SEQUENTIAL PATTERN MINING

Zdeněk Tisoň

Master Degree Programme (3), FIT BUT

E-mail: xtison00@stud.fit.vutbr.cz

Supervised by: Martin Hlosta

E-mail: ihlosta@fit.vutbr.cz

Abstract: Sequential pattern mining aims to find frequent subsequences in a database of sequences. This paper deals with algorithms for mining these patterns like GSP, PrefixSpan and LAPIN-SPAM. As the support decreases the number of sequential patterns will increase rapidly. Possible solution is to mine only closed sequential patterns. Mining algorithms were implemented and their efficiency was tested on different types of data sets.

Keywords: data mining, sequential pattern, closed sequences, GSP, PrefixSpan, LAPIN-SPAM

1. ÚVOD

V dnešní moderní době znalost informací znamená konkurenční výhodu. Každý náš nákup, přístup na webovou stránku nebo návštěva lékaře produkují data, která jsou nejrůznějšími společnostmi zaznamenávána. A právě přeměnou těchto dat na nové znalosti se mohou organizace stát konkurenceschopné. Tradiční databázové systémy však nenabízí možnost extrakce nových informací z uložených dat. Proto bylo zapotřebí vyvinout nový obor, který se tímto zabývá.

Získávání znalostí z databází je obor, který tento nedostatek řeší. Získávání znalostí z databází si klade za cíl vyhledat z velkého množství dat skryté, netriviální a užitečné informace. Typickou aplikací tohoto oboru je analýza nákupního košíku. Ta přináší informace o tom, jaké zboží si zákazníci nejčastěji nakupují dohromady. Možným rozšířením tohoto příkladu je zjištění, v jakém pořadí si zákazníci dané zboží koupí. Tyto znalosti jsou reprezentovány pomocí *sekvenčních vzorů*.

Sekvenční vzor je seřazená posloupnost událostí, která se v analyzovaných datech vyskytuje frekventovaně. Událost může být například návštěva lékaře, bezpečnostní průnik nebo přístup na webovou stránku. Získané sekvenční vzory mají převážně popisný charakter, mohou se však využít i pro předpověď budoucích hodnot. Například zkoumáním symptomů a nemocí pacientů můžeme v brzké fázi díky prvním příznakům detekovat a odvrátit vážnou hrozící chorobu.

2. DOLOVÁNÍ SEKVENČNÍCH VZORŮ

Sekvence je časově seřazená posloupnost událostí, u nichž může, ale také nemusí být explicitně zaznamenán čas výskytu. Každá událost je složena z množiny položek. Příkladem databáze sekvencí jsou DNA sekvence, webové logy nebo nákupy zákazníků v obchodě. Jednoduchou databázi sekvencí zobrazuje Tabulka 1. Například zákazník s identifikátorem *id=100* si během prvního nákupu v obchodě koupil společně počítač, kameru a myš. Následně pak nakoupil tiskárnu a usb kabel.

Dolování sekvenčních vzorů je hledání frekventovaných podsekvencí v databázi sekvencí, jejichž podpora je větší než uživatelem zadaná minimální podpora. Podpora podsekvence je daná počtem sekvencí v databázi, které danou podsekvenci obsahují. Například pro uživatelem zadanou minimální podporu *min_support=2* je podsekvence < (*počítač*, *kamera*) (*tiskárna*) > sekvenčním vzorem. Tento vzor říká, že zákazník, který si někdy společně koupí počítač a kameru, si pravděpodobně v budoucnu koupí i tiskárnu.

id zákazníka	sekvence
100	< (*počítač*, *kamera*, *myš*) (*tiskárna*, *usb kabel*) >
200	< (*monitor*) (*počítač*, *kamera*, *paměť*) (*disk*, *tiskárna*) >
300	< (*počítač*, *kamera*) (*tablet*) (*mobil*) (*tiskárna*) >

Tabulka 1: Příklad databáze sekvencí nákupů zboží v obchodě.

3. ALGORITMY PRO DOLOVÁNÍ SEKVENČNÍCH VZORŮ

Tato kapitola představuje algoritmy pro dolování úplné množiny sekvenčních vzorů. Algoritmy se odlišují ve způsobu generování kandidátů a počítaní podpory kandidátů. Kapitola popisuje algoritmy založené na metodách generování a testování, postupného vzrůstu vzoru a brzkého ořezávání prohledávacího prostoru. Více informací o následujících algoritmech lze nalézt v literatuře [1, 2].

3.1. GSP (GENERALIZED SEQUENTIAL PATTERN)

Algoritmus GSP je založený na metodě *generování a testování* kandidátních sekvencí. Tato metoda v každém běhu spojuje frekventované sekvence z předchozího běhu, pro vytvoření nových, o jeden prvek delších, kandidátů. Následným průchodem databází se určí jejich frekventovanost. Pro urychlení dolování je zde využita *Apriori* vlastnost. Apriori vlastnost říká, že sekvence je nefrekventovaná, pokud obsahuje nějakou nefrekventovanou podsekvenci. GSP během dolování generuje obrovské množství kandidátních sekvencí, spotřebovává mnoho paměti a vyžaduje vícenásobný průchod databází. Z těchto důvodů nemá dostačující výkon při dolování nad velkými datovými sadami.

3.2. PREFIXSPAN (PREFIX-PROJECTED SEQUENTIAL PATTERN GROWTH)

PrefixSpan je efektivní algoritmus pro dolování frekventovaných sekvencí metodou postupného *vzrůstu vzoru*. Hlavní předností algoritmu je, že nevyužívá princip generování a testování kandidátních sekvencí. Namísto toho PrefixSpan pomocí prefixů rekurzivně projektuje databázi sekvencí do více projektovaných databází. Projektované databáze jsou zpravidla mnohem menší než původní databáze, čímž dochází k redukci prohledávácího prostoru a k urychlení procesu dolování. Experimentálně bylo ukázáno, že algoritmus PrefixSpan je mnohem rychlejší než GSP a dokáže si poradit i s obrovskými datovými sadami.

3.3. LAPIN-SPAM (LAST POSITION INDUCTION SEQUENTIAL PATTERN MINING)

Algoritmus LAPIN-SPAM pro urychlení dolování využívá metodu *brzkého ořezávání* vstupního prostoru. Tato metoda staví na myšlence *indukce pozice*. Ta říká, že pokud je poslední pozice položky menší než aktuální pozice prefixu, položka se již nemůže vyskytovat za aktuálním prefixem v téže sekvenci. Díky této heuristice a *bitmapové reprezentaci* databáze LAPIN-SPAM patří mezi nejefektivnější algoritmy pro dolování dlouhých sekvenčních vzorů.

4. UZAVŘENÉ SEKVENČNÍ VZORY

Dolování úplné množiny frekventovaných podsekvencí může generovat obrovské množství výsledných vzorů. Jednou z možností je využít bezztrátové komprese v podobě dolování podmnožiny těchto vzorů a to tzv. *uzavřených sekvenčních vzorů*. Sekvence je uzavřená, když v databázi neexistuje její nadsekvence se stejnou podporou. Díky této kompaktnosti lze navíc využít různé heuristiky pro rychlejší nalezení těchto vzorů.

Momentálně nejefektivnějším algoritmem pro dolování uzavřených sekvencí je BIDE [3]. Tento algoritmus využívá efektivní techniky pro určení uzavřenosti vzorů a prořezání vstupního prostoru pomocí metody *BackScan*. Při porovnání algoritmu BIDE s algoritmy, které hledají úplnou množinu sekvenčních vzorů, je BIDE na reálných datech časově i paměťově efektivnější.

5. IMPLEMENTACE ALGORTIMŮ

V rámci diplomové práce byly implementovány algoritmy PrefixSpan, LAPIN-SPAM a BIDE. Pro každou rodinu algoritmů (vzrůst vzoru, brzké ořezávání, uzavřené sekvenční vzory) byl na základě analýzy algoritmů pro implementaci vybrán nejefektivnější algoritmus, který danou rodinu zastupuje. Zmíněné algoritmy byly implementovány v prostředí Microsoft Analysis Services (MSAS) v programovacím jazyce C#. MSAS je komponenta Microsoft SQL Serveru, která se využívá pro OLAP analýzu a dolování z dat.

Algoritmy byly vytvořeny pro účely projektu TAČR. Tento projekt se zaměřuje na analýzu škodlivého kódu a jedním z typů analýzy je dolování sekvenčních vzorů. V rámci projektu byl již dříve v prostředí MSAS vytvořen algoritmus GSP. Vstupní databáze sekvencí obsahuje obrovské množství dat, proto bylo zapotřebí implementovat efektivnější algoritmy pro analýzu skrytých sekvenčních vzorů.

Během testování algoritmů v prostředí MSAS se ukázalo, že nejvíce procesorového času zabírá načítání záznamů z databáze. Z důvodu, že algoritmus GSP prochází databázi mnohokrát, je jeho výkon nedostačující při dolování dlouhých vzorů. Algoritmy PrefixSpan, LAPIN-SPAM a BIDE prochází vstupní databázi pouze dvakrát, a proto vyhledávají sekvenční vzory mnohem rychleji. Z důvodů, že MSAS neposkytuje data dostatečnou rychlostí, byly zmíněné algoritmy pro snadnější testování implementovány nezávisle na tomto prostředí.

6. EXPERIMENTY

Výkon implementovaných algoritmů byl otestován na reálných a syntetických datech. Experimenty ukázaly, že PrefixSpan a BIDE mají podobný výkon a hodí se především pro dolování nad řídkými daty, tedy takovými, která mají mnoho unikátních položek a kratší sekvence. Algoritmus BIDE byl na reálných datech efektivnější v případě, když data obsahovala uzavřené sekvence. Vždy však spotřebovával nejméně paměti ze všech implementovaných algoritmů. Na druhou stranu LAPIN-SPAM byl nejrychlejší při dolování nad hustými daty, tedy takovými, která obsahují relativně málo položek a dlouhé sekvence, jako jsou DNA sekvence. Spotřeba paměti byla však v případě algoritmu LAPIN-SPAM zpravidla největší.

7. ZÁVĚR

Tento příspěvek představuje sekvenční vzory a jejich různé aplikace. Jsou zde popsány algoritmy, které se využívají pro hledání úplné množiny sekvenčních vzorů. Z důvodu velkého množství výsledných vzorů se článek také zabývá dolováním kompaktnější podoby těchto dat, tzv. uzavřenými sekvenčními vzory. Algoritmy byly v rámci projektu implementovány v prostředí MS Analysis Services. Byly provedeny experimenty pro vyhodnocení vhodnosti algoritmů pro různá data.

PODĚKOVÁNÍ

Tento příspěvek vznikl za podpory výzkumného záměru MSM0021630528, grantů TAČR TA01010858 a FIT-S-11-2 a Centra excelence IT4Innovations CZ.1.05/1.1.00/02.0070.

REFERENCE

[1] HAN, J., KAMBER, M. *Data Mining: Concepts and Techniques*. Second Edition. Morgan Kaufmann Publishers, 2006. 770 s. ISBN 1-55860-901-6.

[2] MABROUKEH, M., EZEIFE, C. A taxonomy of sequential pattern mining algorithms. In: *ACM Computing Surveys*: *2010*. Volume: 43, Issue: 1. ACM, 2010.

[3] WANG, J., HAN, J. BIDE: Efficient Mining of Frequent Closed Sequences. In: *Proceedings of the 20th International Conference on Data Engineering*. ISBN 0-7695-2065-0.