MECHANIZM PERSPEKTYW MATERIALIZOWANYCH W EKSPLORACJI DANYCH

Mikołaj MORZY, Marek WOJCIECHOWSKI

Streszczenie: Eksploracja danych to proces interaktywny i iteracyjny. Użytkownik definiuje zbiór interesujących go wzorców określając eksplorowany zbiór danych i wybierając konkretne wartości parametrów eksploracji. Jest bardzo prawdopodobne, że w celu uzyskania satysfakcjonujących go wyników użytkownik wielokrotnie dokona eksploracji, za każdym razem nieznacznie zmieniając eksplorowany zbiór danych lub modyfikując parametry algorytmu. Aktualnie dostępne algorytmy eksploracji danych charakteryzują się długim czasem przetwarzania, wprost proporcjonalnym do rozmiaru analizowanych danych. Ponieważ eksploracja odbywa się najczęściej w środowisku magazynu danych, długie czasy przetwarzania są nie do przyjęcia z punktu widzenia interaktywnej eksploracji. Z drugiej strony wyniki kolejnych, następujących po sobie zapytań użytkownika są bardzo zbliżone. Jednym z rozwiązań problemu długich czasów przetwarzania zapytań eksploracyjnych jest wykorzystanie zmaterializowanych wyników wcześniejszych zapytań. W tym artykule przedstawiamy koncepcję materializowanych perspektyw eksploracyjnych i sposoby wykorzystania takich perspektyw w przetwarzaniu zapytań eksploracyjnych. Pokazujemy, w jaki sposób mechanizm ten może wydatnie przyśpieszyć proces odkrywania reguł asocjacyjnych lub wzorców sekwencji. Wskazujemy też dalsze kierunki badań w tym zakresie.

1. Wstęp

Eksploracja danych, zwana także odkrywaniem wiedzy w bazach danych (ang. *data mining*, *knowledge discovery in databases*) to proces odkrywania nowych, nieznanych, pożytecznych i zrozumiałych wzorców w dużych wolumenach danych [12]. W ostatnich latach obserwujemy wyraźne odchodzenie od dedykowanych i wyspecjalizowanych systemów eksploracyjnych i dążenie do integracji tych systemów z istniejącymi systemami zarządzania bazami danych. Integracja ta przebiega przede wszystkim w ramach magazynów danych (ang. *data warehouses*), które stanowią doskonałe źródło danych dla różnych technik eksploracyjnych. Z punktu widzenia użytkownika wykonanie algorytmu i odkrycie zbioru wzorców to rodzaj odpowiedzi na zaawansowane zapytanie do bazy danych. Użytkownik określa zbiór eksplorowanych danych (np. za pomocą standardowego zapytania wyrażonego w języku SQL) oraz wyznacza wartości współczynników sterujących danym algorytmem odkrywania wzorców. W odpowiedzi system wykonuje odpowiedni algorytm i prezentuje użytkownikowi uzyskany zbiór wzorców.

Użytkownik z reguły nie zna dokładnego celu eksploracji, lecz dochodzi do

interesujących i satysfakcjonujących go wyników w wielu następujących po sobie krokach. W każdym kroku użytkownik weryfikuje zbiór uzyskanych wzorców i stosownie do swych potrzeb i oczekiwań zmienia zbiór eksplorowanych danych (modyfikując odpowiednie polecenie języka SQL), lub dostraja algorytm odkrywania wzorców przez zmiany wartości parametrów. Praktyka wykazuje, że w typowym procesie odkrywania wzorców użytkownik wykonuje wiele razy ten sam algorytm z nieznacznie zmienionymi parametrami. Z drugiej strony często zdarza się, że użytkownik wydaje dane zapytanie eksploracyjne okresowo, np. raz w tygodniu, w celu znalezienia najbardziej aktualnych wzorców. W takim wypadku system powinien móc przechować wyniki poprzedniej eksploracji i spróbować udzielić odpowiedzi w sposób przyrostowy, bazując na wcześniejszych wynikach i uwzględniając zmiany, jakie zaszły w bazie danych od czasu ostatniej eksploracji. W przypadku magazynu danych wolumen zmian stanowi zazwyczaj znikomą część oryginalnej bazy danych.

Podstawowym problemem, jaki napotyka się podczas eksploracji danych, jest czas przetwarzania typowego zapytania eksploracyjnego. Algorytmy odkrywania reguł potrzebują minut i godzin aby odpowiedzieć na stosunkowo proste zapytanie. Często też rozmiar odpowiedzi przekracza rozmiar eksplorowanej bazy danych. Taka charakterystyka procesu odkrywania wiedzy czyni go zupełnie niezdatnym do zastosowań interaktywnych i iteracyjnych.

Jednym z rozwiązań powyższego problemu jest zastosowanie perspektyw materializowanych. Materializacja wyników zapytań eksploracyjnych może się odbywać automatycznie, lub na żądanie użytkowników. System eksploracyjny powinien umożliwiać wykorzystanie tych wyników i włączenie ich do algorytmu odkrywania wzorców. Mechanizm perspektyw materializowanych został dokładnie zbadany i z powodzeniem wykorzystany w tradycyjnych systemach relacyjnych baz danych. Proponujemy, aby podobne rozwiązanie zastosować w przypadku systemów odkrywania wiedzy.

W artykule pokazujemy, w jaki sposób można wykorzystać wyniki wcześniejszych zapytań eksploracyjnych do skrócenia czasu przetwarzania w przypadku odkrywania zbiorów częstych, reguł asocjacyjnych i wzorców sekwencyjnych. Eksperymenty dowodzą, że wykorzystanie wcześniejszych wyników wielokrotnie skraca czas wykonania zapytania. Jednakże, w przypadku zapytań eksploracyjnych określenie perspektyw materializowanych, które mogą być wykorzystane do odpowiedzi na dane zapytanie, nie jest proste. Perspektywa eksploracyjna może się różnić od danego zapytania nie tylko wartościami współczynników wykonania algorytmu, ale również schematem eksplorowanej bazy danych. Poniżej pokazujemy, w jakich przypadkach można wykorzystać perspektywę materializowaną do odpowiedzi na zapytanie eksploracyjne i jakie dodatkowe czynności są konieczne, aby zwrócona odpowiedź była poprawna. Przykłady przedstawione w artykule zostały wyrażone za pomocą języka MineSQL, deklaratywnego języka do eksploracji danych opartego na języku SQL i rozwijanego od kilku lat w Instytucie Informatyki Politechniki Poznańskiej [14].

2. Podstawowe definicje

2.1. Perspektywy zwykłe i materializowane

Perspektywa to wywiedziona tabela, zdefiniowana w oparciu o tabele bazowe. Perspektywa definiuje funkcję ze zbioru tabel bazowych do tabeli wywiedzionej. Funkcja ta jest zazwyczaj obliczana przy każdym odwołaniu do perspektywy. Perspektywa może zostać zmaterializowana poprzez składowanie krotek perspektywy w bazie danych. Na materializowanej perspektywie można zakładać indeksy, przez co dostęp do tak zapisanych danych może być dużo szybszy niż ponowne wyliczenie perspektywy. Perspektywa materializowana w pewnym sensie przypomina pamięć podręczną - jest kopią danych którą można szybko odczytać. Materializacja perspektywy eliminuje potrzebę ponownego obliczania i rozwijania perspektywy przy każdym dostępie do niej.
Zawartość materializowanej perspektywy staje się nieaktualna w momencie modyfikacji tabel bazowych, z których wywiedziono perspektywę. Proces modyfikowania perspektywy materializowanej w odpowiedzi na zmiany zachodzące w tabelach bazowych nazywamy pielęgnacją perspektywy. Często zmiany w tabelach bazowych powodują zmianę tylko w części perspektywy. Pielęgnacja perspektywy przez wyliczanie całej jej zawartości byłaby w takiej sytuacji marnotrawstwem. Znacznie prościej i szybciej jest wyznaczyć zmianę zachodzącą w perspektywie materializowanej tylko na podstawie zmian jakie zaszły w tabelach bazowych. Taką pielęgnację nazywamy pielęgnacją przyrostową (inkrementalną). Należy jednak pamiętać, że w przypadku wielu typów perspektyw materializowanych pielęgnacja przyrostowa jest niemożliwa.

2.2. Zbiory częste

Niech $L=\{l_1,l_2,...,l_n\}$ będzie zbiorem literałów zwanych elementami. Niech D będzie kolekcją transakcji, gdzie każda transakcja jest dowolnej długości i $\forall T \in D$ $T \subseteq L$. Mówimy, że transakcja T wspiera element x jeśli $x \in T$. Mówimy, że transakcja T wspiera zbiór X, jeśli T wspiera każdy element $x \in X$. *Wsparciem* (ang. support) zbioru X nazywamy stosunek liczby transakcji wspierających X do liczby wszystkich transakcji.

$$support(X, D) = \frac{\left|\{T \in D : T \text{ wspiera } X\}\right|}{|D|}$$

Problem odkrywania zbiorów częstych polega na znalezieniu w danej bazie danych D wszystkich zbiorów, których wsparcie jest wyższe od zdefiniowanej przez użytkownika wartości, zwanej minimalnym wsparciem (*minsup*). Zbiór, którego wsparcie jest wyższe niż *minsup* nazywamy **zbiorem częstym** (ang. *frequent itemset*).

2.3. Reguły asocjacyjne

Reguła asocjacyjna to implikacja postaci $X \rightarrow Y$, gdzie $X \subset L$, $Y \subset L$ i $X \cap Y = \emptyset$. Zbiór X nazywamy głową reguły a zbiór Y ciałem reguły. Z każdą regułą asocjacyjną związane są dwie miary wyznaczające statystyczne znaczenie i siłę reguły. **Wsparciem** (ang. *support*) reguły $X \rightarrow Y$ w bazie danych D nazywamy stosunek liczby transakcji wspierających regułę do liczby wszystkich transakcji. Innymi słowy reguła $X \rightarrow Y$ ma w bazie danych D wsparcie *s*, jeśli *s*% transakcji w bazie danych wspiera $X \cup Y$.

$$support(X \rightarrow Y, D) = \frac{\left|\{T \in D : T \text{ wspiera } X \cup Y\}\right|}{|D|}$$

Ufnością (ang. *confidence*) reguły $X \rightarrow Y$ w bazie danych D nazywamy stosunek liczby transakcji wspierających regułę do liczby transakcji wspierających głowę reguły. Innymi słowy reguła $X \rightarrow Y$ ma w bazie danych D ufność *c*, jeśli *c*% transakcji wspierających X wspiera również Y.

$$confidence(X \rightarrow Y, D) = \frac{\left|\{T \in D : T \text{ wspiera } X \cup Y\}\right|}{\left|\{T \in D : T \text{ wspiera } X\}\right|}$$

Problem odkrywania reguł asocjacyjnych polega na znalezieniu w danej bazie danych D wszystkich reguł asocjacyjnych, których wsparcie i ufność są wyższe od zdefiniowanych przez użytkownika wartości minimalnego wsparcia i minimalnej ufności (*minsup* i *minconf*).

2.4. Wzorce sekwencyjne

Niech $L=\{l_1,l_2,...,l_n\}$ będzie zbiorem literałów zwanych elementami. Sekwencją nazywamy uporządkowaną listę zbiorów elementów. Sekwencję oznaczamy przez $<X_1,X_2,...,X_n>$, gdzie X_i jest zbiorem elementów, $X_i \subseteq L$. Zbiory X_i nazywamy wyrazami sekwencji. Rozmiarem sekwencji nazywamy liczbę występujących w niej elementów. Długością sekwencji nazywamy liczbę występujących w niej wyrazów. Z każdym wyrazem sekwencji związany jest znacznik czasowy. Pomijając ograniczenia czasowe mówimy, że sekwencja $<X_1,X_2,...,X_n>$ zawiera się w sekwencji $<Y_1,Y_2,...,Y_m>$, jeśli istnieją liczby całkowite $i_1<i_2<...<i_n$ takie, że $X_1<Y_{i1},X_2<Y_{i2},...,X_n<Y_{in}$. Sekwencję $<Y_{i1},Y_{i2},...,Y_{in}>$ nazywamy wystąpieniem sekwencji X w sekwencji Y. Odkrywając wzorce sekwencyjne posługujemy się następującymi ograniczeniami czasowymi: minimalnym i maksymalnym przedziałem pomiędzy kolejnymi elementami wystąpienia sekwencji (odpowiednio *min-gap* i *max-gap*), oraz rozmiarem okna czasowego, które pozwala na łączenie tych samych pozycji sekwencji w jedną pozycję (o ile znaczniki czasowe tych pozycji mieszczą się w określonym oknie).
Wsparciem sekwencji $<X_1,X_2,...,X_n>$ w bazie danych D nazywamy stosunek liczby sekwencji zawierających X do liczby wszystkich sekwencji.
Problem odkrywania wzorców sekwencyjnych polega na znalezieniu w danej bazie danych D wszystkich sekwencji, których wsparcie jest wyższe od zdefiniowanej

przez użytkownika wartości minimalnego wsparcia (*minsup*). Sekwencję, której wsparcie jest wyższe niż *minsup*, nazywamy **wzorcem sekwencyjnym**.

3. Aktualny stan badań

Prace nad perspektywami materializowanymi rozpoczęły się w latach 80-tych. Początkowo były one narzędziem do przyspieszenia wykonywania zapytań i udostępnienia starszych kopii danych. Opracowano wiele algorytmów pielęgnacji perspektyw materializowanych. Dalsze badania dotyczyły między innymi tworzenia modeli szacowania kosztów pielęgnacji perspektyw materializowanych oraz określania wpływu obecności perspektyw na efektywność przetwarzania zapytań. Kolejne prace dotyczyły zastosowania perspektyw do narzucania ograniczeń integralnościowych. W pracy [8] można znaleźć podsumowanie i klasyfikację różnych technik pielęgnacji perspektyw. Obszerne przedstawienie tematu znajduje się w [9].

Problem odkrywania reguł asocjacyjnych po raz pierwszy sformułowano w [1]. W [2] wprowadzono pojęcie zbioru częstego i zaproponowano algorytm Apriori, który stał się podstawą bardzo wielu różnych metod odkrywania wzorców. Działanie algorytmu opiera się na następującej obserwacji: zbiór elementów może być częsty wtedy i tylko wtedy, gdy wszystkie jego podzbiory są częste. Apriori generuje zbiory potencjalnie częste (zwane zbiorami kandydującymi) tylko na podstawie dotychczas znalezionych zbiorów częstych. W pierwszym kroku znajdowane są wszystkie zbiory częste o rozmiarze 1, następnie wszystkie zbiory częste o rozmiarze 2. We wszystkich kolejnych krokach tworzone są zbiory n-elementowe na podstawie (n-1)-elementowych zbiorów częstych. Wsparcie zbiorów kandydujących o danym rozmiarze określa się podczas pełnego odczytu bazy danych. Podstawową wadą algorytmu Apriori jest właśnie fakt, że potrzebuje on (k+1) pełnych odczytów bazy danych aby znaleźć wszystkie k-elementowe zbiory częste.

W [6] zaproponowano algorytm FUP, który odkrywał zbiory częste w oparciu o wcześniej znalezione wyniki. Algorytm ten skracał wydatnie czas działania algorytmu Apriori poprzez przetwarzanie tylko zmodyfikowanej części bazy danych. Kolejną propozycją był algorytm zaprezentowany w [17], który minimalizował czas odkrywania zbiorów częstych w zarówno zwiększonej, jak i zmniejszonej bazie danych. Algorytm ten odkrywał zbiory częste korzystając z pojęcia negatywnej granicy, wprowadzonego w [18].

Idea odkrywania wzorców sekwencyjnych została po raz pierwszy przedstawiona w [3] i [4]. Zaproponowano wówczas algorytm GSP odkrywający szeroką klasę wzorców sekwencyjnych (tzw. uogólnione wzorce sekwencyjne) i wykorzystujący ograniczenia czasowe. W [16] zaproponowano materializację wzorców o obniżonych kryteriach wsparcia i wykorzystanie kolekcji zmaterializowanych wzorców do znalezienia odpowiedzi na dane zapytanie eksploracyjne. Większość prac dotyczących wzorców sekwencyjnych koncentrowała się jednak na ulepszaniu algorytmu odkrywania wzorców [10,11].

W pracy [15] po raz pierwszy rozważano koncepcję interaktywnego i iteracyjnego odkrywania zbiorów częstych. Autorzy zaproponowali stworzenie podręcznej pamięci wiedzy (ang. *knowledge cache*), która przechowywałaby ostatnio odkryte zbiory częste wraz z ich wsparciem. Taka pamięć podręczna może być współdzielona przez wielu użytkowników i wiele aplikacji, dzięki czemu użytkownicy mogą nawzajem wykorzystywać wyniki swoich zapytań eksploracyjnych. Poza opracowaniem samej koncepcji autorzy zaproponowali kilka różnych schematów zarządzania zawartością pamięci podręcznej.
Idea wcześniejszego obliczania wsparcia zbiorów częstych w partycjach bazy danych pojawiła się po raz pierwszy w [19]. Zaproponowana metoda wykorzystywała fakt, że dany zbiór może być częsty wtedy i tylko wtedy, gdy jest częsty w którejkolwiek z partycji. Algorytm dokonywał wcześniejszego odkrywania zbiorów częstych w małych, mieszczących się w całości w dostępnej pamięci operacyjnej partycjach bazy danych i wykorzystywał te zbiory do znajdowania zbiorów, które były częste w całej bazie danych.
W [12] przedstawiono ideę Systemu Zarządzania Wiedzą i Danymi (ang. *Knowledge Data Management System*), który powinien być następcą współczesnych systemów zarządzania bazami danych. Autorzy po raz pierwszy zdefiniowali pojęcie zapytań eksploracyjnych oraz podkreślili potrzebę ścisłej integracji systemów odkrywania wiedzy z istniejącą infrastrukturą informatyczną, przede wszystkim z bazami danych i magazynami danych.

3.1. Streszczenie artykułu

Artykuł zorganizowany jest następująco. W rozdziale 4 przedstawiono koncepcję zapytań eksploracyjnych i zaprezentowano przykłady takich zapytań. Rozważono związki zachodzące między wynikami różnych zapytań i zdefiniowano pojęcie perspektywy eksploracyjnej. Rozdział 5 dotyczy optymalizacji zapytań eksploracyjnych z wykorzystaniem perspektyw materializowanych. W rozdziale 6 przedstawiono nierozwiązane problemy i naszkicowano zarys dalszych kierunków badań.

4. Zapytania eksploracyjne i relacje między zapytaniami

4.1. Zapytania eksploracyjne

W [14] przedstawiono deklaratywny język eksploracyjny MineSQL. Służy on do wyrażania problemów odkrywania wiedzy za pomocą **zapytań eksploracyjnych** (ang. *data mining queries*). Język ten oddziela aplikację użytkownika od używanego algorytmu odkrywania wiedzy. Składnia MineSQL przypomina składnię języka SQL i pozwala na ścisłą integrację zapytań eksploracyjnych z tradycyjnymi zapytaniami do bazy danych. Język MineSQL pozwala aktualnie na wyrażanie poleceń służących do odkrywania zarówno zbiorów częstych, jak i reguł asocjacyjnych i wzorców sekwencyjnych. MineSQL definiuje zbiór dodatkowych

typów danych (SET, ITEMSET, RULE) oraz zbiór operatorów i funkcji operujących na tych typach danych (np. CONTAINS, BODY(x), HEAD(x)).
Poniżej przedstawiono przykładowe zapytanie eksploracyjne, odkrywające w podanym zbiorze danych zbiory częste o wsparciu powyżej 20% i zawierające element '*mleko*'.

```
MINE ITEMSET, SUPPORT(ITEMSET)
FOR ITEMS FROM (
        SELECT SET(PURCHASED_ITEM) AS ITEMS
        FROM PURCHASES
        WHERE DATE_OF_PURCHASE > '01.07.2001'
        AND DATE_OF_PURCHASE < '31.12.2001'
        GROUP BY TRANSACTION_ID )
WHERE SUPPORT(ITEMSET) > 0.2
AND ITEMSET CONTAINS TO_SET('mleko');
```

Analogicznie można w języku MineSQL wyrazić polecenie odkrywania reguł asocjacyjnych, których wsparcie jest większe niż 10%, ufność większa niż 30% i których głowa zawiera element '*masło*'.

```
MINE RULE r, BODY(r), HEAD(r)
FOR ITEMS FROM (
        SELECT SET(PURCHASED_ITEM) AS ITEMS
        FROM PURCHASES
        GROUP BY TRANSACTION_ID )
WHERE SUPPORT(r) > 0.1
AND CONFIDENCE(r) > 0.3
AND HEAD(r) CONTAINS TO_SET('masło');
```

4.2. Relacje zachodzące między wynikami zapytań eksploracyjnych

W [5] określono trzy rodzaje relacji, jakie zachodzą między dwoma zapytaniami eksploracyjnymi Q_1 i Q_2 odkrywającymi wzorce w tej samej bazie danych. Są to równoważność, zawieranie się oraz dominacja.

- Dwa zapytania eksploracyjne są **równoważne**, jeśli dla każdego zbioru danych zwracają ten sam zbiór odkrytych wzorców i dla każdej pary wzorców wartości współczynników statystycznych (np. wsparcia i ufności) są identyczne.
- Zapytanie eksploracyjne Q_2 **zawiera** zapytanie Q_1 jeżeli dla każdego zbioru danych każdy wzorzec odkryty przez zapytanie Q_1 jest też odkryty przez Q_2 i wartości współczynników statystycznych są identyczne w obu przypadkach.
- Zapytanie eksploracyjne Q_2 **dominuje** zapytanie Q_1 jeżeli dla każdego zbioru danych każdy wzorzec odkryty przez zapytanie Q_1 jest też odkryty

przez Q_2 i wartości współczynników statystycznych wyznaczone przez Q_1 są nie mniejsze niż wartości współczynników wyznaczone przez Q_1.

Równoważność zapytań eksploracyjnych jest więc szczególnym przypadkiem relacji zawierania, zaś relacja zawierania jest szczególnym przypadkiem relacji dominacji.

Relacje te zachodzą między wynikami zapytań i mogą być wykorzystane do zidentyfikowania sytuacji, w których można efektywnie udzielić odpowiedzi na zapytanie eksploracyjne Q_1 wykorzystując wynik innego zapytania Q_2. Relacje te mają charakter ogólny i można je zastosować do wielu typów wzorców (zbiorów częstych, reguł asocjacyjnych) oraz wielu modeli ograniczeń.

Jeżeli dla danego zapytania eksploracyjnego Q_1 istnieją zmaterializowane wyniki równoważnego mu zapytania Q_2, wówczas żadne przetwarzanie nie jest konieczne (ponieważ zapytania mają ten sam wynik). Jeśli dostępne są wyniki zapytania Q_2 zawierającego oryginalne zapytanie, konieczny jest jeden pełny odczyt zmaterializowanych wyników i odrzucenie wzorców nie spełniających ograniczeń nałożonych na Q_1. Jeśli dostępne są wyniki zapytania Q_2 dominującego oryginalne zapytanie, konieczny jest jeden pełny odczyt bazy danych i określenie statystycznych współczynników wzorców zmaterializowanych w Q_2. Dodatkowo trzeba z Q_2 odfiltrować te wzorce, które nie spełniają ograniczeń nałożonych na Q_1.

4.3. Perspektywy eksploracyjne

Tradycyjne użycie perspektyw ma na celu przede wszystkim ukrycie przed użytkownikiem skomplikowanych konstrukcji zapytań i uproszczenie dostępu do często odczytywanych danych. Dodatkowo perspektywa zapewnia uniezależnienie aplikacji od bieżącej struktury bazy danych. Wszelkie zmiany zachodzące w bazie danych muszą zostać uwzględnione tylko w definicji perspektywy. Każdy odczyt danych z perspektywy powoduje ponowne wykonanie zapytania definiującego tę perspektywę.

Ponieważ eksploracja danych jest czynnością iteracyjną i powtarzalną, zaś zapytania eksploracyjne mogą być skomplikowane, w [14] wprowadzono pojęcie perspektyw eksploracyjnych. Poniżej przedstawiono polecenie tworzące perspektywę eksploracyjną V_ASSOC_RULES.

```
CREATE VIEW V_ASSOC_RULES AS
MINE RULE, BODY(RULE), SUPPORT(RULE)
FOR ITEMS FROM (
  SELECT SET(PURCHASED_ITEM) AS ITEMS
  FROM PURCHASES
  WHERE TRANS_DATE BETWEEN ’01.01.2002’ AND ’31.01.2002’
  GROUP BY TRANSACTION_ID
  HAVING COUNT(*) >= 3)
WHERE SUPPORT(RULE) > 0.2
AND HEAD(RULE) CONTAINS TO_SET(‘chleb’);
```

W powyższej definicji można wyróżnić dwie klasy ograniczeń: ograniczenia bazodanowe (klauzula WHERE w zapytaniu SELECT) oraz ograniczenia eksploracyjne (klauzula WHERE w zapytaniu MINE). Ograniczenia bazodanowe definiują zbiór danych, w którym następuje odkrywanie wzorców. Ograniczenia eksploracyjne definiują warunki, jakie muszą spełnić odkrywane wzorce.
Dzięki zastosowaniu perspektyw eksploracyjnych aplikacje nie muszą być ściśle powiązane z algorytmem odkrywania wzorców. Zmiany parametrów algorytmu lub zmiany zbioru, w którym odbywa się eksploracja, mogą być wprowadzane do definicji perspektywy, separując tym samym aplikację od szczegółów implementacji algorytmu. Podobnie jak w przypadku tradycyjnych perspektyw, każde odwołanie do perspektywy eksploracyjnej powoduje wykonanie odpowiedniego zapytania, czyli wykonanie algorytmu odkrywania wzorców.
Ponieważ algorytmy odkrywania wzorców są bardzo czasochłonne, wykonanie zapytania do perspektywy eksploracyjnej mogłoby trwać zbyt długo z punktu widzenia interaktywnego procesu odkrywania wiedzy. Rozwiązaniem tego problemu jest materializacja wyników uzyskanych we wcześniejszych zapytaniach eksploracyjnych. Pomysł materializowanych perspektyw eksploracyjnych został po raz pierwszy sformułowany w [14]. Materializowana perspektywa eksploracyjna to obiekt w bazie danych, który przechowuje wzorce (zbiory częste, reguły asocjacyjne, wzorce sekwencyjne) odkryte podczas wykonywania zapytania eksploracyjnego. Wzorce przechowywane w takiej perspektywie mają związany z sobą znacznik czasowy, określający moment ich odkrycia (i ważności). Z każdą perspektywą materializowaną może też być związany przedział czasowy, po którym następuje automatyczne odświeżenie zawartości perspektywy. Poniżej przedstawiono polecenie tworzące perspektywę materializowaną MV_ASSOC_RULES.

```
CREATE MATERIALIZED VIEW MV_ASSOC_RULES
REFRESH 7 AS
MINE RULE, SUPPORT(RULE), CONFIDENCE(RULE)
FOR ITEMS FROM (
        SELECT SET(PURCHASED_ITEM) AS ITEMS
        FROM PURCHASES
        WHERE ITEM_GROUP = 'nabiał'
        GROUP BY TRANSACTION_ID )
WHERE SUPPORT(RULE) > 0.3
AND CONFIDENCE(RULE) > 0.5;
```

Odświeżanie perspektywy materializowanej może odbywać się automatycznie lub na żądanie użytkownika. W większości przypadków perspektywy takie można odświeżać za pomocą jednego z efektywnych algorytmów odświeżania przyrostowego [6,7,17], zamiast wykonywać kosztowny algorytm od podstaw. Na korzyść perspektyw materializowanych przemawia dodatkowo fakt, że eksploracja odbywa się najczęściej w środowisku magazynu danych, w którym zmiany w relacjach bazowych nie mają ciągłego charakteru, lecz następują wszystkie w

jednym momencie, podczas ładowania lub odświeżania zawartości magazynu. Tak więc reguły odkryte i przechowywane w perspektywie materializowanej przez długi czas pozostają poprawne, zaś ich weryfikacja powinna się odbyć podczas odświeżania całego magazynu.

5. Optymalizacja zapytań eksploracyjnych za pomocą perspektyw materializowanych

5.1. Zbiory częste i reguły asocjacyjne

W wielu przypadkach zawartość perspektywy materializowanej może posłużyć do odpowiedzi na zapytanie eksploracyjne, które jest podobne do zapytania definiującego perspektywę. Jeśli np. zapytanie definiujące perspektywę Q_v dominuje lub zawiera dane zapytanie eksploracyjne Q, to w celu udzielenia odpowiedzi na zapytanie Q wystarczy odczytać zawartość perspektywy i odfiltrować te wzorce, które nie spełniają warunków sformułowanych w Q. W celu wykorzystania perspektyw materializowanych do optymalizacji zapytań eksploracyjnych trzeba jednak najpierw określić warunki, jakie muszą być spełnione, aby można było udzielić poprawnej odpowiedzi na zapytanie Q przy użyciu zawartości perspektywy. W tym celu należy najpierw zdefiniować relacje, jakie mogą wystąpić między dwoma zapytaniami eksploracyjnymi.

- Zapytanie Q **rozszerza** ograniczenia bazodanowe zapytania Q_v jeżeli:
  - dodaje do ograniczeń bazodanowych zapytania Q_v dodatkowe klauzule WHERE lub HAVING
  - dodaje koniunkcję nowego warunku do warunków bazodanowych w klauzulach WHERE lub HAVING
  - usuwa warunek z alternatywy warunków bazodanowych w klauzulach WHERE lub HAVING
- Zapytanie Q **redukuje** ograniczenia bazodanowe zapytania Q_v jeżeli:
  - usuwa z ograniczeń bazodanowych zapytania Q_v klauzule WHERE lub HAVING
  - usuwa warunek z koniunkcji warunków bazodanowych w klauzulach WHERE lub HAVING
  - dodaje alternatywę nowego warunku do warunków bazodanowych w klauzulach WHERE lub HAVING
- Zapytanie Q **rozszerza** ograniczenia eksploracyjne zapytania Q_v jeżeli:
  - dodaje do ograniczeń eksploracyjnych zapytania Q_v dodatkowe klauzule WHERE lub HAVING
  - dodaje koniunkcję nowego warunku do warunków eksploracyjnych w klauzulach WHERE lub HAVING
  - usuwa warunek z alternatywy warunków eksploracyjnych w klauzulach WHERE lub HAVING
  - zastępuje ograniczenie eksploracyjne występujące w Q_v ograniczeniem bardziej restryktywnym (np. wyższy próg

minimalnego wsparcia)

- Zapytanie Q **redukuje** ograniczenia eksploracyjne zapytania Q_v jeżeli:
  - redukuje ograniczenia eksploracyjne zapytania Q_v o klauzule WHERE lub HAVING
  - usuwa warunek z koniunkcji warunków eksploracyjnych w klauzulach WHERE lub HAVING
  - dodaje alternatywę nowego warunku do warunków eksploracyjnych w klauzulach WHERE lub HAVING
  - zastępuje ograniczenie eksploracyjne występujące w Q_v ograniczeniem mniej restryktywnym (np. niższy próg minimalnego wsparcia)

Rozszerzenie ograniczeń bazodanowych oznacza zawężenie zbioru danych, w którym ma przebiegać eksploracja. Redukcja ograniczeń bazodanowych oznacza rozszerzenie eksplorowanego zbioru. Rozszerzenie ograniczeń eksploracyjnych oznacza zawężenie zbioru wzorców zaś redukcja ograniczeń eksploracyjnych oznacza rozszerzenie wynikowego zbioru wzorców.

W zależności od okoliczności można wykorzystać jeden z czterech sposobów eksploracji. **Pełna eksploracja** (ang. *full mining*) polega na wykonaniu całego algorytmu odkrywania wzorców, bez wykorzystania zawartości perspektywy. Ta sytuacja następuje wówczas, gdy dane zapytanie Q rozszerza ograniczenia bazodanowe zapytania definiującego perspektywę Q_v. **Eksploracja przyrostowa** (ang. *incremental mining*) polega na wykonaniu któregoś z algorytmów przyrostowego odkrywania wzorców (np. [6]) w rozszerzonym zbiorze danych. Tę metodę można zastosować w przypadku redukcji ograniczeń bazowych względem zapytania Q_v. **Eksploracja uzupełniająca** (ang. *complementary mining*) polega na odkrywaniu wzorców na podstawie wcześniej znalezionych wzorców. Eksploracja uzupełniająca znajduje zastosowanie w przypadku, gdy zapytanie Q redukuje ograniczenia eksploracyjne zawarte w perspektywie (wszystkie wzorce zawarte w perspektywie znajdują się również w odpowiedzi na zapytanie Q). Wreszcie **eksploracja weryfikująca** (ang. *verifying mining*) polega na odfiltrowaniu tych wzorców przechowywanych w perspektywie, które nie spełniają rozszerzonych ograniczeń eksploracyjnych.

Poniżej zamieszczono przykład ilustrujący zastosowanie zawartości materializowanej perspektywy eksploracyjnej do udzielenia odpowiedzi na zapytanie eksploracyjne.

Dana jest następująca definicja perspektywy Q_v:

```
MINE ITEMSET, SUPPORT(ITEMSET)
FOR ITEMS FROM (
        SELECT SET(PURCHASED_ITEM) AS ITEMS
        FROM PURCHASES
        GROUP BY TRANSACTION_ID
HAVING COUNT(*) > 5 )
WHERE SUPPORT(ITEMSET) > 0.3;
```

oraz następujące zapytanie Q:

```
MINE ITEMSET
FOR ITEMS FROM (
        SELECT SET(PURCHASED_ITEM) AS ITEMS
        FROM PURCHASES
        GROUP BY TRANSACTION_ID )
WHERE SUPPORT(ITEMSET) > 0.5
AND ITEMSET CONTAINS TO_SET('mleko','masło');
```

Zapytanie Q rozszerza ograniczenia eksploracyjne Q_v (większa wartość współczynnika minimalnego wsparcia oraz dodatkowa klauzula z ograniczeniami eksploracyjnymi) i jednocześnie redukuje ograniczenia bazodanowe Q_v (usunięta klauzula HAVING z ograniczeń bazodanowych Q_v, zatem wykonywana jest eksploracja weryfikująca (odrzucenie zbiorów częstych o wsparciu niższym niż 0.5 i nie zawierających zbioru {'*mleko*','*masło*'}), a następnie eksploracja przyrostowa, wyszukująca zbiory częste w transakcjach składających się z mniej niż 5 elementów.

5.2. Wzorce sekwencyjne

Podobnie jak w przypadku odkrywania zbiorów częstych lub reguł asocjacyjnych, materializacja wyników wcześniejszych zapytań może wydatnie przyspieszyć odkrywanie wzorców sekwencyjnych. W celu wykorzystania zawartości perspektywy materializowanej do odpowiedzi na zapytanie o wzorce sekwencyjne Q należy najpierw określić rodzaj relacji pomiędzy oboma zapytaniami (zapytaniem eksploracyjnym i zapytaniem definiującym perspektywę). Relacja ta jest uzależniona od związków między poszczególnymi klasami ograniczeń zawartymi w obu zapytaniach.
W podstawowym ujęciu problem odkrywania wzorców sekwencyjnych jest zdefiniowany za pomocą trzech klas ograniczeń:

- ograniczenia bazodanowe: wykorzystywane do zawężenia eksplorowanej bazy danych do interesującego podzbioru
- ograniczenia eksploracyjne: parametry algorytmu odkrywania wzorców sekwencyjnych, aktualnie stosuje się tylko współczynniki minimalnego wsparcia poszukiwanych wzorców (ang. *minsup*)
- ograniczenia czasowe: wykorzystywane do określania rozmiaru okna przetwarzania sekwencji, aktualnie stosuje się współczynniki minimalnej i maksymalnej odległości między elementami oraz szerokości okna (ang. *min-gap*, *max-gap*, *window-width*)

Dla dwóch zapytań eksploracyjnych Q_1 i Q_2 można określić następujące wzajemne relacje w odniesieniu do wyżej wymienionych ograniczeń:

- zapytanie Q_2 **rozszerza** ograniczenia eksploracyjne zapytania Q_1 jeżeli ograniczenia eksploracyjne Q_1 mogą być uzyskane poprzez dodanie do

ograniczeń eksploracyjnych obecnych w Q_2 nowych predykatów elementarnych, lub przez zastąpienie predykatów Q_2 przez silniejsze (bardziej restrykcyjne) predykaty.

- zapytanie Q_2 **rozszerza** ograniczenia czasowe zapytania Q_1 jeżeli zaostrza któryś z współczynników *min-gap*, *max-gap* lub *window-width*, jednocześnie nie łagodząc innych współczynników.

Powyższe uwagi dotyczą składni zapytań eksploracyjnych. Wpływ różnic składniowych na występowanie jednej z ogólnych relacji między zapytaniami (równoważność, dominacja i zawieranie) został zbadany w [20]. Wykazano tam, że w przypadku zapytań eksploracyjnych dotyczących wzorców sekwencyjnych zachodzą następujące zależności:

- Niech zapytania Q_1 i Q_2 operują na tym samym zbiorze danych (czyli posiadają te same ograniczenia bazodanowe) i posiadają te same ograniczenia czasowe. Jeżeli Q_2 rozszerza ograniczenia eksploracyjne Q_1, wówczas Q_1 zawiera Q_2.
- Niech zapytania Q_1 i Q_2 operują na tym samym zbiorze danych i posiadają te same ograniczenia eksploracyjne. Jeżeli Q_2 rozszerza ograniczenia czasowe Q_1, wówczas Q_1 dominuje Q_2.
- Niech zapytania Q_1 i Q_2 operują na tym samym zbiorze danych. Jeśli Q_2 rozszerza ograniczenia eksploracyjne i czasowe Q_1, wówczas Q_1 dominuje Q_2.

Te zależności stanowią podstawę algorytmów wykorzystujących zmaterializowane wyniki wcześniejszych eksploracji do udzielenia odpowiedzi na dane zapytanie. Poniżej przedstawiono krótki opis możliwości wykorzystania poszczególnych technik eksploracji z wykorzystaniem perspektyw materializowanych. Pełny opis tych algorytmów i analiza kosztów ich wykonania znajduje się w [20]. We wszystkich przykładach Q oznacza zapytanie eksploracyjne odkrywające wzorce sekwencyjne, zaś Q_v oznacza zapytanie stanowiące definicję perspektywy materializowanej MV.

Jeżeli Q i Q_v operują na tym samym zbiorze danych (ograniczenia bazodanowe są takie same) i posiadają te same ograniczenia czasowe i eksploracyjne, to między zapytaniami zachodzi równoważność. Wyniki obu zapytań są identyczne i żadne dodatkowe przetwarzanie nie jest konieczne, cały wynik znajduje się w perspektywie materializowanej MV.

Jeżeli Q i Q_v operują na tym samym zbiorze danych i mają te same ograniczenia czasowe, zaś Q rozszerza ograniczenia eksploracyjne Q_v, wówczas można udzielić odpowiedzi przez odrzucenie z perspektywy tych wzorców, które nie spełniają ograniczeń Q (Q_v zawiera Q). Algorytm dokonuje pojedynczego odczytu całej zawartości perspektywy materializowanej MV i dla każdego wzorca sprawdza, czy wzorzec ten spełnia dodatkowe ograniczenia eksploracyjne nałożone przez Q.

Jeżeli Q i Q_v operują na tym samym zbiorze danych i mają te same ograniczenia eksploracyjne, zaś Q rozszerza ograniczenia czasowe Q_v, wówczas można udzielić odpowiedzi poprzez sprawdzenie wsparcia wzorców przechowywanych w perspektywie z użyciem ograniczeń czasowych Q (zgodnie z definicją Q_v dominuje Q). Oczywiście w odpowiedzi znajdą się tylko te wzorce z perspektywy MV, które

posiadają odpowiednio wysoką wartość współczynnika wsparcia.
Jeżeli Q i Q_v operują na tym samym zbiorze danych i Q rozszerza zarówno ograniczenia eksploracyjne jak i czasowe Q_v (czyli Q_v dominuje Q), wówczas można udzielić odpowiedzi poprzez sprawdzenie wsparcia wzorców przechowywanych w perspektywie MV z użyciem ograniczeń czasowych Q i uwzględnieniem ograniczeń eksploracyjnych Q. Algorytm w pierwszym kroku odczytuje zawartość perspektywy MV i usuwa z niej wszystkie wzorce nie spełniające ograniczeń eksploracyjnych Q. Następnie podczas pełnego odczytu zbioru danych źródłowych określana jest wartość wsparcia pozostałych wzorców przy użyciu ograniczeń czasowych Q. W ostatnim kroku algorytm usuwa raz jeszcze te wzorce, których wartość wsparcia jest niższa niż ograniczenie eksploracyjne Q.
Powyższe metody umożliwiają udzielenie odpowiedzi na zapytanie eksploracyjne dotyczące odkrywania wzorców sekwencyjnych bez konieczności wykonywania kosztownego algorytmu. Wykorzystują one wzorce odkryte w poprzednich sesjach i zmaterializowane w postaci perspektyw. Jak pokazują wyniki eksperymentów, czas odpowiedzi z wykorzystaniem wyników wcześniejszych jest wielokrotnie krótszy niż czas wykonania pełnego algorytmu odkrywania wzorców.

6. Podsumowanie

W niniejszym artykule przedstawiono aktualny stan wiedzy dotyczący optymalizacji zapytań eksploracyjnych przy użyciu perspektyw materializowanych. Przeanalizowano możliwości wykorzystania rezultatów wcześniejszych eksploracji do odkrywania zarówno zbiorów częstych, jak i reguł asocjacyjnych i wzorców sekwencyjnych. W tej nowej i fascynującej dziedzinie pozostaje wciąż bardzo wiele otwartych kwestii. Większość przedstawionych algorytmów zakłada, że eksploracja odbywa się na tym samym zbiorze danych, z którego została wyprowadzona perspektywa. Dodatkowo zakłada się, że kształt analizowanych transakcji pozostaje stały. Nierozstrzygnięte pozostają pytania o efektywne metody pielęgnacji materializowanych perspektyw eksploracyjnych. Brakuje wreszcie modeli kosztów dla zapytań eksploracyjnych.
Najbliższa praca autorów skupiać się będzie na rozszerzeniu zakresu stosowalności opisanych metod do zapytań, które różnią się od perspektywy schematem eksplorowanej bazy danych oraz na konstruowaniu modelu kosztów wykonania zapytań eksploracyjnych. Taki model jest konieczny do osiągnięcia ambitnego celu zbudowania optymalizatora szerokiej klasy zapytań eksploracyjnych.

7. Literatura

1. Agrawal, R., Imielinski, T., Swami, A., Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD International Conference on Management of Data, Washington, USA, May 1993

2. Agrawal, R., Srikant, R., Fast Algorithms for Mining Association Rules. In Proc. of the 20th International Conference on Very Large Data Bases (VLDB'94), Santiago, Chile, 1994.
3. Agrawal, R., Srikant, R., Mining Sequential Patterns. In Proc. of the 11th International Conference on Data Engineering (IDCDE'95), Taipei, Taiwan, March 1995
4. Agrawal, R., Srikant, R., Mining Sequential Patterns: Generalizations and Performance Improvements. In Proc. of the 5th International Conference on Extending Database Technology (EDBT'96), Avignon, France, September 1996
5. Baralis, E., Psaila, G., Incremental refinement of mining queries. In Proc. of the 1st International Conference on Data Warehousing and Knowledge Discovery (DaWaK'99), Florence, Italy, September 1999
6. Cheung, D.W., Han, J., Ng, V., Wong, C.Y., Maintenance of discovered association rules in large databases: An incremental updating technique. In Proc. of the 12th International Conference on Data Engineering (ICDE'96), New Orleans, USA, February 1996
7. Cheung, D. W., Lee, S. D. and Kao, B., A General Incremental Technique for Maintaining Discovered Association Rules In Proc. of the 5th International Conference on Database Systems for Advanced Applications (DASFAA'97), Melbourne, Australia, April 1997
8. Gupta, A., Mumick, I.S., Maintenance of Materialized Views: Problems, Techniques, and Applications. IEEE Data Engineering Bulletin, Special Issue on Materilaized Views and Data Warehousing, 18(2), June 1995
9. Gupta, A., Mumick, I.S., Materialized Views: Techniques, Implementations, and Applications, The MIT Press, 1999
10. Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., Hsu, M.-C., FreeSpan: frequent pattern-projected sequential pattern mining. In Proc. of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'2000), Boston, USA, August 2000
11. Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.-C., Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In Proc. of the 17th International Conference on Data Engineering (ICDE'01), Heidelberg, Germany, April 2001
12. Imielinski, T., Mannila, H., A Database Perspective on Knowledge Discovery. Communications of the ACM, Vol.39, No.11, 1996
13. Morzy, T., Wojciechowski, M., Zakrzewicz, M., Data Mining Query Optimization Using Materialized Views
14. Morzy, T., Zakrzewicz, M., SQL-like Language for Database Mining. In Proc. of the 1st East European Symposium on Advances in Databases and Information Systems (ADBIS'97), St-Petersburg, Russia, September 1997
15. Nag, B., Deshpande, P., DeWitt, D.J., Using a Knowledge Cache for Interactive Discovery of Association Rules. In Proc. of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99), San Diego, USA, August 1999

16. Parthasarathy, S., Zaki, M.J., Ogihara, M., Dwarkadas, S., Incremental and interactive sequence mining. In Proc. of the ACM International Conference on Information and Knowedge Management (CIKM'99), November 1999
17. Thomas, S., Bodagala, S., Alsabti, K., Ranka, S., An Ecient Algorithm for the Incremental Updation of Association Rules in Large Databases. In Proc. of the 3rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'97), Newport Beach, USA, August 1997
18. Toivonen, H., Sampling large databases for association rules. In Proc. of the 22th International Conference on Very Large Data Bases (VLDB'96), Bombay, India, September 1996
19. Wojciechowski, M., Zakrzewicz, M., Itemset Materializing for Fast Mining of Association Rules. In Proc. of the 2nd East European Conference on Advances in Databases and Information Systems (ADBIS'98), Poznań, Poland, September 1998
20. Wojciechowski, M., Interactive Constraint-Based Sequential Pattern Mining. In Proc. of the 5th East European Conference on Advances in Databases and Information Systems (ADBIS'01), Vilnius, Lithuania, September 2001

Mgr inż. Mikołaj MORZY
Instytut Informatyki Politechniki Poznańskiej
ul. Piotrowo 3A tel.: (0-61) 665-21-27
email: Mikolaj.Morzy@cs.put.poznan.pl

Dr inż. Marek WOJCIECHOWSKI
Instytut Informatyki Politechniki Poznańskiej
ul. Piotrowo 3A tel.: (0-61) 665-23-78
email: Marek.Wojciechowski@cs.put.poznan.pl