TRABAJO TEÓRICO FINAL

MÁSTER EN CIENCIA DE DATOS

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

Autor
José Ángel Díaz García

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Granada, Enero de 2018

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

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

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

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

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

1.1. Aprendizaje supervisado y no supervisado

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

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

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

1.2. Organización del trabajo

El trabajo está organizado siguiendo como hilo conductor las transparencias de la asignatura 'Minería de datos: Aprendizaje no supervisado y detección de anomalías', del máster en Ciencia de Datos de la Universidad de Granada. Tras este capítulo de introducción encontramos un capítulo donde se profundiza entre la diferencia de las técnicas dirigidas y no dirigidas, para posteriormente pasar al estudio de cada una de estas técnicas, comenzando por el clustering, las anomalías y finalizando con las reglas de asociación.

Capítulo 2

Clustering

Capítulo 3

Detección de anomalías

Capítulo 4

Reglas de Asociación

La reglas de asociación dentro del ámbito de la informática no son muy distintas, al menos en el concepto general, de la búsqueda de relaciones en cualquier ámbito. Las reglas de asociación se enmarcan dentro del aprendizaje automático o minería de datos y no es algo nuevo sino que llevan siendo usadas y estudiadas desde mucho tiempo atrás, datando una de las primeras referencias a estas, del año 1993 [6]. Su utilidad es la de obtener conocimiento relevante de grandes bases de datos y se representan según la forma **X->Y** donde **X**, es un conjunto de items que representa el antecedente e **Y** un item consecuente, por ende, podemos concluir que los ítems **consecuentes** guardan una relación de co-ocurrencia con los ítems **antecedentes**. Esta relación puede ser obvia en algunos casos, pero en otros necesitará del uso de algoritmos de extracción de reglas de asociación que podrán desvelar relaciones no triviales y que puedan ser de mucho valor. Podremos presentar por tanto a las reglas de asociación, como un método de extracción de relaciones aparentemente ocultas entre ítems o elementos dentro de bases de datos transaccionales, *datawarehouses* u otros tipos de almacenes de datos de los que es interesante extraer información de ayuda en el proceso de toma de decisiones de las organizaciones.

4.1. Medidas Clásicas

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

- Soporte: Se representa como *supp (X->Y)*, y representa la fracción de las transacciones que contiene tanto a X como a Y.
- Confianza: Se representa como *conf (X->Y)*, y representa la fracción de transacciones en las que aparece el ítem Y, junto en las que aparece el ítem X.

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

4.2. Obtención de reglas

Si nos centramos en la manera de obtener las reglas, estas pueden abordarse desde dos perspectivas, solución por fuerza bruta (prohibitivo) o desde un enfoque basado en dos etapas. La primera de estas etapas es la generación de itemsets frecuentes, a partir de los cuales, en la segunda etapa se obtienen las reglas de asociación, que tendrán si todo ha ido correctamente un valor de confianza aceptable o elevado. La primera etapa de obtención de itemsets frecuentes puede conllevar problemas de memoria ya que en una base de datos con muchos items o transacciones el número de estos será muy elevado, es por ello que surgen aproximaciones en el proceso de representación de itemsets frecuentes que nos permitirán obtener estos en bases de datos de gran tamaño. Estas aproximaciones son:

- Itemsets maximales: Son aquellos itemsets frecuentes para los que ninguno de los superconjuntos inmediatos al itemsets en cuestión, son frecuentes. A partir de estos podremos recuperar todos los itemsets frecuentes de manera sencilla sin tener que mantenerlos todos en memoria.

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

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

4.3. Principales algoritmos

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

4.3.1. Apriori

El algoritmo **Apriori**, fue propuesto por Agrawal y Srikant en 1994 [7] y desde entonces sigue siendo el algoritmo más extendido para la obtención de itemsets frecuentes, con los que construiremos en una segunda etapa las reglas de asociación. Se basa en el principio de que si un itemset es frecuente, entonces todos sus subconjuntos también lo son por lo que al encontrar uno de estos, podremos podar el árbol de búsqueda evitando hacer comprobaciones y aumentando la eficiencia. Para obtener los itemsets frecuentes, el algoritmo en base a un valor mínimo de soporte fijado por el experto en la materia, generará todas las posibles combinaciones de itemsets y comprobará si son o no frecuentes. En cada iteración, se generan todos los posibles itemsets distintos que se pueden formar combinando los de la anterior, por lo que los itemsets irán creciendo de tamaño.

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

1. Soporte: Umbrales demasiado bajos conllevarán a una explosión del número de itemsets frecuentes lo que está directamente relacionado con una mayor necesidad de memoria y tiempo.
2. Número de ítems distintos: Esta limitación, está ligada a la necesidad del algoritmo apriori de almacenar el soporte de cada uno de éstos, lo que puede conllevar problemas de memoria.
3. Tamaño de la base de datos: Este punto está ligado, al anterior, pero en lugar de tener en cuenta los ítems individuales se tienen en cuenta el número de transacciones. Apriori al ser exhaustivo realiza múltiples pasadas por toda la base de datos por lo que el tiempo de ejecución puede ser muy elevado o incluso no llegar a acabar en varios días o semanas.
4. Longitud de las transacciones: Ligado al problema anterior, si las transacciones a su vez están formadas por muchos ítems, almacenar esto en memoria puede llegar a ser privativo e incluso imposible.

4.3.2. Eclat

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

4.3.3. FP-Growth

El algoritmo **FP-Growth** [8] fue propuesto en el año 2000, como una solución a los problemas de memoria generados por los métodos típicos como el Apriori, visto anteriormente. Es un algoritmo muy eficiente y ampliamente extendido en problemas y soluciones que podrían ser enmarcados bajo el nombre de Big Data.

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

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

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

1. Se extrae la sección del árbol donde aparece el ítem reajustando los valores de soporte de los ítems que aparecen en esa sección.
2. Considerando esa sección extraída, se crea un nuevo ***FP-tree***.
3. Se extraen los itemsets que superen el mínimo soporte de este último ***FP-tree*** creado.

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

Bibliografía

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

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

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

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

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

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

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

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