SEKVENČNÉ PRAVIDLÁ

Sekvenčné pravidlá sú odvodené od asociačných, rozdiely preto nie sú príliš veľké. *K* - sekvencia je sekvencia dĺžky *k*, t. j. obsahuje *k* stránok. Frekventovaná *k* - sekvencia je obdobou frekventovanej *k* - položkovej množiny, resp. kombinácie. Zvyčajne, frekventovaná 1 - položková množina je rovnaká ako frekventovaná 1 - sekvencia.

Na nasledujúcom príklade naznačíme rozdiely medzi algoritmom Apriori a AprioriAll, kde algoritmus Apriori slúži na hľadanie asociačných pravidiel a AprioriAll na hľadanie sekvenčných pravidiel:

$D = \{T1 = \{U1, <A, B, C>\}, T2 = \{U2, <A, C>\}, T3 = \{U1, <B, C, E>\}, T4 = \{U3, <A, C, D, C, E>\}\}$,

kde *D* je databáza transakcií s časovou nálepkou, *A, B, C, D, E* sú webové časti a *U*1, *U*2, *U*3 sú používatelia. Každá transakcia je identifikovaná používateľom. Predpokladajme, že minimálna podpora (*min support*) je 30%.

V tomto príklade má používateľ *U*1 aktuálne dve transakcie. Pri hľadaní sekvenčných pravidiel uvažujeme jeho sekvenciu ako aktuálne spojenie webových častí v transakciách *T*1 a *T*3, t. j. sekvencia môže pozostávať z viacerých transakcií, pričom sa nevyžadujú súvislé prístupy na stránky. Rovnako podpora sekvencie je určená nie percentom transakcií, ale percentom používateľov, ktorý majú danú sekvenciu. Sekvencia je veľká/frekventovaná ak sa prinajmenšom nachádza v jednej sekvencii identifikovanej používateľom a spĺňa podmienku minimálnej podpory. Prvým krokom je zoradenie transakcií podľa používateľa s časovou nálepkou každej ním navštívenej stránky, zostávajúce kroky sú podobné ako v algoritme Apriori. Po zoradení sme získali aktuálne sekvencie identifikované používateľom, ktoré predstavujú kompletné odkazy od jedného používateľa:

$D = \{T1 = \{U1, <A, B, C>\}, T3 = \{U1, <B, C, E>\}, T2 = \{U2, <A, C>\}, T4 = \{U3, <A, C, D, C, E>\}\}$.

Podobne ako v algoritme Apriori začíname generovaním množiny kandidátov dĺžky 1: $C1 = \{<A>, <B>, <C>, <D>, <E>\}$, z nej určíme množinu $L1 = \{<A>, <B>, <C>, <D>, <E>\}$ jednoprvkových sekvencií takých, kde každá stránka je odkazovaná prinajmenšom jedným používateľom.

Následne množiny kandidátov *C*2 vygenerujeme z *L*1 tzv. úplným spájaním, t. j. zohľadňujeme, že používateľ webu prehľadáva stránky dopredu alebo dozadu. Z tohto dôvodu algoritmus Apriori nie je vhodný na objavovanie znalostí z webových prístupov (web log mining), na rozdiel od algoritmu AprioriAll, ktorý spomínanú skutočnosť zohľadňuje. Z množiny kandidátov dĺžky 2: $C2 = \{<A, B>, <A, C>, <A, D>, <A, E>, <B, A>, <B, C>,$

<B, D>, <B, E>, <C, A>, <C, B>, <C, D>, <C, E>, <D, A>, <D, B>, <D, C>, <D, E>, <E, A>, <E, B>, <E, C>, <E, D>}, určíme množinu *L2 = {<A, B>, <A, C>, <A, D>, <A, E>, <B, C>, <B, E>, <C, B>, <C, D>, <C, E>, <D, C>, <D, E>*} dvojprvkových sekvencií takých, kde sa každá sekvencia prinajmenšom nachádza v jednej sekvencii identifikovanej používateľom. Analogicky postupujeme ďalej.