

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:

2.1 Niefunkcjonalne:

1. Użyty język programowania; platforma: C#; .NET Framework 3.5.
2. Obsługiwane systemy operacyjne: kompatybilne z .NET Framework 3.5¹ (aplikację testowano na systemie Microsoft Windows 7 Ultimate).
3. Rodzaj aplikacji: aplikacja konsolowa uruchamiana z wiersza poleceń.

2.2 Funkcjonalne:

1. Aplikacja pobiera dane z pliku (patrz sekcja 3.1).
2. Aplikacja zwraca wynik działania w dwóch formatach: “przyjaznym dla człowieka” i “excelowym” (patrz sekcja 4).
3. Aplikacja pozwala mierzyć czas wykonania poszczególnych kroków algorytmu.

2.3 Dotyczące danych wejściowych:

1. Dane wejściowe zawierają jedynie wartości atrybutów transakcji i ewentualnie nazwy atrybutów transakcji.
2. Wartości atrybutów w pliku wejściowym są oddzielone przecinkami.
3. Każda transakcja ma przypisaną decyzję.
4. Brakujące wartości atrybutów (tzn. wartości nieznane bądź nieustalone) są oznaczone jako wartości puste lub złożone z białych znaków.

3 Dane wejściowe

Aplikacja będąca wynikiem projektu przyjmuje jednocześnie dwa rodzaje danych wejściowych:

- plik zawierający dane transakcji,
- parametry podane przez użytkownika w wierszu poleceń.

¹Lista 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”.

3.1 Plik wejściowy

Plik wejściowy powinien zawierać kolejne wartości atrybutów transakcji według reguł przedstawionych w sekcji 2.3. Przykładowy format pliku wejściowego:

```
a,b,c,d,e, ,g, ,+
a,b,c,d,e,f, , ,+
a,b,c,d,e, , ,h,+
a,b, ,d,e, , , ,+
a, ,c,d,e, , ,h,-
, b,c, ,e, , , , -
```

Opcjonalnie, pierwszy wiersz pliku wejściowego może zawierać nazwy atrybutów transakcji (nagłówki), na przykład:

```
Attr A,Attr B,Attr C,Attr D,Attr E,Attr F,Attr G,Attr H,Decision
a,b,c,d,e, ,g, ,+
, b,c, ,e, , , , -
```

Jeden z atrybutów transakcji musi reprezentować przypisaną jej decyzję. Domyślnie, aplikacja uznaje ostatni atrybut transakcji za “decyzyjny” - użytkownik może jednak samodzielnie wskazać odpowiedni atrybut (patrz sekcja 3.2).

3.2 Parametry wiersza poleceń

Aplikacja przyjmuje następujące parametry:

- **--help** - Powoduje wyświetlenie informacji o dostępnych parametrach i wyjście z programu.
- **-f, --file=VALUE** - Ścieżka do pliku wejściowego. Parametr wymagany.
- **--sup, --minSup=VALUE** - Próg (bezwzględny) wsparcia - wykryte zostaną reguły decyzyjne o poprzednikach cechujących się wsparciem większym **lub równym** progowi wsparcia. Parametr wymagany.
- **-h, --headers** - Flaga oznaczająca, że plik wejściowy zawiera nagłówki atrybutów. Parametr opcjonalny.
- **--dec, --decAttr=VALUE** - Pozycja atrybutu zawierającego wartości decyzji (1 - pierwszy atrybut, 2 - drugi atrybut itd.). Parametr opcjonalny (jeśli nie zostanie podany, ostatni atrybut zostanie uznany za decyzyjny).
- **--sort=VALUE** - strategia sortowania elementów (patrz sekcja 5.1). Parametr opcjonalny. Dopuszczalne wartości:
 - **AscendingSupport** (lub 0; wartość domyślna),
 - **DescendingSupport** (lub 1),
 - **Lexicographical** (lub 2).
- **--store=VALUE** - strategia przechowywania identyfikatorów transakcji (patrz sekcja 5.2). Parametr opcjonalny. Dopuszczalne wartości:
 - **TIDSets** (lub 0; wartość domyślna),
 - **DiffSets** (lub 1).
- **--supgen=VALUE** - Strategia przechowywania generatorów decyzji, a także wykrywania i usuwania ich nadgeneratorów (patrz sekcja 5.3). Parametr opcjonalny. Dopuszczalne wartości:
 - **InvertedLists** (lub 0; wartość domyślna),
 - **BruteForce** (lub 1).

- `--track=VALUE` - Poziom monitorowania wydajności programu (patrz sekcja 5.4). Parametr opcjonalny.
 - `NoTracking` (lub 0),
 - `Task` (lub 1; wartość domyślna).
 - `Steps` (lub 2).
 - `Substeps` (lub 3; Uwaga: może powodować znaczący spadek ogólnej wydajności programu).
- `-o, --output=VALUE` - Ścieżka plików wyjściowych. Parametr opcjonalny. Poprawna wartość jest ścieżką pliku z pominięciem jego rozszerzenia (np. *wyniki/wynik*. Wartość domyślna: *[ścieżka pliku wejściowego]-rules*.

4 Dane wyjściowe

Na wyjście aplikacji składają się trzy rodzaje danych:

- komunikaty diagnostyczne wypisywane na standardowe wyjście,
- plik zawierający znalezione reguły decyzyjne w formacie czytelnym dla człowieka,
- plik zawierający znalezione reguły decyzyjne w formacie przystosowanym do dalszej obróbki.

4.1 Plik czytelny dla człowieka

Plik w formacie czytelnym dla człowieka zawiera listy generatorów dla każdej decyzji. Pojedynczy generator jest przedstawiony jako lista wartości atrybutów wraz z ich nazwami. Przykład:

```
=====
Decision: '+' :
-----
Attr A (a), Attr B (b)
Attr B (b), Attr D (d)
```

Plik ten jest zapisywany pod ścieżką *[plik].txt*, gdzie *[plik]* to wartość parametru `output` (patrz sekcja 3.2).

4.2 Plik przystosowany do dalszej obróbki

Plik w formacie czytelnym dla człowieka zawiera listę generatorów wraz generowanymi przez nie decyzjami. Pojedynczy generator jest przedstawiony jako lista wartości oddzielonych średnikami², z uwzględnieniem atrybutów niewystępujących w generatorze. Przykład:

```
Attr A;Attr B;Attr C;Attr D;Attr E;Attr F;Attr G;Attr H;Decision
a;b;;;;;+
;b;d;;;;;+
```

Plik ten jest zapisywany pod ścieżką *[plik].csv*, gdzie *[plik]* to wartość parametru `output` (patrz sekcja 3.2).

5 Implementacja

wszystkie istotne kwestie związane z projektowaniem (np. diagramy klas) i implementacja projektowanie:
 - podział na moduły (console, dataset processing, GRM) - testy - diagram klas Logic implementacja:
 - jakiś algorytm, może z diffsetami - różne sortowania - tidset/diffset - bruteforce/inv list - tracking (poziomy) biblioteki zewnętrzne: - NDeskOptions - xunit - moq

²Patrz [http://pl.wikipedia.org/wiki/CSV_\(format_pliku\)](http://pl.wikipedia.org/wiki/CSV_(format_pliku)). W omawianej aplikacji zamiast przecinków użyto średników, aby plik wyjściowy mógł być odczytany przez program Microsoft Excel.

5.1 Strategie sortowania elementów

5.2 Strategie przechowywania identyfikatorów transakcji

5.3 Strategie przechowywania generatorów decyzji

5.4 Monitorowanie wydajności programu

5.5 Optymalizacje

- 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)

6 Podręcznik użytkownika

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

7 Analiza poprawności

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

8 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 juz wziesniej ze mna podczas konsultacji projektowych) - uzasadnic supergenerators Inverted Lists - ze sortowanie AscendingSupport - ze storage TIDSets - wykresy, wykresy - ze dla duzej liczby atrybutow malo wydajny

9 Wnioski

wnioski z realizacji projektu - ze trzeba by poprawic wykrywanie supergeneratorow - ze sortowanie ma duzy wplyw - ze ogolnie dziala spoczko (nurse)

Literatura

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