Tema 6
Actividad 2

Autor
Juan José Méndez Torrero

Comparar dos algoritmos de extracción de patrones secuenciales usando SPMF

Introducción

En esta actividad se tendrán en cuenta un total de 4 conjuntos de datos, Bible, Leviathan, MSNBC y Sign. Todos estos conjuntos de datos se pueden descargar en la página oficial de SPMF.

Además, se han elegido los algoritmos PrefixSpan y SPAM para realizar la comparación de ambos algoritmos para la extracción de patrones secuenciales frecuentes. Además, ambos algoritmos se ejecutarán con un soporte mínimo del 10%, para poder así comparar la eficacia de ambos algoritmos.

Conjuntos de datos

La siguiente tabla muestra el nombre del conjunto de datos, el número total de secuencias con las que cuenta el conjunto de datos y por último, el número de ítems con el que cuenta cada conjunto de datos.

Conjunto de datos	Número de secuencias	Número de items
Bible	36369	13905
Leviathan	5834	9025
MSNBC	989818	17
Sign	800	267

Algoritmo PrefixSpan

En esta sección se muestran los resultados tras ejecutar el algoritmo PrefixSpan sobre cada uno de los conjuntos de datos. Para ello, se tendrá en cuenta el tiempo que ha tardado en ejecutarse el algoritmo y el número de patrones secuenciales frecuentes encontrados. La Tabla 1 muestra estos valores para cada uno de los conjuntos de datos seleccionados.

Conjunto de datos	Tiempo de ejecución	Número de secuencias frecuentes
Bible	~ 831 ms	174
Leviathan	~ 291 ms	651
MSNBC	~ 511 ms	338
Sign	~ 3301 ms	110417

Tabla 1

Algoritmo SPAM

En esta sección se muestran los resultados tras ejecutar el algoritmo SPAM sobre cada uno de los conjuntos de datos. Para ello, se tendrá en cuenta el tiempo que ha tardado en ejecutarse el algoritmo y el número de patrones secuenciales frecuentes encontrados. La Tabla 2 muestra estos valores para cada uno de los conjuntos de datos seleccionados.

Conjunto de datos	Tiempo de ejecución	Número de secuencias frecuentes
Bible	~ 4046 ms	174
Leviathan	~ 1045 ms	651
MSNBC	~ 1667 ms	338
Sign	~ 14603 ms	110417

Tabla 2

Conclusiones

Como se puede observar, ambos algoritmos han encontrado el mismo número de patrones frecuentes, pero el tiempo de ejecución entre ambos ha sido bastante diferente, siendo en ocasiones hasta 4 veces más lento el algoritmo SPAM que el algoritmo PrefixSpan.

Esto puede ser debido a la forma en que cada uno de los algoritmos encuentra esos patrones secuenciales frecuentes, mientras que PrefixSpan está basado en el crecimiento de patrones, el algoritmo SPAM convierte la base de datos en una representación vertical, y esto hace que, aunque sólo haya que leer una vez la base de datos, el algoritmo tarde más en crear esa representación que en leer múltiples veces la base de datos inicial, como en el caso de PrefixSpan.