Projekt MED-P3, algorytm GRM. Raport.

Przedmiot: Metody eksploracji danych w odkrywaniu wiedzy.

Michał Aniserowicz, Jakub Turek

1 Opis zadania

Celem projektu jest zaimplementowanie algorytmu wyznaczania reguł decyzyjnych o minimalnych poprzednikach, które są częstymi generatorami. Algorytm ten jest modyfikacją algorytmu odkrywania częstych generatorów (GRM), opisanego w [1].

2 Założenia

Projekt zrealizowano w oparciu o następujące założenia:

Niefunkcjonalne:

1. Użyty język programowania; platforma: C#; .NET Framework 3.5.
2. Obsługiwane systemy operacyjne: kompatybilne z .NET Framework 3.5[1] (aplikację testowano na systemie Microsoft Windows 7 Ultimate).
3. Rodzaj aplikacji: aplikacja konsolowa.

Funkcjonalne:

1. Aplikacja pobiera dane z pliku (patrz sekcja 3).
2. Aplikacja zwraca wynik działania w dwóch formatach: "przyjaznym dla człowieka" i "excelowym" (patrz sekcja 3).
3. Aplikacja pozwala mierzyć czas wykonania poszczególnych kroków.
4. Zakłada się, że każda transakcja zawarta w danych wejściowych ma przypisaną decyzję.

3 Dane wejściowe i wyjściowe

opis danych wejsciowych i wyjsciowych - opcje (zostana opisane pozniej), minsup bezwzgledne! - dane oddzielone przecinkami (decyzja razem z atrybutami, na dowolnym miejscu) - naglowki w pierwszym wierszu - brak danych - spacja (biale znaki) - dwa formaty wynikow - oprocz tego wynik na konsoli

4 Implementacja

wszystkie istotne kwestie zwiazne z projektowaniem (np. diagramy klas) i implementacja projektowanie: - podzial na moduly (console, dataset processing, GRM) - testy - diagram klas Logic implementacja: - jakis algorytm, moze z diffsetami - rozne sortowania - tidset/diffset - bruteforce/inv list - tracking (poziomy)

[1] Lista systemów kompatybilnych z .NET Framework 3.5 dostępna jest pod adresem: *http://msdn.microsoft.com/en-us/library/vstudio/bb882520%28v=vs.90%29.aspx*, sekcja "Supported Operating Systems".

4.1 Opymalizacje

- wszystkie wartosci otrzymuja identyfikatory liczbowe - skonfliktowane generatory - transaction ids - posortowane (szybkie intersect, except)

roznice z GRM: - dany node jest decyzyjny - nie rozwijamy go (bo generatory dzieci nie beda minimalne) - generatory decyzji trzymane w slowniku (klucz - decyzja), posortowane wg hasha - w ogole nie ma granicy - dla diffsetow transaction ids trzymane w slowniku (klucz - decyzja)

5 Podręcznik użytkownika

podrecznik potencjalnego uzytkownika wytworzonego oprogramowania (zamierzam korzystać z niego podczas sprawdzania Panstwa rozwiazan) - wszystkie opcje programu - przykladowa komenda i wynik na konsoli

6 Analiza poprawności

wszystkie wyniki wytwarzane przez program otrzymane dla malego, przykladowego zbioru danych (w celu weryfikacji poprawnosci działania programu) - przyklad z konsultacji

7 Analiza wydajności

wyniki jakosciowe i ilosciowe na (np. czas dzialania; liczba wzorcow) uzyskane dla wiekszych (wielkich) zbiorow danych(np. z http://archive.ics.uci.edu/ml/ or http://fimi.cs.helsinki.fi/data/ lub uzgodnionych już wcześniej ze mna podczas konsultacji projektowych) - wykresy, wykresy - ze dla duzej liczby atrybutow malo wydajny

8 Wnioski

wnioski z realizacji projektu - ze trzeba by poprawic wykrywanie supergeneratorow - ze ogolnie dziala spoczko (nursey)

Literatura

[1] *Odkrywanie reprezentacji generatorowej wzorców częstych z wykorzystaniem struktur listowych*, Kryszkiewicz M., Pielasa P., Instytut Informatyki, Politechnika Warszawska.