DATAMINING IN MS SQL USING INCREMENTAL ALGORITHMS

Lukáš David
Master Degree Programme (2), FIT VUT
E-mail: xdavid00@stud.fit.vutbr.cz

Supervised by: Michal Šebek
E-mail: isebek@fit.vutbr.cz

Abstract: This work deals with issues in data stream mining which is nowadays a very dynamic area of information technology. The paper gives a brief introduction to data mining data streams. The last section provides a final draft of the implementation of the chosen algorithm technology on Microsoft SQL server.

Keywords: Microsoft SQL server, CluStream, StreamInsight Server

1 ÚVOD

V posledních letech dochází k obrovskému rozmachu Internetu, mobilních technologií, cloudových technologií atp. Stále více aktivit spojených s každodenním životem probíhá skrze Internet, případně jiné informační subjekty. Tyto aktivity pak tvoří nekonečný proud dat, který je potřeba zpracovávat. Cílem příspěvku je uvést čtenáře do problematiky získávání znalostí z proudu dat a možnosti implementace jednoho z algoritmů pro získávání znalostí nad technologií Microsoft SQL Serveru. V následující části textu je uveden stručný úvod k proudům dat a získávání znalostí z proudů dat. Posléze je uveden vlastní návrh řešení vybraného algoritmu nad danou technologií.

2 ZÍSKÁVÁNÍ ZNALOSTÍ A PROUDY DAT

V dnešní době jsme doslova zahlceni obrovským množstvím různorodých informací. Jedná se např. o podnikové databáze, bankovní transakce, prodeje zboží atp. Proces získávání znalostí nazývaný také *dolování z dat* je proces, který se snaží v těchto datech vydolovat nějakou netriviální, zajímavou a pro naše účely užitečnou informaci případně znalost. Netriviální informací je myšlena taková informace, kterou nelze získat například pouhým SQL dotazem, nicméně je potřeba použít nějaký sofistikovaný přístup.

2.1 PROUDY DAT

Proudem dat chápeme potenciálně nekonečný datový tok, který do systému přichází a zároveň z něj může odcházet. Rychlost tohoto toku se v čase může měnit. Jako příklad můžeme uvést prohlížení webových stránek, kreditní operace, provoz na síti, vyhledávaná slovních spojení ve vyhledávačích atp. Vzrůstá tedy neustále množství proudových dat a s tím se zákonitě zvětšuje i tlak na jejich efektivní zpracování a dolování zajímavých a relevantních informací z těchto dat.
Dolování v proudu dat se od klasického dolování nad relační databází liší v několika směrech:

- Některá, případně všechna data, nejsou dostupná pro náhodné čtení z pevného disku či paměti, ale přichází z jednoho čí více datových proudů.
- Datový proud má potenciálně neomezenou velikost.
- Rychlost datového proudu se v čase mění a může dosahovat místy extrémních hodnot.

Z výše uvedených vlastností plyne několik zásadních omezení pro algoritmy dolující z proudu dat [1]:

- Při zpracování většího množství dat není možné tato data zpracovávat víceprůchodově. Většina algoritmů by tedy měla datový tok zpracovávat jednoprůchodově.
- Vzhledem k tomu, že zpracováváme data, která se postupně vyvíjí v čase, musí být algoritmy navrženy tak, aby byly schopné zpracovávat takto se vyvíjející data. Čas je přirozenou součástí většiny dolovacích algoritmů u proudů dat.

Vzhledem k výše uvedenému se pro zpracování proudu dat používají *inkrementální* algoritmy, neboť ty zpracovávají vstupní data postupně na rozdíl od *dávkových* algoritmů.

2.2 SHLUKOVÁNÍ A ALGORITMUS CLUSTREAM

Pro implementaci vlastní práce byl vybrán shlukovací algoritmus *CluStream*. Shlukování je proces, který hledá množiny (třídy) objektů, které mají co nejvíce společného a zároveň se od ostatních tříd co nejvíce liší. Výběr algoritmu byl dán tím, že algoritmus má velmi dobré vlastnosti v porovnání s ostatními shlukovacími algoritmy (*BIRCH* a *STREAM*). Výhodou tohoto algoritmu je větší přesnost a rychlost, dále možnost vytvářet vysokoúrovňové shluky pro historii dat libovolné délky [2]. Algoritmus je založen na Micro-clusterech (dále jen MC).

Definice 2.1. [1] *MC* pro množinu *d-dimensionálních* bodů X_{i_1}, X_{i_2}, .. , X_{i_n} s časovými razítky T_{i_1}, T_{i_2}, .., T_{i_n} je $(2*d+3)$ n-tice $\left(\overline{CF2^x}, \overline{CF1^x}, CF2^t, CF1^t, n\right)$, kde $\overline{CF2^x}$ a $\overline{CF1^x}$ odpovídají vektoru d záznamů. Definice každé z těchto položek je následující:

- $\overline{CF2^x}$ – vektor sum druhých mocnin hodnot jednotlivých dimenzí všech vektorů, kde p-tý prvek je tedy roven $\sum_{j=1}^{n}\left(x_{i_j}^p\right)^2$.
- $\overline{CF1^x}$ – vektor sum hodnot jednotlivých dimenzí všech vektorů, kde p-tý prvek prvek je tedy roven $\sum_{j=1}^{n} x_{i_j}^p$.
- $\overline{CF2^t}$ – suma druhých mocnin hodnot všech časových razítek T_1,T_2, .., T_n.
- $\overline{CF1^t}$ – suma hodnot všech časových razítek T_1,T_2, .., T_n.
- n – počet zpracovávaných vektorů.

Volba právě takovéto struktury jednotlivých MC vychází z vlastního principu algoritmu. Algoritmus udržuje neustále určitý počet MC $\mathcal{M}_1$,$\mathcal{M}_2$, ... , $\mathcal{M}_q$. Objeví-li se nový bod $\overline{X_{i_k}}$ v proudu, musí dojít k absorpci tohoto bodu do některého z MC, případně k vytvoření nového micro-clusteru, určeného právě tímto příchozím bodem. Nejprve tedy dojde k nalezení MC $\mathcal{M}_p$, jenž má nejmenší vzdálenost od dobu $\overline{X_{i_k}}$. Posléze se vypočte rozptyl MC $\mathcal{M}_p$ pomocí vektorů $\overline{CF2^x}$ a $\overline{CF1^x}$, který určuje hodnotu hranice. Leží-li bod uvnitř daných hranic, je přidán do nalezeného MC. Leží-li vně, je nutné vytvořit nový MC. Pokud algoritmus vytvořil nový MC, musí odstranit jeden ze stávajících, případně dva spojit dohromady. Nejdříve algoritmus hledá MC, který může odstranit. Zde se používají pro výpočet hodnot průměru a směrodatné odchylky časová razítka daného MC. Posléze se hledá čas příchodu bodu odpovídajícímu *m/(2*n)-tému* percentilu bodů v MC za předpokladu, že časy jednotlivých příchodů mají normální rozložení. Tato hodnota se značí jako *relevantní razítko*. MC s nejmenší takovou hodnotou se odstraní, je-li tato hodnota menší než uživ. definovaný práh. Pokud je větší, k odstranění nedojde a musí se sloučit dva MC do jednoho. Zde se vyberou dva nejbližší MC a ty se sloučí. Tento proces se posleze opakuje pro každý další příchozí bod.

3 NÁVRH ŘEŠENÍ

Návrh řešení vychází z předpokladu, že proudem dat nad MS SQL jsou příchozí transakce, které vkládají nová data do jednotlivých tabulek. Nad těmito novými daty bude prováděno shlukování dle vybraného algoritmu. Zpracování tohoto proudu dat bude prováděno dvěma způsoby. Prvním způsobem bude využití platformy Microsoft StreamInsight. Druhým způsobem zpracování bude implementace vlastního rozhraní pro zpracování tohoto proudu dat. Jedním z výsledků práce bude posléze porovnání efektivnosti zpracování proudu dat pomocí výše uvedených způsobů. Výsledná aplikace se bude skládat ze sedmi balíčků. Popis jednotlivých balíčků je uveden zde:

- *MS SQL* – balíček reprezentující třídy, které poběží na serveru za využití CLR Integration. Tyto třídy budou sledovat změny v databázi (proud dat) a příslušná data budou posílat dál ke zpracování. Data budou vybírána dle nastavení z jednotlivých tabulek, pohledů atp. a dále odesílána pomocí socketů.
- *StreamInsight Server* – třídy balíčku budou využity k vlastnímu zpracování proudu dat. Pomocí vstupního adaptéru bude připojen na příslušný server, jehož data bude číst. Tato data bude posléze pomocí dotazovací logiky a jazyka LINQ upravovat (čištění dat atp.) a posílat k dalšímu zpracování. Využití této technologie pro dané řešení není přímo nutné, neboť síla této technologie spočívá zejména ve zpracování většího množství vstupních dat z více zdrojů, jejich souběžné zpracování a poskytnutí výsledků jednotlivým výstupním zařízením. Nicméně její využití sebou přináší možnost snadného rozšíření aplikace o zpracování proudových dat i z jiných typů zdrojů, případně více MS SQL serverů zároveň. Toto u druhého řešení nebude možné, neboť to bude uzpůsobeno přímo pro tuto úlohu. Dále tato cesta nabízí zajímavou možnost srovnání efektivnosti obou řešení. Balíček bude také implementovat některé abstraktní třídy balíčku API.
- *MicroStream Server* – jedná se o balíček, jehož úkoly budou stejné jako u předchozího balíčku. Rozdíl bude v zpracování proudu dat, zde nebudeme moci využít síly jazyka LINQ a data budou zpracovávána běžnými řídícími strukturami (if, for atp.).
- *API* – jak již plyne z názvu, balíček bude obsahovat abstraktní třídy, reprezentující rozhraní mezi balíčkem MicroClusters a balíčky MicroStream Server a StreamInsight Server.
- *MicroClusters* – tento balíček tvoří jádro celé aplikace. Bude obsahovat třídy implementující hlavní část algoritmu CluStream, a to je správa Micro-clusterů.
- *MacroClusters* – balíček, jehož součástí budou třídy, které na základě uživatelského vstupu vytvoří vysoko-úrovňové shluky pro vlastní prezentaci uživateli.
- *GUI* – slouží k ovládání aplikace a k vizualizaci výsledků shlukování. Bude umožňovat volbu nastavení pro výpočet shluků, volbu použitých metrik pro výpočet vzdálenosti atributů jednotlivých typů atp.

4 ZÁVĚR

V tomto příspěvku byl uveden lehký úvod do oblasti získávání znalostí z proudu dat. Uvedl jsem zde návrh řešení výsledné implementace algoritmu CluStream pro dolování v proudu dat nad Microsoft SQL serverem. Jedná se o novou funkcionalitu v rámci tohoto prostředí, která bude dále využívána na FIT VUT.

PODĚKOVÁNÍ

Tento příspěvek vznikl za podpory grantu FIT-S-11-2, výzkumného záměru MSM0021630528 a Centra excelence IT4 Innovations CZ.1.05/1.1.00/02.0070.

REFERENCE

[1] Aggarwal, C. C.: *Data Streams: Models and Algorithms*, New York, Springer Science+Business Media, 2007. ISBN-10: 0-387-28759-0.

[2] Aggarwal, C. C., Han, J., Wang, J. et al. A framework for clustering evolving data streams. In *Proceedings of the 29th international conference on Very large data bases - Volume 29*. VLDB Endowment, 2003. S. 81–92. VLDB '03. ISBN 0-12-722442-4.