

Politechnika Poznańska  
Wydział Informatyki i Zarządzania  
Instytut Informatyki

Praca dyplomowa magisterska

**RÓWNOLEGŁE ODKRYWANIE REGUŁ ASOCJACYJNYH  
ZAIMPLEMENTOWANE NA PROCESORY GRAFICZNE**

inż. Tomasz Kujawa

Promotor  
dr inż. Witold Andrzejwski

Poznań, 2011

Tutaj przychodzi karta pracy dyplomowej;  
oryginał wstawiamy do wersji dla archiwum PP, w pozostałych kopiach wstawiamy ksero.

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
1.1	Cel i zakres pracy . . . . .	2
<b>2</b>	<b>Podstawy teoretyczne</b>	<b>3</b>
2.1	Definicje . . . . .	3
2.1.1	Model teoretyczny . . . . .	3
2.2	Aktualna wiedza . . . . .	4
2.2.1	Algorytm Apriori . . . . .	4
	Generowanie zbiorów częstych . . . . .	4
2.2.2	Generowanie reguł asocjacyjnych . . . . .	6
2.2.3	Algorytm FP-growth . . . . .	6
	<b>Literatura</b>	<b>7</b>

# Rozdział 1

## Wstęp

Proces informatyzacji przedsiębiorstw, rozpoczęty kilka dekad temu, wprowadził światową gospodarkę na nowe, dotąd nieznane tory rozwoju. Skrócenie procesu produkcyjnego, wprowadzenie kontroli komputerowych, czy też skomputeryzowanych maszyn skróciło i ułatwiło produkcję, a także zarządzanie procesami w firmach i przedsiębiorstwach. Przed ludźmi stanęły możliwości, ale także wyzwania, z którymi nigdy wcześniej nikt nie musiał sobie radzić. Zmiany, jakie nastąpiły przez ostatnie trzy dekady są nieodwracalne i zmuszają informatyków do tworzenia nowych aplikacji, które będą w stanie sprostać wymaganiom im stawianym.

Informatyzacja firm, instytucji oraz innych jednostek organizacyjnych powinna realizować dwa podstawowe cele. Z jednej strony powinna ona usprawniać pracę pojedynczego pracownika poprzez automatyzację realizowanych przez niego rutynowych zadań. Dzięki wykorzystaniu możliwości komputerów działania te powinny być wykonywane szybciej i w sposób bardziej niezawodny. Z drugiej strony celem informatyzacji jest wpływanie na działanie całych firm w wyniku wspomagania decyzji kadry zarządzającej przedsiębiorstwami. Szybka analiza bazująca na pełnej i aktualnej informacji o stanie firmy może ułatwić kadrze zarządzającej podejmowanie trafnych i szybkich decyzje o strategicznym znaczeniu dla rozwoju danego przedsiębiorstwa.

Wprowadzenie komputerów do właściwie każdej przestrzeni ludzkiego życia wpłynęło na wyprodukowanie olbrzymich ilości danych. Reprezentowane są one w sposób umożliwiający ich składowanie i przetwarzanie komputerowe przez aplikacje analityczne. W chwili obecnej ludzkość jest świadkiem eksplozji ilości danych produkowanych przez różnego rodzaju systemy komputerowe. Analiza tych danych przynieść może wymierne korzyści nie tylko w kwestiach finansowych, ale również poznawczych. Dzięki analizie danych zebranych w przeszłości możliwe jest lepsze dopasowanie planów w przyszłości - na tej podstawie planowane mogą być np. akcje marketingowe, czy też promocje w supermarketach spożywczych. Wykorzystanie wiedzy uzyskanej w ten sposób jest niezwykle szerokie i może być użyte w każdym rejonie działalności firmy.

Odkrycie zależności pomiędzy zgromadzonymi danymi bez zastosowania narzędzi informatycznych jest procesem bardzo skomplikowanym i wymagającym do realizacji dużo czasu. Przy obecnej złożoności większości systemów oraz rozmiarom danych produkowanych przez te systemy, koszt czasowy jest na tyle duży, że ręczna analiza tych danych stała się niemożliwa. Dlatego też tworzone są narzędzia umożliwiające odkrywanie prawidłowości w dużych zbiorach danych, by człowiek na tej podstawie mógł podejmować decyzje i wyciągać wnioski.

Dział informatyki, który zajmuje się odkrywaniem ukrytych dla człowieka prawidłowości i reguł w danych nazywa się eksploracją danych (ang. *Data Mining*), który jest jednym z etapów procesu *odkrywania wiedzy z baz danych* (*KDD*, ang. *Knowledge Discovery in Databases*). Proces odkrywania wiedzy w bazach danych obejmują zwykle działania bardziej złożone niż tylko eksploracja

danych. Eksploracja danych to proces odkrywania wiedzy w postaci nowych, użytecznych, poprawnych i zrozumiałych wzorców w bardzo dużych wolumenach danych [FPSSU96]. Możliwości stosowania technik eksploracji danych w praktyce, wymagają efektywnych metod przeszukiwania ogromnych plików lub baz danych. Warto przy tym wspomnieć, że tego typu technologie nie są w chwili obecnej dobrze zintegrowane z systemami zarządzania bazami danych.

Eksploracja danych (w literaturze spotkać można również określenie drążenie danych, ekstrakcja danych, pozyskiwanie wiedzy, czy też wydobywanie danych [EN05]) odbywa się najczęściej w środowisku baz lub hurtowni danych, które stanowią doskonałe źródła danych do analizy - głównie ze względu na łatwość dostępu oraz usystematyzowaną strukturę przechowywanych informacji. Ponieważ liczba odkrytych wzorców w wielu przypadkach może być bardzo duża, odkryte wzorce bardzo często zapisuje się w osobnych relacjach bazy lub hurtowni danych. Pozwala to na ich dalsze przetwarzanie w trybie off-line przez użytkowników końcowych. Pojęcie eksploracji zyskuje coraz większą popularność (również w wymiarze marketingowym) i jest wykorzystywane w wielu dziedzinach ludzkiego życia.

Jednym z najczęściej wykorzystywanych modeli wiedzy w eksploracji danych są reguły asocjacyjne. Reguła asocjacyjna ma postać  $\mathbf{X} \Rightarrow \mathbf{Y}$ , gdzie  $\mathbf{X}$  oraz  $\mathbf{Y}$  są wzajemnie rozłącznymi zbiorami elementów. Przykładem reguły, która mogła zostać odkryta w badzie danych sklepu komputerowego, może być reguła postaci *komputer*  $\wedge$  *myszka*  $\Rightarrow$  *monitor*. Reguła ta prezentuje fakt, że klienci kupujący komputer oraz myszkę z dużym prawdopodobieństwem kupią również monitor. W [Agr94] po raz pierwszy sformułowany został problem odkrywania reguł asocjacyjnych. Podstawą wielu algorytmów odkrywania reguł asocjacyjnych jest algorytm Apriori, zaprezentowany w [AIS93].

W ostatnich latach pojawiły się nowe możliwości wykorzystania współczesnych komputerów. W roku 2007 firma NVIDIA udostępniła programistom uniwersalną architekturę obliczeniową CUDA (ang. *Compute Unified Device Architecture*), który umożliwia wykorzystanie mocy obliczeniowej procesorów graficznych (*GPU*, ang. *Graphics Processing Unit*), bądź innych procesorów wielordzeniowych, do rozwiązywania ogólnych problemów numerycznych w sposób znacząco wydajniejszy niż w przypadku tradycyjnych, sekwencyjnych procesorów [Cor07a]. Choć w grach komputerowych moc obliczeniową jednostek graficznych można wykorzystać do obliczeń fizyki, to CUDA idzie jeszcze dalej, umożliwiając przyspieszenie obliczeń w takich dziedzinach, jak biologia, fizyka, kryptografia, bioinformatyka oraz inne. Dla potrzeb tego segmentu NVidia opracowała specjalny procesor graficzny *Tesla* [Cor07b].

Do tej pory bardzo małe jest zainteresowanie wykorzystaniem tej technologii w procesie odkrywania wiedzy, a w szczególności znajdowania reguł asocjacyjnych. Wyniki przeprowadzonych eksperymentów pozwalają przypuszczać, że algorytm wykorzystujący możliwości procesorów wielordzeniowych będzie wyraźnie szybszy od klasycznych algorytmów eksploracji danych zaimplementowany na tradycyjnych procesorach.

## 1.1 Cel i zakres pracy

Celem pracy jest zaprojektowanie i zaimplementowanie algorytmu odkrywającego reguły asocjacyjne, który będzie wykorzystywał możliwości współczesnych kart graficznych dzięki wykorzystaniu technologii CUDA oraz porównanie zaprojektowanego i zaimplementowanego algorytmu do innych, podstawowych algorytmów odkrywania reguł asocjacyjnych. W ramach pracy dokonane zostanie również zebranie wiedzy dotyczącej algorytmów eksploracji reguł asocjacyjnych.

Rozdział 1 - wstęp.. Tutaj dalszy opis struktury pracy - zrobiony na koniec, gdy wszystko dalej będzie już znane.

## Rozdział 2

# Podstawy teoretyczne

W rozdziale tym przedstawiony zostanie przegląd literatury, który stanowi podstawy wiedzy na temat eksploracji danych, a w szczególności problemu odkrywania reguł asocjacyjnych w dużych zbiorach danych.

### 2.1 Definicje

#### 2.1.1 Model teoretyczny

Niech  $I = \{i_1, i_2, \dots, i_m\}$  będzie zbiorem elementów o liczności  $|I| = m$ . *Transakcją* nazwano dowolny, niepusty podzbiór  $X \subseteq I$  zbioru elementów. Bazą danych  $DB$  nazwano dowolny zbiór par  $(id, X)$ , gdzie  $X$  jest transakcją, a  $id$  jest dowolną wartością unikalną w ramach bazy danych nazywaną *identyfikatorem transakcji*. Bez utraty ogólności założono iż  $id \in \mathbb{N}$ .

*Wsparciem* (ang. *support*)  $sup(X)$  transakcji  $X$ , w bazie danych nazwano częstość wystąpień transakcji w bazie danych. Formalnie przedstawia to wzór 2.1.

$$sup(X) = \frac{|\{id : (id, Y) \in DB \wedge X \subseteq Y\}|}{|DB|} \quad (2.1)$$

**Definicja 1** Niech będą dane dwie transakcje  $X$  i  $Y$  takie, że  $X \cap Y = \emptyset$ . Implikację postaci  $X \Rightarrow Y$  nazwano regułą asocjacyjną.

Łatwo zauważyć, że jeśli poziom ten jest niski, to oznacza to, że nie ma jednoznacznych dowodów na łączne występowanie elementów zbioru  $Z = X \cup Y$ , ponieważ zbiór  $Z$  występuje w niewielkiej liczbie transakcji.

*Poziom ufności* (ang. *confidence*) jest miarą zdefiniowaną dla implikacji reprezentowanej przez regułę asocjacyjną [EN05].

**Definicja 2** *Poziom ufności* ( $conf$ ) reguły asocjacyjnej  $X \Rightarrow Y$  jest równy

$$conf(X \Rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)} \quad (2.2)$$

Po analizie definicji 2 łatwo zauważyć, że poziom ufności może być interpretowany, jako estymacja prawdopodobieństwa  $P(Y|X)$ .

Reguły asocjacyjne zazwyczaj powinny spełniać pewne wymagania zdefiniowane przez użytkownika - minimalne wsparcie oraz minimalny poziom ufności, oznaczane odpowiednio *minsup* oraz *minconf*. Wyznaczają one dla aplikacji progi, jakie powinny spełniać zbiory oraz reguły, aby były brane pod uwagę w trakcie analizy.

**Definicja 3** Zbiorem częstym  $X \subseteq I$  nazywamy taki zbiór, który spełnia zależność  $\text{sup}(X) \geq \text{minsup}$ .

Generowanie reguł asocjacyjnych zazwyczaj sprowadza się do dwóch, osobnych kroków:

1. Minimalne wsparcie jest używane do odnalezienia wszystkich zbiorów częstych w bazie danych  $DB$ .
2. Znalezione zbiory często oraz minimalny poziom ufności są używane do wygenerowania reguł.

Znalezienie wszystkich zbiorów częstych w bazie danych  $DB$  jest zadaniem wymagającym przeszukania wszystkich możliwych kombinacji bez powtórzeń ze zbioru  $I$ . Zbiór możliwych zbiorów elementów ma liczbę równą  $2^n - 1$  (wszystkie zbiory, poza zbiorem pustym, który nie jest w tym wypadku sensownym zbiorem).

Warto zauważyć, że dla każdego zbioru częstego  $Y$ , każdy jego podzbiór  $X$  jest również zbiorem częstym [AIS93]. Korzystając z tej właściwości wsparcia możliwe jest w sposób efektywny znalezienie wszystkich zbiorów częstych w zadanej bazie danych. Dodatkowo, wszystkie reguły zbudowane na podstawie zbioru częstego  $Y$  muszą spełniać warunek minimalnego wsparcia ponieważ spełnia ten warunek zbiór  $Y$ , a suma zbiorów reguły jest zbiorem wyjściowym  $Y$ .

## 2.2 Aktualna wiedza

W rozdziale tym zebrana została oraz opracowana dotychczasowa wiedza (ang. *state-of-the-art*) na temat odkrywania reguł asocjacyjnych. Przedstawione zostaną dwa podstawowe algorytmy wykorzystywane w tym procesie: Apriori oraz FP-growth. Na podstawie tych dwóch algorytmów w chwili obecnej budowane są nowe algorytmy, ich modyfikacje wykorzystujące możliwości współczesnych komputerów.

### 2.2.1 Algorytm Apriori

Pierwszy algorytm odkrywania reguł asocjacyjnych został przedstawiony w roku 1994 w pracy [AS94]. Niżej przedstawione zostaną szczegóły działania algorytmu *Apriori*.

Zagadnienie odkrywania reguł asocjacyjnych można podzielić na dwa etapy [AIS93]:

1. Generowanie zbiorów częstych, których wartość wsparcia jest wyższa od wartości *minsup*
2. Generowanie reguł asocjacyjnych na podstawie zbiorów częstych. Reguła  $X \Rightarrow Y$  jest wynikiem działania algorytmu, jeżeli spełnia ona nierówność  $\text{conf}(X \Rightarrow Y) \geq \text{minconf}$ . Ponieważ zbiór  $Z = X \cup Y$  jest zbiorem częstym, to reguła spełnia również warunek przekraczania minimalnego wsparcia.

Na tym etapie możliwe jest tworzenie reguł, w których w zbiorze *poprzedników* ( $X$  z oznaczeń z definicji 1) jest wiele elementów [AIS93] oraz jeden w *następniku* (zbiór  $Y$  z definicji 1) lub dopuszczana jest możliwość wielu elementów w następniku również [AS94]. W niniejszej pracy analizowany jest taki sposób generowania reguł, w którym oba zbiory mogą być zbiorami więcej niż jednoelementowymi.

### Generowanie zbiorów częstych

W celu wyznaczenia zbiorów częstych algorytm dokonuje analizy bazy danych, by w kolejnych iteracjach generować rodziny coraz to liczniejszych zbiorów, będących zbiorami częstymi dla zadanej wartości *minsup*. Algorytm zaczyna od znalezienia wszystkich zbiorów jednoelementowych,

które są zbiorami częstymi. W każdym kolejnym kroku generowane są zbiory częste na podstawie zbiorów wygenerowanych w kroku poprzednim. Proces ten jest kontynuowany do momentu aż nie zostaną znalezione żadne zbiory częste.

Algorytm generuje zbiory kandydatów jedynie na podstawie zbiorów częstych odkrytych w kroku poprzednim - co ważne generowanie ich odbywa się bez przeglądania bazy danych transakcji. Intuicja podpowiada, że każdy podzbiór zbioru częstego jest zbiorem częstym. Zatem, każdy zbiór częsty zawierający  $k$  elementów może być wygenerowany na podstawie połączenia dwóch zbiorów posiadających  $k - 1$  elementów, a na koniec kasując te zbiory, których jakkolwiek podzbiór nie jest częsty [AS94].

Procedura ALGORYTMAPRIORI przedstawia pseudokod realizujący opisywany w tym rozdziale algorytm Apriori.

#### ALGORYTMAPRIORI

```

1   $L_1 \leftarrow \{1\text{-elementowe zbiory częste}\}$ 
2  for ( $k = 2; L_{k-1} \neq \emptyset; k++$ )
3      do
4           $C_k \leftarrow \text{aprioriGen}(L_{k-1})$ 
5          forall transakcja  $t \in DB$ 
6              do
7                   $C_t \leftarrow \text{subset}(C_k, t)$ 
8                  forall kandydat  $c \in C_t$ 
9                      do  $c.\text{count}++$ 
10              $L_k \leftarrow \{c \in C_k \mid c.\text{count} \geq \text{minsup}\}$ 
11   $\text{Answer} \leftarrow \bigcup_k L_k$ 
```

**Procedura aprioriGen** Procedura *aprioriGen* reprezentuje proces tworzenia zbiorów  $k$ -elementowych kandydatów na podstawie zbiorów wejściowych  $k - 1$ -elementowych. Procedura ta jest podzielona na dwa etapy: łączenia oraz przycinania.

Jak łatwo zauważyć wynikiem działania JOINSTEP są zbiory  $k$ -elementowe, które powstały na podstawie zbiorów wejściowych  $L_{k-1}$ , a ich zawartość różni się tylko jednym elementem.

#### JOINSTEP

```

1  insert into  $C_k$ 
2  select  $p.\text{item}_1, p.\text{item}_2, \dots, p.\text{item}_{k-1}, q.\text{item}_{k-1}$ 
3  from  $L_{k-1} p, L_{k-1} q$ 
4  where  $p.\text{item}_1 = q.\text{item}_1, \dots, p.\text{item}_{k-2} = q.\text{item}_{k-2}, p.\text{item}_{k-1} < q.\text{item}_{k-1}$ 
```

Następnym krokiem jest PRUNESTEP, w którym usuwane są wszystkie elementy  $c \in C_k$ , którego jakkolwiek podzbiór  $(k - 1)$ -elementowy zbioru  $c$  nie należy do  $L_{k-1}$ .

#### PRUNESTEP

```

1  forall zbiór  $c \in C_k$ 
2      do
3          forall  $(k - 1)$ -elementowy podzbiór  $s$  zbioru  $c$ 
4              do
5                  if  $s \notin L_{k-1}$ 
6                      then
7                          delete  $c$  z  $C_k$ 
```



### 2.2.2 Generowanie reguł asocjacyjnych

Po wyznaczeniu zbiorów częstych algorytm przystępuje do drugiego etapu, czyli do budowania reguł asocjacyjnych na podstawie odkrytych zbiorów. Podobnie, jak w [AS94] algorytm będący przedmiotem analizy niniejszej pracy generuje wszystkie możliwe reguły asocjacyjne dla zadanego zbioru. Mniej ogólny sposób generowania reguł został przedstawiony w pracy [AIS93], jednakże podjęto decyzję, że jest to sposób zbyt mało użyteczny w środowisku produkcyjnym.

Aby wygenerować reguły, dla każdego zbioru częstego  $l$  znajdowane są niepuste podzbiory - podzbiór taki oznaczony jest jako  $a$ . Dla takich oznaczeń wygenerowana zostanie reguła  $a \Rightarrow (l - a)$ , jeżeli spełniona jest nierówność  $\frac{\text{support}(l)}{\text{support}(a)} \geq \text{minconr}$ . Warto zauważyć, że dla każdego zbioru częstego generowane są wszystkie możliwe niepuste podzbiory - zapewnia to, że odkryte zostaną wszystkie możliwe reguły.

### 2.2.3 Algorytm FP-growth

# Literatura

- [Agr94] Rakesh Agrawal. Quest: a project on database mining. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1994.
- [AIS93] Rakesh Agrawal, Tomasz Imielinski, Arun Swami. Mining association rules between sets of items in large databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, strony 207–216, 1993.
- [AS94] Rakesh Agrawal, Ramakrishnan Srikant. Fast algorithms for mining association rules. *VLDB*, 1994.
- [Cor07a] NVIDIA Corporation. Cuda - zone. [on-line]  
[http://www.nvidia.pl/object/cuda\\_home\\_new\\_pl.html](http://www.nvidia.pl/object/cuda_home_new_pl.html), 2007.
- [Cor07b] NVIDIA Corporation. Nvidia tesla 20. [on-line]  
[http://www.nvidia.pl/object/cuda\\_home\\_new\\_pl.html](http://www.nvidia.pl/object/cuda_home_new_pl.html), 2007.
- [EN05] Ramez Elmasri, Shamkand B. NAvathe. *Wprowadzenie do systemów baz danych*. Addison-Wesley, Reading, MA, USA, 2005.
- [FPSSU96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, Ramasamy Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.



© 2011 Tomasz Kujawa

Instytut Informatyki, Wydział Informatyki i Zarządzania  
Politechnika Poznańska

Skład przy użyciu systemu L<sup>A</sup>T<sub>E</sub>X.

BibT<sub>E</sub>X:

```
@mastersthesis{ key,  
  author = "Tomasz Kujawa",  
  title = "{Równoległe odkrywanie reguł asocjacyjnych zaimplementowane na procesory graficzne}",  
  school = "Poznan University of Technology",  
  address = "Pozna{\n}, Poland",  
  year = "2011",  
}
```