2. Fundamentos de Minería de Datos

Introducción

En este capítulo se introducen más en profundidad los conceptos de Data Mining y proceso KDD. Seguidamente, se expone una clasificación de los distintos procedimientos de minería de datos. Y, por último, se presenta el estado del arte de los algoritmos de Data Mining para los procedimientos de clustering, clasificación y descubrimiento de reglas de asociación.

2.1. Data Mining y el proceso KDD

Data Mining (o la minería de datos) forma parte del proceso total de descubrimiento de conocimiento en bases de datos (KDD o Knowledge Discovery in Databases). A pesar de ello, en la literatura se tiende a usar como sinónimos "Data Mining" y KDD (Knowledge Discovery in Databases).

El proceso KDD, de forma general, consiste en la recuperación de conocimiento no evidente a partir de datos en bruto. El concepto empieza a ser interesante desde el momento en el que el volumen de datos a tratar no es abarcable por un humano y se requiere el uso de computación para examinar los datos de forma ágil y, generalmente, detectar patrones no evidentes, útiles y no redundantes.

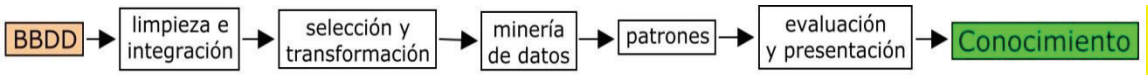

Figura 1. Descomposición del proceso KDD

La Figura 1 muestra el esquema del proceso KDD. A partir de una (o varias) bases de datos se extrae conocimiento. Para ello, en primer lugar se limpian los datos y en el caso de tener distintas fuentes, se integran. Este proceso consiste en agrupar los datos si existen distintas fuentes, y eliminar registros anómalos, en los que falten atributos o se especifiquen valores fuera de rango.

Seguidamente se seleccionan los datos relevantes para el análisis en cuestión y se transforman en caso de ser necesario. Este proceso consiste en elegir los atributos que serán analizados y normalizar sus valores o cambiarlos de tipo.

El siguiente paso, con el que también se da nombre a todo el proceso, es el de minería de datos. En este momento se aplican los diferentes procesos y algoritmos capaces de obtener conocimiento no evidente a partir de los datos previamente procesados. En general, se obtienen patrones. Una vez se obtienen los resultados del Data Mining, se evalúa la calidad de los patrones obtenidos con medidas propias de cada algoritmo. Y finalmente se presenta el resultado de forma inteligible a un usuario humano, habiéndose obtenido de este modo nuevo conocimiento.

2.2. Procedimientos de Data Mining

Un procedimiento de Data Mining es un problema que puede ser resuelto usando algoritmos de minería de datos. Cada procedimiento puede ser completamente diferente de otro, teniendo sus propios requisitos y obteniendo información de diferente tipo. Debido a la amplia variedad de procesos existentes y la poca homogeneidad en la literatura a la hora de tratarlos, a continuación se enumeran los más usuales:

- **Clasificación**: En este procedimiento cada registro pertenece a una clase predefinida, y el objetivo es predecir la clase a la que se corresponden nuevos registros, o bien registros que por otro motivo no estén clasificados de forma previa.

- **Regresión**: El objetivo de este procedimiento es el de predecir un valor numérico asociado a un registro, minimizando el error entre el valor predicho y el real.

- **Clustering**: Este procedimiento es similar a una clasificación, pero con la relevante diferencia de que los grupos en los que hay que clasificar a los diferentes registros no están previamente definidos. El objetivo de este procedimiento es pues encontrar grupos diferenciados de registros similares respecto a los de su propio grupo.

- **Correlaciones**: El objetivo de este procedimiento es obtener el grado de similitud existente entre, al menos, un par de variables numéricas.

- **Reglas de asociación**: Este procedimiento es tradicionalmente el más importante a la hora de ser aplicada la minería de datos a comercios. Consiste en encontrar relaciones no evidentes entre atributos no numéricos. Las reglas suelen darse en la forma "SI ocurre a, ENTONCES ocurre b". Además suele extenderse la búsqueda de reglas a reglas multinivel (clasificando previamente los atributos de los registros de forma jerárquica), y reglas de asociación secuenciales cuando existen marcas de tiempo.

En el caso del uso de Data Mining aplicado a comercio electrónico serán de especial importancia los procesos de clasificación, clustering y extracción de reglas de asociación (tanto básicas, como multinivel y secuenciales).

2.3. Algoritmos en procedimientos de Data Mining. Estado del Arte.

En esta sección se presentan los algoritmos existentes que pueden ser utilizados para realizar los distintos procedimientos de Data Mining aplicados a comercio electrónico. Hay que destacar que un procedimiento puede ser resuelto por varios algoritmos distintos, pero que a su vez un mismo algoritmo puede ser utilizado para realizar procedimientos distintos. No obstante, se presenta una clasificación asociando cada algoritmo a su procedimiento más natural.

2.3.1. Clustering

El objetivo de este procedimiento es encontrar grupos diferenciados de elementos similares respecto a los de su propio grupo. Se enumeran a continuación los algoritmos más conocidos aplicados al procedimiento de clustering, divididos en distintos grupos. No siempre hay una división clara al poder mezclarse varios conceptos en el mismo algoritmo.

- **Basados en particiones**: Estos métodos construyen "k" particiones de datos, donde cada partición representa un clúster, con la condición de que cada grupo debe contener al menos un objeto, y cada objeto debe estar contenido en exactamente un grupo.

- **kmeans**: Este algoritmo toma como entrada un parámetro k y realiza una partición de un conjunto de n objetos en k clústeres de tal modo que la similitud entre objetos es alta dentro de cada clúster, pero baja entre clústeres distintos. La similitud es medida considerando el valor medio de los objetos en el clúster. El uso de este algoritmo es considerado un problema NP-hard.

    - **kmodes**: Variante del algoritmo kmeans que remplaza el valor medio de los objetos por la moda de los mismos. El objetivo es poder aplicar el algoritmo sobre datos categóricos directamente, sin tener que convertirlos a numéricos.

    - **kprototypes**: Variante del algoritmo que integra kmeans y kmodes para ser aplicado sobre datos con atributos mixtos.

    - **kmedoids**: Variante del algoritmo kmeans que reemplaza el concepto de "media" del conjunto de objetos de un clúster por el concepto de "objeto más representativo". La idea es tomar el objeto más representativo del clúster en lugar de la media la hora de iterar. Al tratar los objetos de este modo se evita la desviación en la media que pueden provocar elementos anómalos.

- **EM** (*Expectation Maximization*): El algoritmo EM extiende la idea de los métodos basados en particiones de un modo distinto. En lugar de asignar cada objeto a un clúster, asigna cada objeto a un clúster de acuerdo a un peso que representa la probabilidad de pertenencia, con lo cual no existen fronteras estrictas entre clústeres.

- **CLARA** (*Clustering Large Applications*): El algoritmo CLARA se basa en un muestreo y es utilizado cuando el conjunto de datos es excesivamente amplio. Consiste en aplicar el algoritmo sobre una porción representativa de los datos, en lugar de sobre el conjunto completo.

- **Basados en jerarquía**: Estos métodos crean una descomposición jerárquica del conjunto de datos dado. Pueden ser agrupativos o divisivos según se forme la descomposición.

- **AGNES** (*Aglomerative Nesting*): Este algoritmo es de tipo agregativo. Inicialmente identifica cada un objeto con el clúster más pequeño posible, en el que solo está contenido él mismo. Los clústeres se van agrupando paso a paso de acuerdo con algún criterio definido.

- **DIANA (***Divisive Analysis***)**: Este algoritmo es de tipo divisivo y sigue el camino contrario a AGNES. Inicialmente todos los objetos forman el clúster inicial, y este clúster se va dividiendo paso a paso de acuerdo con algún criterio definido.

- **BIRCH** (*Balanced Iterative Reducing and Clustering Using Hierarchies*): El algoritmo BIRCH es una mejora de AGNES-DIANA para solventar el problema de que las decisiones de agrupamiento o división son críticas al no poder volver atrás una vez tomada tal decisión. BIRCH comienza clasificando los objetos jerárquicamente usando estructuras de árbol y aplica otros algoritmos de clustering para clarificar los clústeres.

- **CURE** (*Clustering Using Representatives*): CURE representa cada clúster por un cierto número fijo de elementos representativos, tratando de esta forma de encontrar clústeres no esféricos.

- **ROCK**: Este algoritmo es de tipo agregativo y constituye una alternativa a CURE usada en el caso de tratar con atributos categóricos, teniendo en cuenta la interconectividad entre los mismos.

- **Chameleon**: La idea de Chameleon es tener en cuenta tanto la interconectividad como la cercanía a la hora de identificar el par de clústeres más parecidos.

- **Basados en densidad**: La mayoría de los métodos de clustering están basados en encontrar la distancia entre objetos. Tales métodos solo puede encontrar clústeres esféricos y con dificultad pueden encontrar clústeres con formas arbitrarias. La idea de los métodos basados en densidad consiste en aumentar el crecimiento de los clústeres mientras la densidad en el "vecindario" exceda cierto umbral.

  - **DBSCAN**: Este algoritmo desarrolla regiones con densidad suficientemente alta en clústeres y descubre clústeres de formas arbitrarias en bases de datos espaciales con ruido. Funciona bien si la densidad de cada clúster es constante.

- **OPTICS** (*Ordering Points To Identify the Clustering Structure*): El algoritmo OPTICS es similar a DBSCAN, con la diferencia de que es capaz de detectar clústers de densidad variable.

- **DENCLUE** (*Density-based Clustering*): DENCLUE es un algoritmo basado en un conjunto de funciones de distribución de densidad. Se nutre de la idea de que cada punto puede ser formalmente modelado usando una función matemática que describe el impacto de cada punto de su "barrio"; que la densidad del conjunto total del espacio de datos puede ser modelado analíticamente como la suma de funciones de influencia de todos los puntos; y de que los clústeres pueden ser determinados matemáticamente identificando tractores de densidad.

- **Basados en rejilla**: Estos métodos cuantifican el espacio de los objetos a tratar en un número finito de celdas que forman una estructura de rejilla. La principal ventaja de estos métodos es la alta velocidad de ejecución conseguida, que en este caso suele ser independiente del número de objetos a tratar, y solo dependiente del número de células en cada dimensión en el espacio cuantificado.

  - **STING** (*Statistical Information Grid*): El algoritmo STING divide el espacio en celdas rectangulares y según distintas resoluciones o capas. En cada celda se almacena información estadística correspondiente con los elementos incluidos en dicha celda.

  - **WaveCluster**: Este algoritmo impone una estructura de rejilla multidimensional en el espacio de datos. En cada celda sumariza la información del grupo de puntos que caen dentro de ella. Esta información de recopilación es utilizada por una transformación de onda (Wavelet Transform) y su posterior análisis de clustering.

  - **CLIQUE** (*Clustering In Quest*): CLIQUE integra clustering basado en densidad y basado en rejilla. Está basado en que, en general, dado un conjunto muy grande de puntos multidimensionales, el espacio no está ocupado de forma uniforme por los puntos (identifica áreas o unidades pobladas y despobladas), y en que una área o unidad es densa si la fracción del total de puntos contenidos excede un parámetro del modelo de entrada.

- **Basados en modelos**: Estos métodos parten de una hipótesis de modelo por cada uno de los clústeres y encontrar el mejor encaje posible de los datos en el modelo dado.

    - **COBWEB**: Este algoritmo tiene como entrada objetos descritos con pares del tipo atributo categórico-valor. COBWEB crea un clustering jerárquico en forma de árbol de clasificación usando una medida de evaluación heurística llamada "category utility".

En este apartado se han presentado los algoritmos de clustering. En el siguiente apartado se presentan los algoritmos de clasificación.

2.3.2. Clasificación

En este procedimiento cada registro pertenece a una clase predefinida, y el objetivo es predecir la clase a la que se corresponden nuevos registros, o bien registros que por algún otro motivo no estén clasificados previamente. Se enumeran a continuación los algoritmos más conocidos aplicados a este procedimiento.

- **Inducción de árboles de decisión**: Un árbol de decisión es un diagrama de flujo en forma de árbol, donde cada nodo interno indica una prueba sobre un atributo, cada rama representa el resultado de dicha prueba, y los nodos hoja representa las clases entre las que clasificar. El algoritmo básico para la construcción de árboles de decisión parte de un conjunto de muestras de entrenamiento y una lista de atributos candidatos, y obtiene como resultado el árbol de decisión propiamente dicho. Dependiendo del sistema de aprendizaje los algoritmos se pueden dividir en aprendizaje de árboles de decisión por participación o por cobertura.

    - **Por partición**: Partiendo de la suposición de que las clases en las que se clasifican los objetos son disjuntas se deriva la propiedad de que cada objeto será clasificado en una y sólo una clase. En los algoritmos por partición se divide el espacio de instancias de arriba a abajo utilizando cada vez una partición, es decir, un conjunto de condiciones excluyentes y exhaustivas; y una vez elegida la partición, ésta no se puede cambiar. Los algoritmos de partición se diferencian entre ellos según los distintos criterios de partición y las particiones a considerar.

        - **ID3**: Es el algoritmo base a partir del que se derivan los demás. Usa una medida de entropía para decidir cuál nodo dividir en el siguiente paso del algoritmo, y una medida de ganancia para estimar la ganancia producida por la división sobre un atributo.

        - **C4.5**: C4.5 es una mejora de ID3, añadiendo la posibilidad de manejar atributos continuos y discretos, manejar atributos con costos diferentes y podando los árboles después de la creación.

    - **Por cobertura**: Los árboles de decisión se pueden expresar como conjuntos de reglas. Los métodos por cobertura van añadiendo reglas mientras vayan cubriendo ejemplos de una forma consistente. A diferencia del método anterior, se descartan ejemplos ya cubiertos por las reglas obtenidas.

        - **AQ**: Es el algoritmo básico a partir del cual se derivan los demás. Necesita como entrada un conjunto de ejemplos (positivos y negativos).

        - **CN2**: Es una mejora de AQ que incluye preprocesos y postprocesos para generalizar las reglas.

        - **FOIL**: Este es un algoritmo similar al AQ, pero con la diferencia de que obtiene árboles de decisión relacionales, en los que las pruebas de los nodos intermedios son conjunciones de literales cuyas variables están cuantificadas existencialmente, en vez de ser condiciones sobre el valor de un atributo.

- **SLIQ**: Es un algoritmo de inducción de árboles de decisión para conjunto de entrenamiento tan grandes que no quepan en memoria. Define el uso de una nueva estructura de datos para facilitar la construcción del árbol y hace uso del disco para guardar la lista de atributos.

- **SPRINT**: Es un algoritmo similar a SLIQ pero que define una estructura de datos distinta.

- **Naïve Bayes**: Se corresponde con la red bayesiana más simple posible. Este método de clasificación supone que todos los atributos son independientes conocido el valor de la variable clase. Se tiene una estructura fija y sólo es necesario aprender el valor de las distintas probabilidades.

- **Redes bayesianas**: Las redes bayesianas se representan mediante un grafo dirigido acíclico. El conocimiento se expresa en la definición de las relaciones de dependencia o independencia entre las variables que componen el modelo. Habrá por tanto que aprender (construir) un modelo.

- **Aprendizaje**: Se enumeran los distintos algoritmos de aprendizaje de redes bayesianas conocidos:

  - **K2**: Busca soluciones candidatas cada vez mejores siguiendo un esquema voraz y partiendo de que las variables de entrada están ordenadas, y los posibles padres de una variable aparecen en orden antes que ella misma.

  - **B**: Este algoritmo es similar al K2, pero sin la restricción de ordenación previa de las variables.

  - **HC** (*Hill Climbing*): El algoritmo HC parte de una solución inicial y a partir de ésta se calcula el nuevo valor de la métrica utilizada de todas las soluciones vecinos a la solución actual, quedándose con el vecino que resulte con mejor valor de tal métrica.

- **Clasificadores**: Se enumeran los distintos modelos clasificadores como redes bayesianas conocidos:

  - **TAN** (*Tree Augmented Naïve Bayes*): Es una red similar a la Naïve Bayes, pero que admite ciertas dependencias entre los atributos. Supone que los atributos forman una red bayesiana con forma de arbol.

  - **BAN** (*Bayesian Network Augmented Naïve Bayes*): En BAN se aprende una red bayesiana para todos los atributos distintos del de la clase a clasificar y, más tarde, se aumenta el modelo añadiendo la variable clase y aristas desde tal variable hacia todos los atributos.

- **Redes neuronales**: Una red neuronal está formada por una capa de entrada, una capa de salida y una o múltiples capas ocultas entre la entrada y la salida. Cada capa está compuesta por uno o varios nodos, los nodos están conectados todos con todos entre dos capas consecutivas y no existe conexión alguna entre nodos de capas no consecutivas. A cada conexión se le asigna un peso. Es necesario definir la topología de las redes neuronales antes de ser entrenadas para ser usadas en clasificación.

- **Propagación hacia atrás**: Es un proceso iterativo realizado usando un conjunto de datos de entrenamiento que compara la predicción de la red para cada muestra con el valor real esperado. Tras cada muestra se modifica el valor de los pesos para minimizar el error. El proceso se detiene al cumplir una condición definida, como puede ser el que el porcentaje de muestras mal clasificadas sea menor que cierto umbral.

- **KNN** (*K-Nearest Neighbor*): Este algoritmo es un método de clasificación no paramétrico que estima la probabilidad de que un elemento pertenezca a una clase. Se basa en el aprendizaje por analogía. Almacena las muestras de entrenamiento como puntos en el espacio euclidiano.

- **CBR** (*Case-Based Reasoning*): Este algoritmo es similar al KNN, pero almacena las muestras de entrenamiento como descripciones simbólicas complejas.

- **Algoritmos genéticos**: Este tipo de algoritmos intentan incorporar ideas de la evolución natural. Se parte de una población inicial consistente en reglas generadas de forma aleatoria que evolucionan según cruces y mutaciones.

En este apartado se han presentado los algoritmos de clasificación. En el siguiente apartado se presentan los algoritmos que descubren reglas de asociación.

2.3.3. Reglas de Asociación

Este procedimiento consiste en descubrir o inferir reglas de asociación en un conjunto de datos. El objetivo es descubrir patrones no evidentes y novedosos que proporcionen nuevo conocimiento útil.

- **Apriori**: Se basa en la búsqueda de los conjuntos de ítems con una determinada cobertura previamente especificada. En primer lugar se construyen los conjuntos formados con un solo ítem que superan la cobertura mínima. Este conjunto de conjuntos se utiliza para construir el conjunto de conjuntos de dos ítems, y así sucesivamente hasta que se llegue a un tamaño en el cual no existan conjuntos de ítems que cumplan con la cobertura mínima requerida. Y en segundo lugar se extraen de estos conjuntos de reglas las que tengan un nivel de confianza mínimo.

- **Apriori multinivel**: Aprovechando una clasificación de conceptos de forma jerárquica, pueden inferirse reglas multinivel, apareciendo nuevas versiones del algoritmo:

  - **Soporte uniforme**: En esta versión se utiliza el mismo límite de cobertura en todos los niveles. Tiene el problema de que los conjuntos de ítems de niveles inferiores no son tan frecuentes como los de niveles superiores, por lo que no es, en general, razonable utilizar el mismo límite de cobertura para cada nivel.

  - **Soporte reducido**: En esta versión en cada nivel se establece un límite de cobertura específico que normalmente se va incrementando conforme se desciende en la jerarquía de conceptos.

  - **Niveles cruzados**: Existen además versiones del algoritmo capaces de detectar patrones entre distintos niveles de agregación.

- **AprioriAll**: Esta es la extensión del algoritmo **Apriori** utilizada para encontrar patrones de ítems ordenados secuencialmente. En este algoritmo la cobertura se define con respecto a secuencias de conjuntos de ítems.

- **FP-growth** (*Frequent Pattern Growth*): Este algoritmo es similar al **Apriori**, pero usa un método de crecimiento de patrones por fragmentos que evita el costoso proceso de generación de candidatos y prueba de Apriori.

- **ARCS** (*Association Rule Clustering System*): Este algoritmo busca reglas de asociación cuantitativas y bidimensionales. Proviene del procesamiento de imágenes.

- **Eclat**: Este algoritmo forma parte de un conjunto de seis algoritmos propuestos que difieren entre sí en la forma de implementar los dos pasos de los que consta: computación de ítems frecuentes y búsqueda de los mismos.

- **Restricciones adicionales**: Una extensión más de los algoritmos de inferencia de reglas de asociación consiste en añadir restricciones adicionales a la búsqueda, más allá de la cobertura y la confianza, como puede ser por ejemplo especificar la forma de las reglas a encontrar o proporcionar al algoritmo exclusivamente datos considerados relevantes.

En esta sección se han presentado gran parte de los algoritmos existentes que pueden ser utilizados para realizar los distintos procedimientos de Data Mining aplicados a comercio electrónico. Un procedimiento puede ser resuelto por varios algoritmos distintos, y a su vez un mismo algoritmo puede ser utilizado para realizar procedimientos distintos.

Resumen del capítulo

En este capítulo se han introducido los conceptos de Data Mining y KDD, se ha expuesto una clasificación de los distintos procedimientos de minería de datos, y se ha presentado el estado del arte de los algoritmos de Data Mining. En el siguiente capítulo se describen las fases del diseño y el desarrollo del proyecto a lo largo de la duración del mismo, detallando los algoritmos kmedoids y Apriori, el diseño y adaptación de los campos de la base de datos, y el diseño del modelo de datos de la aplicación.