy rápido, aunque por otra parte las tid-lists intermedias pueden llegar a ser demasiado grande si la base de datos contiene muchas transacciones.

3.4.3 FP-growth

Esta técnica utiliza una estructura de árbol conocida como **FP-tree** para crear una representación comprimida de la base de datos. El FP-tree consta de:

- Una tabla de cabecera: donde para cada ítem de la base de datos hay una lista que enlaza todos los nodos del grafo donde aparece ese ítem.

- Un grafo de transacciones: donde se describen de forma abreviada todas las transacciones de la base de datos, indicando en cada nodo el soporte del itemset que se forma siguiendo el camino que va desde la raíz a dicho nodo.

Procedimiento

Una vez construido el FP-tree, se utiliza un enfoque recursivo basado en divide y vencerás para obtener los itemsets frecuentes. La extracción de los itemsets frecuentes se realiza en dos pasos:

1. Se calcula en soporte de cada ítem que aparece en el problema, recorriendo su lista correspondiente de la tabla de cabecera.
2. Para cada ítem que supere el soporte mínimo:
 - se extraen las ramas en las que aparece dicho ítem y se reajusta el soporte de los ítems de esas ramas en función del soporte del ítem.
 - se genera un nuevo FP-tree con las ramas extraídas.
 - se extraen los itemsets del FP-tree cuyo soporte supere el soporte mínimo.

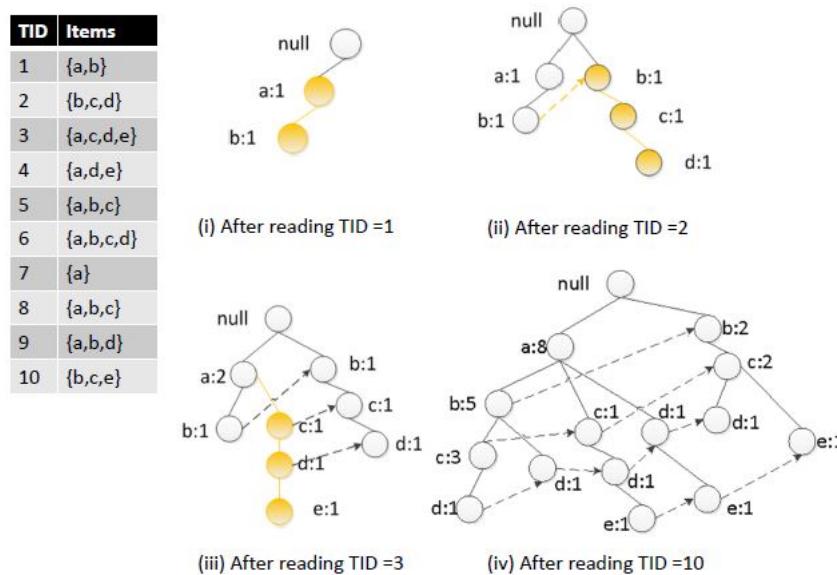


Figura 3.4: construcción de un FP-tree, foto de thardes.de

3.5 Conjuntos maximales y cerrados

Cuando el número de itemsets frecuentes se incrementa exponencialmente, esto genera un problema de rendimiento en términos de espacio y cómputo. Para ello surgen formas alternativas de representación, que permitan reducir el conjunto inicial pero no permitan generar todos los itemsets frecuentes a partir de ellas. Dos de esas representaciones son los **itemsets maximales** y los **itemsets cerrados**.

3.5.1 Itemsets maximales

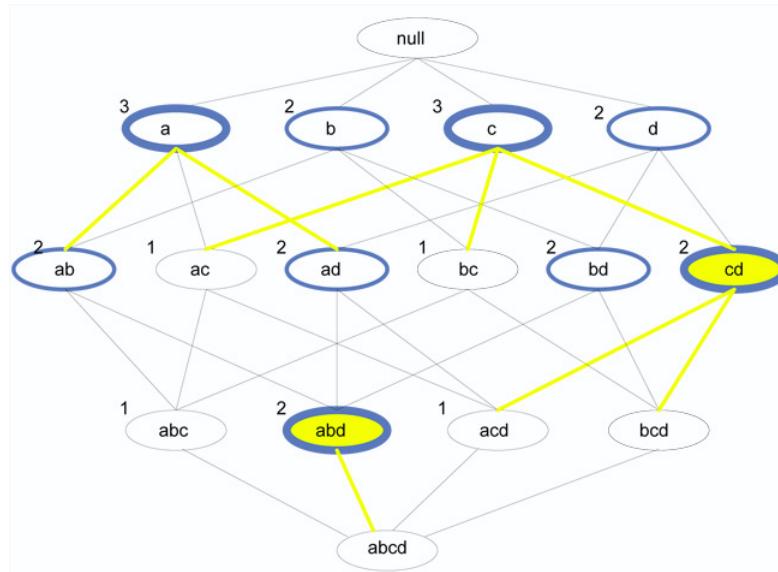
Definición 3.5.1 — Itemsets maximales. Son aquellos itemsets frecuentes para los cuales ninguno de sus superconjuntos inmediatos son frecuentes.

A partir de los itemsets maximales se pueden obtener todos los itemsets frecuentes, ya que serán todos los subconjuntos de ítems que se puedan formar a partir de ellos. Sin embargo, desconoceremos el soporte de los itemsets frecuentes obtenidos de este modo.

3.5.2 Itemsets cerrados

Definición 3.5.2 — Itemsets cerrados. Son aquellos itemsets frecuentes para los que ninguno de sus superconjuntos inmediatos tienen un soporte igual al suyo.

La ventaja de los itemsets cerrados es que a partir de ellos se pueden obtener tanto todos los itemsets frecuentes como el soporte de cada uno de ellos, ya que cualquier subconjunto de ellos que no sea otro itemset cerrado tendrá el mismo soporte que ellos. Como principal desventaja está que son más cantidad que los itemsets maximales y por tanto necesitarán más espacio para ser almacenados.



Tanto los itemsets maximales como los cerrados son subconjuntos de los itemsets frecuentes, aunque los maximales son una representación más compacta que los cerrados. Los cerrados se utilizan más que los maximales cuando la eficiencia es más importante que el espacio en disco.

3.6 Problemas

En este apartado se mencionan algunos problemas presentes en los mecanismos descritos a lo largo de este capítulo.

P Las reglas explicadas se centran en extraer conocimiento de bases de datos categóricas, sin embargo esto no siempre es así en datos reales.

P Las reglas explicadas tampoco explican como lidiar cuando tenemos una jerarquía presente en los items de los que extraer información.

P Las medidas de soporte y confianza presentan problemas cuando se usan para guiar la búsqueda de reglas dado que:

- un alto valor del soporte da lugar a reglas poco útiles
- la confianza no detecta cuando el soporte del consecuente es muy alto



4. Reglas de asociación: Aspectos avanzados

4.1 Problemas de interpretabilidad

Concluimos el capítulo anterior mencionando algunos de los problemas de las técnicas y medidas de evaluación básicas, revisadas en el capítulo 3. Otro problema muy ligado a las reglas de asociación son los problemas de interpretabilidad, que mayoritariamente suelen deberse a una de las tres causas siguientes:

Problemas derivados de los datos cuando los datos presentan problemas, las reglas resultantes pueden ser inservibles. Los problemas en los datos pueden deberse a un problema en su variabilidad (ítems muy frecuentes o muy poco frecuentes), falta de representatividad de todos los datos, presencia de sesgos, valores perdidos...etc. Por tanto, es importante preparar previamente los datos a la extracción de reglas.

Problemas derivados de los usuarios no siempre se dispone de expertos que puedan valorar las reglas, por tanto se pueden cometer confusiones semánticas, como por ejemplo, confundir la tendencia con la causalidad cuando la tendencia no tiene porqué implicar causalidad. Por tanto, se debe realizar un análisis semántico.

Problemas derivados de las medidas como se mencionó al final del capítulo anterior, las medidas clásicas de soporte y confianza presentan problemas cuando se utilizan como criterio para decidir cuando una regla es buena o no. Por tanto, y aunque no exista una medida perfecta, es conveniente usar además otras medidas de evaluación de la calidad de las reglas como el lift o el factor de certeza.

4.2 Medidas de calidad

Cuando se habla de medidas de calidad, se habla de medidas objetivas y subjetivas. Las primeras están basadas puramente en el cálculo de frecuencias, mientras las otras tienen en cuenta otros factores como el interés de la regla.

4.2.1 Medidas objetivas

Según Piatetsky-Shapiro, las medidas de calidad deben de tener una serie de características deseables, como:

- $I(A \rightarrow C) = 0$ cuando son independientes ($Sop(A \rightarrow C) = Sop(A) \cdot Sop(C)$), donde I es el interés de la regla.
- $I(A \rightarrow C)$ crece monotonamente con $Sop(A \rightarrow C)$ cuando se mantiene el resto de valores.
- $I(A \rightarrow C)$ decrece monotonamente con $Sop(A)$ o $Sop(C)$ cuando se mantiene el resto de valores.

Confianza confirmada

Toma valores entre -1 y 1, donde 0 denota independencia o imposible de predecir, 1 denota que el antecedente predice al consecuente y -1 denota que el antecedente predice la no ocurrencia del consecuente.

Definición 4.2.1 — Confianza confirmada. Se define la confianza confirmada para una regla $A \rightarrow C$, como:

$$Conf(A \rightarrow C) = Conf(A \rightarrow \neg C)$$

Es decir, la confianza confirmada de una regla trata de medir la utilidad de A (antecedente) para predecir C (consecuente).

Lift

Es una medida simétrica de asociación que puede tomar valores entre 0 e ∞ . El valor 1 denota independencia estadística, mientras que los valores inferiores a 0 denotan dependencia negativa.

Definición 4.2.2 — Lift. Se define el lift para una regla $A \rightarrow C$, como:

$$\frac{Conf(A \rightarrow C)}{Sop(C)} = \frac{Sop(A \rightarrow C)}{Sop(A) \cdot Sop(C)}$$

Al considerar el $Sop(C)$, permite resolver el problema de itemsets muy frecuentes.

Convicción

Es una medida que puede tomar valores entre 0 (abierto) e ∞ . El valor 1 denota independencia estadística, mientras que los valores inferiores a 0 denotan dependencia negativa. Los valores entre 1.01 y 5 se consideran interesantes, siendo 5 el umbral a partir del cual se consideran reglas inútiles por su obviedad.

Definición 4.2.3 — Convicción. Se define la convicción para una regla $A \rightarrow C$, como:

$$\frac{Sop(A) \cdot Sop(\neg C)}{Sop(A \rightarrow \neg C)}$$

Al considerar el $Sop(\neg C)$, permite resolver el problema de itemsets muy frecuentes.

Factor de certeza

Es una medida de implicación que mide la variación de nuestra creencia en el consecuente cuando se cumple el antecedente con respecto a la creencia a priori en el consecuente. Toma valores entre -1 y 1, donde 0 indica independencia estadística.

Definición 4.2.4 — Factor de certeza. Se define el factor de certeza para una regla $A \rightarrow C$,

como:

$$FC(A \rightarrow C) = \frac{Conf(A \rightarrow C) - Sop(C)}{1 - Sop(C)}, \text{ si } Conf(A \rightarrow C) \geq Sop(C)$$

$$FC(A \rightarrow C) = \frac{Conf(A \rightarrow C) - Sop(C)}{Sop(C)}, \text{ si } Conf(A \rightarrow C) < Sop(C)$$

Al considerar el $Sop(C)$, permite resolver el problema de itemsets muy frecuentes.

Yule's Q

Esta medida representa la correlación entre dos eventos dicotómicos relacionados positivamente. Toma valores entre -1 y 1, donde valor 0 denota independencia estadística, valores negativos denotan dependencia negativa y valores positivos denotan dependencia positiva.

Definición 4.2.5 — Yule's Q. Se define el Yule's Q para una regla $A \rightarrow C$, como:

$$\frac{Sop(AC) \cdot Sop(\neg A \neg C) - Sop(A \neg C) \cdot Sop(\neg AC)}{Sop(AC) \cdot Sop(\neg A \neg C) + Sop(A \neg C) \cdot Sop(\neg AC)}$$

Diferencia absoluta de confianza

Esta medida de implicación cuyo rango va de -1 a 1, donde 0 implica independencia estadística.

Definición 4.2.6 — Diferencia absoluta de confianza. Se define la diferencia absoluta de confianza para una regla $A \rightarrow C$, como:

$$Conf(A \rightarrow C) - Sop(C)$$

Al considerar el $Sop(C)$, permite resolver el problema de itemsets muy frecuentes.

Ratio de confianza

Esta medida de implicación cuyo rango va de -1 a 1, donde 0 implica independencia estadística. Es muy útil cuando se busca descubrir reglas que correspondan a itemsets poco frecuentes.

Definición 4.2.7 — Ratio de confianza. Se define el ratio de confianza para una regla $A \rightarrow C$, como:

$$1 - \frac{Conf(A \rightarrow C)}{Sop(C)}$$

o

$$1 - \frac{Sop(C)}{Conf(A \rightarrow C)}$$

Al considerar el $Sop(C)$, permite resolver el problema de itemsets muy frecuentes.

Además de estas medidas de interés, existen muchas más, como por ejemplo la diferencia de información, basada en entropía y que se ve muy afectada por el soporte o el Chi cuadrado normalizado, que también se ve afectado por el soporte además de resultar poco intuitivo de interpretar.

4.2.2 Medidas subjetivas

A diferencia de las medidas objetivas no sólo tienen en cuenta los datos, sino que miden el interés de la regla en función de la utilidad y novedad de la misma. Por ello, estas medidas deben tener en cuenta las creencias o necesidades del usuario. Para medir la utilidad de las reglas, hay que tener en cuenta las **restricciones** de la misma (bajo qué condiciones se da ese patrón), el **tiempo**

de vida (durante cuento tiempo será útil ese patrón), el **esfuerzo** que debemos hacer para actuar como indica el patrón, los **efectos laterales** o el **impacto**.

Reglas inesperadas

Un caso claro de regla útil, o, al menos, interesante son las reglas inesperadas. Las reglas inesperadas son aquellas que contradicen o sorprenden lo que el usuario piensa o espera encontrar. Para determinar cuando una regla es inesperada, es preciso especificar como representar lo que el usuario cree y como medir lo inesperado, tarea que normalmente se hace mediante el empleo de estadísticos y distancias. Los enfoques principales para detectar reglas inesperadas, son:

- Medidas probabilísticas: como las redes bayesianas para el uso de probabilidades condicionadas y determinio de la coherencia de la regla.
- Medidas de la distancia sintáctica: se basan en la distancia entre las nuevas reglas y el conjunto de creencias del usuario.
- Contradicción lógica: usan una medida objetiva para indicar lo que el usuario espera y analiza si hay alguna diferencia con los grados esperados por el usuario de esas medidas. Un tipo de contracción son las paradojas, como la paradoja de Simpson.

4.3 Interpretación

Definición 4.3.1 — Interpretación. Una interpretación es una correspondencia que se establece entre elementos de la estructura de los datos y elementos del marco formal.

Tabular común

Un ejemplo de interpretación es la tabular común. Suponiendo que los datos tienen una estructura tabular, los **items** serían las parejas (atributo,valor) y las **transacciones** serían los registros de la base de datos. Por ejemplo, dada la tabla:

DNI	Puesto	Sueldo	Estudios
111111111	Administrativo	Bajo	Medios
222222222	Programador	Medio	Medios
333333333	Analista	Medio	Superiores
444444444	Gerente	Alto	Superiores

Figura 4.1: ejemplo de base de datos en formato tabla

Los items serían: (Sueldo, Medio),(Sueldo,Bajo),(Sueldo,Alto),(Puesto, Administrativo),(Puesto, Programador),(Puesto, Analista),...etc, mientras que un ejemplo de transacción sería: (Puesto, Administrativo), (Sueldo,Bajo), (Estudios, Medios).

Items negados

Otro ejemplo de interpretación es la que se puede obtener de datos binarios. Una posible

i_1	i_2	i_3	i_4
1	0	1	0
0	0	0	0
0	1	1	0
0	1	1	1
1	1	1	1
1	1	1	1

Figura 4.2: ejemplo de base de datos binarios

interpretación sería tomar cada columna de la tabla como un item y cada fila como una transacción,

donde un ítem estará en una transacción si su valor es 1. Otra posible interpretación en este caso sería considerar que cada columna son dos ítems (i y $\neg i$), y que las transacciones son las filas, donde el ítem i estará en la transacción si el valor es 1, y el ítem $\neg i$ estará en la transacción si el valor es 0.

Reglas jerárquicas

En ellas se dispone de una o varias jerarquías de ítems. Los datos son un conjunto de transacciones que contienen ítems básicos y la jerarquía de categorías que los agrupa en distintos niveles. Estas reglas son especialmente útiles cuando los ítems de los niveles más bajos no tienen soporte suficiente para generar reglas, ya que se puede recurrir a mirar los ítems inmediatamente superiores en la jerarquía. Si al contrario, el soporte es demasiado alto, se recurriría a mirar ítems de más bajo nivel. En este caso se interpretan como ítems la unión de los ítems básicos y las categorías de la

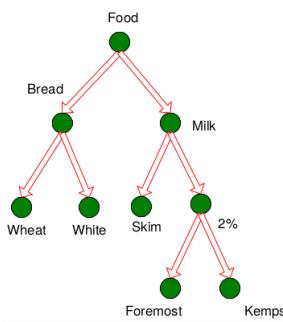


Figura 4.3: Se dispone de una jerarquía de ítems

jerarquía, y como transacciones se toman las transacciones de ítems básicos y se les añaden los ancestros en la jerarquía de los ítems presentes.

Reglas secuenciales

Definición 4.3.2 — patrón secuencial. Un patrón secuencial se define como una secuencia de ítems básicos que aparecen en un orden prefijado en tiempo o en cualquier otro criterio.

Se pueden extraer reglas secuenciales cuando la aparición de los ítems básicos en los datos está ordenada, por ejemplo, datos sobre cestas de compra donde cada artículo tiene asociada una fecha de compra. En este caso, los ítems serían las secuencias ordenadas de ítems básicos, mientras que las transacciones serían los conjuntos formados con esos ítems.

Estas reglas tienen una sola secuencia tanto en su parte derecha como en su parte izquierda, aunque ambas partes pueden estar formadas por varios ítems básicos ordenados.

Reglas cuantitativas

Se utilizan cuando se tienen datos que, aunque tengan una estructura definida, tienen dominios numéricos con muchos valores. Por ejemplo, una base de datos con información de edad, peso y altura de varias personas (Figura 4.4).

El problema de este tipo de bases de datos es que cuando se intentan extraer reglas de ellas el soporte de la mayoría de los ítems suele ser muy bajo, y además las reglas generadas son muy pobres en interés o significado. Para solucionarlo, el dominio de los atributos se divide en intervalos, determinando los ítems como pares (atributo, intervalo), y sobre los datos transformados se extraen las reglas.

La definición de los intervalos se puede hacer, o bien a priori, por parte del experto, o bien usando

Edad	Peso	Altura
10	40	130
50	90	185
2	10	85
70	70	165

Figura 4.4: BD de datos numéricos que toman muchos valores diferentes

algún algoritmo automático (por ejemplo uno que aprenda los intervalos y a partir de ellos extraiga las reglas, o uno que busque los mejores intervalos para dar reglas con buen soporte).

Dependencias aproximadas

Se aplican cuando se dan ciertos patrones en BD relacionales que se corresponden a dependencias funcionales con excepciones. Las medidas de estas reglas son útiles para valorar la calidad de las dependencias.

Definición 4.3.3 — Dependencia funcional. La dependencia funcional entre dos atributos A y B , se da en una base de datos cuando el valor de A determina el valor de B .

En este caso se considerarían como items los atributos de la tabla y las transacciones serían los pares de tuplas de la tabla. Es decir, el item asociado al atributo A está en la transacción asociado al par de tuplas (t,s) si $t[A] = s[A]$.

Dependencias graduales

Se aplican cuando se asocian entre las variaciones de los atributos, y representan correlaciones (positivas y/o negativas).

Definición 4.3.4 — Dependencia gradual. La dependencia gradual entre dos atributos A y B , se da en una base de datos si se dan reglas del tipo:

$$\forall t, s \in DB, \text{ si } t[A] > s[A] \Rightarrow t[B] > s[B]$$

En este caso se considerarían como items los pares (atributo, variación), donde la variación puede ser $<$ o $>$. Las transacciones serían las asociadas a los pares de tuplas de la BD, es decir, el item asociado al par $(A, >)$ está en la transacción asociada al par de tuplas $(t,s) \iff t[A] > s[A]$.

4.4 Reglas de asociación difusas

Las reglas cuantitativas emplean intervalos precisos para extraer reglas de interés en conjuntos de datos con atributos numéricos que toman muchos valores. La diferencia entre las reglas de asociación difusas y las reglas de asociación cuantitativas se encuentra en la naturaleza de los intervalos considerados, ya que para reglas cuantitativas se emplean intervalos precisos, mientras que para reglas difusas se emplean intervalos difusos que reflejan el cambio entre intervalos de forma más gradual.

Definición 4.4.1 — Soporte de un itemset (reglas difusas). Sea (X) un itemset, se define el Soporte de (X) para reglas difusas como:

$$Sop(X) = \sum_i^{\text{total ejemplos}} \mu_x(i) / \text{total ejemplos}$$

donde $\mu_x(i)$ es el grado con el que el itemset (X) cubre el ejemplo i .

Definición 4.4.2 — Soporte de una regla (reglas difusas). Sea $(X \rightarrow Y)$ una regla difusa, se define su soporte como:

$$Sop(X \rightarrow Y) = \sum_i^{\text{total ejemplos}} \mu_{xy}(i)/\text{total ejemplos}$$

Definición 4.4.3 — Confianza de una regla (reglas difusas). Sea $(X \rightarrow Y)$ una regla difusa, se define su confianza como:

$$Conf(X \rightarrow Y) = \frac{Sop(X \rightarrow Y)}{Sop(X)}$$

Extracción de reglas difusas

Para extraer este tipo de reglas se suele emplear alguno de estos enfoques:

- Realizar las particiones difusas a priori: posteriormente extrayendo de ellas las reglas según los umbrales de soporte y confianza mínimos. Como se mencionó previamente, las particiones pueden ser proporcionadas por un experto o un modelo automático.
- Aprender las reglas y las particiones difusas simultáneamente.

4.5 Reglas Jerárquicas

Cuando en la sección (4.3) se habló de reglas jerárquicas se comentó el enfoque habitual para generarlas, que consiste en encontrar los itemsets frecuentes combinando entre sí los items de un mismo nivel, donde luego los itemsets frecuentes obtenidos en cada nivel se combinaban de nuevo para construir las reglas. Sin embargo esto presenta dos problemas. Por un lado, se generarán muchas reglas obvias y por otro lado, el tiempo de cómputo será muy elevado.

Como solución a lo anterior, surgen dos alternativas. Ambas usan el soporte de los itemsets de cada nivel para podar la búsqueda de itemsets frecuentes.

1. La primera alternativa consiste en utilizar el soporte mínimo para todos los niveles de la jerarquía, utilizándolo como poda (si en un nivel un item no supera el soporte mínimo no se comprueban sus descendientes). Este enfoque tiene el problema de que si los $minSop$ son altos se pueden perder asociaciones interesantes en los niveles bajos, e igualmente $minSop$ bajos pueden generar asociaciones poco interesantes.
2. La segunda alternativa se conoce como **Soporte reducido**. Consiste en ir reduciendo el valor del $minSop$ a medida que se desciende en la jerarquía. Existen varias estrategias de búsqueda en este enfoque:
 - Level by level independent: reduce el $minSop$ en cada nivel y examina todos los niveles mientras que sus nodos padres sean frecuentes.
 - Level-cross filtering by k-itemset: reduce el $minSop$ en cada nivel. Examina un k-itemset del nivel $i \iff$ su correspondiente k-itemset padre del nivel $i - 1$ es frecuente.
 - Level-cross filtering by single item: un k-itemset del nivel i se examina \iff todos sus items tienen padres frecuentes en el nivel $i - 1$.
 - Controlled level-cross filtering by single item: se asigna un umbral a cada nivel. Si el itemset tiene un soporte superior al umbral se examina, aunque tenga algún antecesor que no supere el $minSop$ de su nivel.

Una vez obtenidos los itemsets frecuentes, se combinan independientemente de los niveles a los que pertenezcan, formando las reglas de asociación. Sin embargo, pueden surgir muchas reglas redundantes por este proceso.

4.6 Análisis de las reglas por grupos

Se trata de realizar el análisis de las reglas de forma colectiva, es decir, teniendo varias reglas en cuenta en lugar de evaluar las reglas de forma individual, lo cual puede aportar más información. Sería, por ejemplo, estudiar simultáneamente las reglas: $(A \rightarrow C)$ con $(A \rightarrow \neg C)$. Esto permite estudiar mejor la causalidad de ciertos items en casos donde sea dudosa, estudiar la evolución de los conjuntos de reglas a lo largo del tiempo, las excepciones y las anomalías.



Bibliografía

Artículos

- [2] George Karypis, Eui-Hong Han y Vipin Kumar. “Chameleon: Hierarchical clustering using dynamic modeling”. En: *Computer* 32.8 (1999), páginas 68-75 (véanse páginas 8, 13, 14).

Libros

- [1] Charu C. Aggarwal. *Outlier Analysis*. Springer Publishing Company, Incorporated, 2013. ISBN: 1461463955, 9781461463955 (véanse páginas 17, 18).