

Dariusz R. AUGUSTYN, Łukasz WARCHAŁ
Politechnika Śląska, Instytut Informatyki

ZASTOSOWANIE SIECI BAYESA W SZACOWANIU SELEKTYWNOŚCI ZAPYTAŃ W OPTYMALIZATORZE ZAPYTAŃ SERWERA BAZY DANYCH ORACLE

Streszczenie. Artykuł opisuje zastosowanie sieci Bayesa do wyznaczania selektywności zapytań. Wartość selektywności zapytania pozwala na wstępne oszacowanie rozmiaru zbioru wynikowego zapytania, co z kolei pozwala na wybór optymalnego sposobu realizacji zapytania. Obliczenie selektywności dla zapytań z warunkiem selekcji opartym na kilku atrybutach tablicy wymaga użycia estymatora wielowymiarowej funkcji gęstości prawdopodobieństwa dla rozkładu wartości atrybutów. Sieć Bayesa może stanowić oszczędną, w sensie zajętości pamięci, reprezentację wielowymiarowego rozkładu wartości atrybutów. Artykuł omawia sposób rozszerzenia funkcjonalności optymalizatora zapytań, tzn. użycia modułów systemu Weka do implementacji metody wyznaczania selektywności (oparej na koncepcji sieci Bayesa) w ramach optymalizatora SZBD Oracle.

Słowa kluczowe: plan wykonania zapytania, selektywność zapytań, estymacja funkcji wielowymiarowego rozkładu prawdopodobieństwa atrybutów, sieci Bayesa, Weka, optymalizator zapytań serwera bazy danych Oracle

APPLYING BAYESIAN NETWORKS FOR QUERY SELECTIVITY ESTIMATION IN QUERY OPTIMIZER OF ORACLE DBMS

Summary. The paper presents applying Bayesian network-based method of a selectivity estimation. The query selectivity allows estimate query result size, which allows to choose the optimal method of query execution. Obtaining the selectivity for a query with a selection condition based on many attributes, requires an estimator of a multidimensional probability density function of attribute values. Bayesian network can be used as a memory-efficient representation of the multidimensional distribution of attribute values. The article shows Bayesian network approach applied for extending the functionality of the query optimizer. Some Weka modules are used for implementing Bayesian network-based selectivity estimation in Oracle DBMS optimizer.

Keywords: query execution plan, query selectivity estimation, estimation of a multidimensional attribute value distribution, Bayesian network, Weka, query optimizer of Oracle DBMS

1. Wprowadzenie

Wykonanie zapytania SQL (lub ogólniej zdania SQL) w tzw. fazie execute jest poprzedzone jego przygotowaniem, tzn. fazą prepare. W fazie przygotowania następuje, oprócz syntaktycznego sprawdzenia zapytania SQL, weryfikacja istnienia obiektów użytych w zadaniu (tablic, perspektyw, kolumn, funkcji), a w szczególności następuje wypracowanie sposobu realizacji (tzw. planu wykonania). Ta ostatnia czynność jest wykonywana przez moduł optymalizatora SZBD.

Optymalizator zapytań wyznacza na ogół kilka sposobów realizacji zapytania SQL, wartościując je przez przypisanie im wartości kosztu. Jest on na ogół wyrażany liczbą jednostek alokacji (tzw. stron, bloków bazy danych), jakie trzeba pobrać z bazy danych przy przyjętym sposobie realizacji zdania. Oczywiście, jest wybierana optymalna metoda realizacji, czyli ta o najniższym koszcie.

1.1. Pojęcie selektywności zapytania

Wczesne oszacowanie liczby stron niezbędnych do realizacji zapytania wymaga znajomości wartości tzw. współczynnika selektywności (ang. *selectivity*). Selektywność dla zapytań jednotablicowych (warunek selekcji oparty na atrybutach jednej tablicy – ang. *single-table query*) to stosunek liczby wierszy spełniających kryteria zapytania do liczby wszystkich wierszy tablicy. Probabilistyczna interpretacja selektywności to prawdopodobieństwo wylosowania wiersza spełniającego kryteria zapytania ze zbioru wszystkich wierszy tablicy. Selektywność jest parametrem o wartościach ze zbioru $[0, 1]$.

Istnieje bardzo wiele metod wyznaczania selektywności. Opracowano teoretycznie i zaimplementowano w komercyjnych SZBD wiele metod wyznaczania selektywności dla zapytań, w których warunek selekcji jest oparty na jednym atrybucie. Wtedy można mówić o wyznaczaniu selektywności na podstawie jednowymiarowego rozkładu prawdopodobieństwa, opisującego wartości tego atrybutu. Większość metod wykorzystuje nieparametryczny estymator funkcji gęstości prawdopodobieństwa – jednowymiarowy histogram (np. histogram typu equi-width, equi-high, MaxDiff, V-Optimal i inne).

1.2. Wyznaczenie selektywności dla wielowymiarowych rozkładów wartości atrybutów

Pewnym problemem jest wyznaczanie selektywności dla zapytań z warunkami selekcji opartymi na wielu atrybutach, zapytań umownie nazwanych zapytaniami wielowymiarowymi. Większość metod w takiej sytuacji działa wg reguły AVI [2] (ang. *attribute independence values*), czyli zakłada niezależność wartości atrybutów. Sprowadza się to do założenia, że selektywność dla złożonego warunku opartego na kilku atrybutach jest równa iloczynowi selektywności warunków prostych (określonych na pojedynczych atrybutach). Jest to bezpośrednio wykorzystanie elementarnej reguły o sposobie wyznaczania prawdopodobieństwa iloczynu zdarzeń niezależnych. Selektywności dla warunków prostych są wyznaczone jako selektywności rozkładów jednowymiarowych. Oczywiście, założenie o niezależności atrybutów jest często niespełnione i takie podejście obarczone jest pewnym błędem.

Rozwiązaniem problemu jest wyznaczenie selektywności dla warunków opartych na wielowymiarowym rozkładzie wartości atrybutów (a nie na podstawie rozkładów brzegowych, jak ma to miejsce przy wykorzystaniu reguły AVI). Wymaga to zastosowania estymatora wielowymiarowej funkcji gęstości rozkładu prawdopodobieństwa, np. wielowymiarowego histogramu. Bezpośrednie użycie histogramu może być zbyt kosztowne pod względem zajętości pamięci (rozmiaru reprezentacji histogramu w metaschemacie bazy danych), szczególnie dla dużej liczby wymiarów (tzw. przekleństwo wymiarowości – ang. *problems with many dimensions* – “*Curse of Dimensionality*”). Zatem, zaproponowano metody wykorzystujące „skompresowaną”, przybliżoną reprezentację wielowymiarowych rozkładów prawdopodobieństwa, np. oparte na wielowymiarowym estymatorze jądrowym [3], obciętych widmie dyskretnej transformaty cosinusowej [4] lub transformaty falkowej [5], czy wykorzystujące algorytmy GenHist [3], STHOLES [6] i wiele innych.

Jednym z obiecujących podejść może być wykorzystanie sieci Bayesa [1] jako przybliżonej i oszczędnej pod względem zajętości pamięci reprezentacji rozkładu wielowymiarowego.

Przedmiotem niniejszego artykułu jest weryfikacja możliwości praktycznego wykorzystania sieci Bayesa przez optymalizator zapytań serwera bazy danych Oracle, w metodach wyznaczania selektywności zapytań ze złożonym warunkiem selekcji.

2. Zastosowanie sieci Bayesa w reprezentacji wielowymiarowego rozkładu wartości atrybutów i wyznaczaniu selektywności zapytań

Sieci Bayesa (ang. Bayesian Networks – BN, belief networks) mają wiele zastosowań, a jednym z nich jest, efektywna pod względem zajętości, reprezentacja wielowymiarowego rozkładu prawdopodobieństwa. Sieć Bayesa to acykliczny graf skierowany. Węzły grafu

(wierzchołki) reprezentują zmienne (tutaj, w omawianym zastosowaniu, atrybuty tablicy wchodzące w skład złożonego warunku selekcji). Skierowane krawędzie opisują związki informacyjne (np. przyczynowe) pomiędzy zmiennymi. Ilościowy opis zależności pomiędzy atrybutami jest wyrażony przez tzw. warunkowe rozkłady prawdopodobieństwa (ang. *Conditional Probability Density – CPD*), przypisane do każdego wierzchołka grafu. Opisują one zależność danego wierzchołka A od poprzedników-rodziców B_i , tzn.:

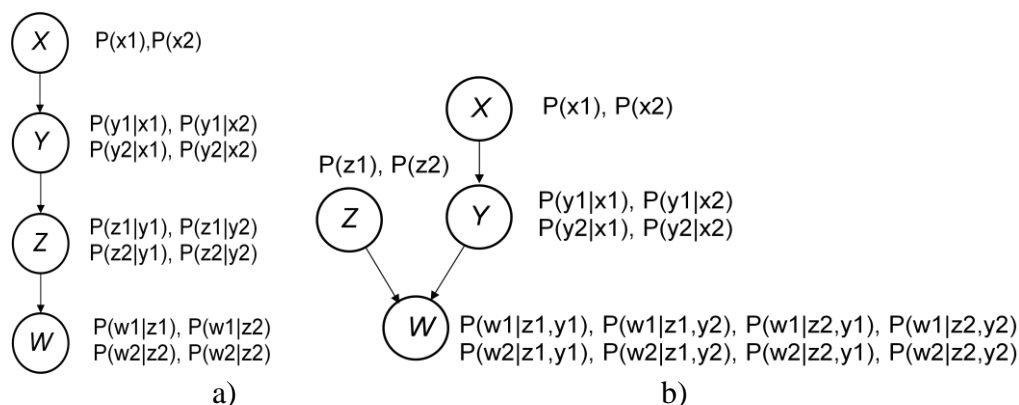
$$P(A|B_1, B_2, \dots, B_n), \quad (1)$$

gdzie B_1, B_2, \dots, B_n są bezpośrednimi poprzednikami atrybutu A w grafie. Każdy wierzchołek jest opisany tablicą CPD, której liczba wymiarów jest równa $n+1$. Zakłada się, że atrybuty niepołączone krawędziami są warunkowo niezależne.

Wyznaczenie łącznego rozkładu prawdopodobieństwa na podstawie sieci Bayesa sprowadza się do użycia tzw. reguły łańcuchowej (opartej na regułach wyznaczania prawdopodobieństwa zdarzeń niezależnych i prawdopodobieństwa zdarzeń warunkowych):

$$P(X_1, X_2, \dots, X_N) = \prod_{i=1}^N P(X_i | Pa(X_i)), \quad (2)$$

gdzie $Pa(X_i)$ oznacza zbiór wszystkich bezpośrednich poprzedników X_i (od ang. parents).



Rys. Przykładowe struktury Bayesa z warunkowymi rozkładami prawdopodobieństw w węzłach: a) lista, b) sieć

Fig. 1. Examples of Bayesian networks with probability tables of conditional distributions in nodes: a) list, b) network

Przykładowo, założmy, że atrybuty X, Y, Z, W przyjmują odpowiednio po dwie wartości $X=\{x_1, x_2\}, \dots, W=\{w_1, w_2\}$. Dwa przykłady reprezentacji łącznego rozkładu zmiennych X, Y, Z, W w postaci sieci Bayesa przedstawiono na rys. 1. Przyjęto uproszczoną notację, gdzie np. $P(x_1)$ oznacza $P(X = x_1)$, a $P(y_1|x_2)$ oznacza $P(Y=y_1|X=x_2)$, itp.

Prawdopodobieństwo $P(x_1, y_2, z_2, w_1)$ na podstawie reguły łańcuchowej (wzór 2) dla przykładu z rys. 1a wynosi $P(x_1) P(y_2|x_1) P(z_2|y_1) P(w_1|z_2)$. Prawdopodobieństwo $P(x_1, y_2, z_2, w_1)$ dla przykładu z rys. 1b wynosi $P(x_1, y_2, z_2, w_1) = P(w_1|z_2, y_2) P(z_2) P(y_2, x_1) = P(x_1) P(y_2|x_1) P(z_2) P(w_1|y_2, z_2)$.

Sieć Bayesa można wykorzystać do wyznaczania prawdopodobieństw przy niepełnej informacji o wszystkich wartościach X, Y, Z, W – np. prawdopodobieństwo dla $P(x1, ?, z2, w1) = P(x1, z2, w1)$ dla przykładu z rys. 1a wynosi $P(x1) P(y1|x1) P(z2|y1) P(w1|z2) + P(x1) P(y2|x1) P(z2|y2) P(w1|z2)$.

2.1. Tworzenie sieci Bayesa jako reprezentacji rozkładu wartości atrybutów tabeli bazy danych

W kolejnym prostym przykładzie przyjmijmy, że istnieje tablica o strukturze i zawartości przedstawionej na rys. 2. Łatwo zauważyć korelację pomiędzy wartościami danych w kolumnach A, B, C , np. wartości B są na ogół większe dla $A = 'N'$, wartości $C = 'B'$ występują tylko dla $A = 'N'$, a wartość $C = 'C'$ występuje tylko dla $A = 'T'$ itp.

```
Create table Tablica
( A Char(1),
  B Int,
  C Char(1) )
```

a)

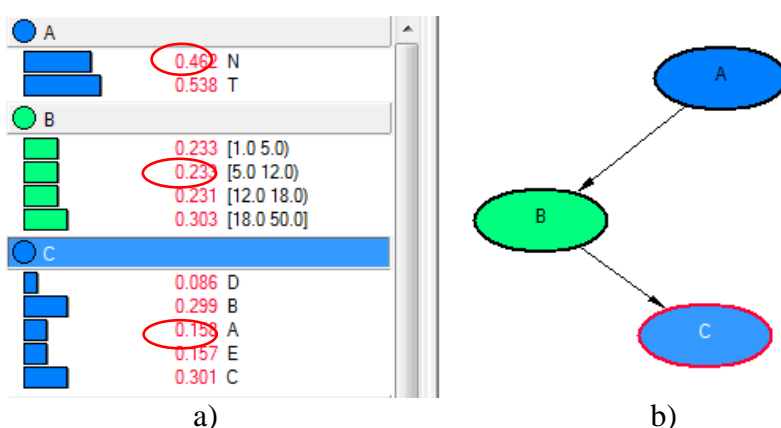
A	B	C
T	1	A
N	12	B
T	3	C
N	16	B
T	7	C
N	18	B
N	23	D
N	14	B
T	2	A
T	5	C
T	50	E
N	36	E
T	8	C

b)

Rys. 2. Przykładowa tablica: a) definicja struktury, b) zawartość

Fig. 2. A sample table: a) structure definition, b) table content

Przed utworzeniem sieci Bayesa dla pewnych zmiennych (np. dla zmiennych z ciągłą dziedziną wartości lub zmiennych dyskretnych o dużej liczbie unikalnych wartości) należy przeprowadzić tzw. proces dyskretyzacji (od ang. *discretisation*), który na ogół sprowadza się do zbudowania histogramu equi-width lub equi-high, opisującego z zadaną rozdzielczością brzegowy rozkład zmiennej. W rozpatrywanym przypadku dokonano dyskretyzacji zmiennej B – wybrano histogram kwantylowy, czyli histogram o (prawie) stałej wysokości z 4 podprzedziałami (rys. 3a).



Rys. Tworzenie sieci Bayesa: a) rozkłady brzegowe wartości atrybutów, b) struktura sieci 3.

Fig. 3. Creation of Bayesian network: a) marginal distributions of attribute values, b) network structure

Tworzenie sieci odbywa się dwuetapowo:

- 1) etap wyznaczania struktury sieci,
- 2) etap obliczania tablic wartości prawdopodobieństw rozkładów warunkowych w węzłach.

Etap 1, dotyczący automatycznego wyznaczania struktury sieci na podstawie danych, zalicza się do problemów z zakresu optymalizacji kombinatorycznej i polega na wyborze struktury o największej przyjętej funkcji oceny (ang. *local score function*)¹. Funkcja oceny na ogół budowana jest na podstawie miary Kullback-Leiblera [7] do wyrażenia „odległości” pomiędzy dwoma rozkładami prawdopodobieństwa². Różnica Kullback-Leiblera (tzw. entropia względna) pomiędzy dwoma rozkładami P i Q jest zdefiniowana następująco:

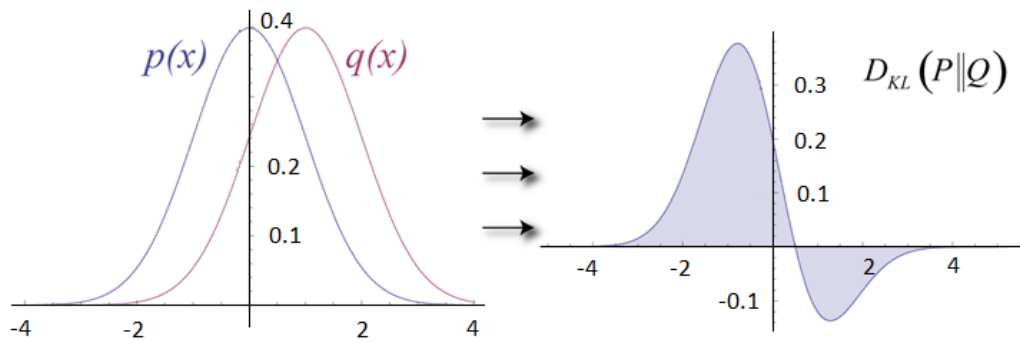
$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \quad \text{dla rozkładów dyskretnych,} \quad (3)$$

$$D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx \quad \text{dla rozkładów ciągłych,} \quad (4)$$

gdzie $p(x)$ i $q(x)$ są funkcjami gęstości prawdopodobieństwa. Rysunek 4 stanowi ilustrację odległości D_{KL} dla pewnych rozkładów ciągłych.

¹ Ogólnie, algorytmy realizujące ten etap charakteryzują się wysoką złożonością (problem NP-trudny). Jednak w omawianym zastosowaniu, tworzenie sieci Bayesa na podstawie danych z tablicy jest wykonywane sporadycznie, w ramach tzw. aktualizacji statystyk bazy danych, czyli czasochłonność tworzenia sieci nie jest elementem dyskredytującym omawiane rozwiązanie. Natomiast krytycznym elementem, pod względem czasu realizacji, jest wyznaczanie selektywności przez optymalizator „na podstawie sieci”, co, jak zostanie pokazane później, jest realizowane przez prosty algorytm przejścia przez sieć.

² D_{KL} jest zwyczajowo nazywana miarą, chociaż nie spełnia warunku symetryczności.



Rys. 4. Odległość Kullback-Leiblera dla przykładowych rozkładów Gaussa (źródło: [7])

Fig. 4. Kullback-Leibler divergence for two sample Gaussian distributions (source: [7])

Odległość D_{KL} może być wykorzystana do określenia rozbieżności rozkładu łącznego $P(X_i, X_j)$ w stosunku do rozkładu zmiennych niezależnych $P(X_i)P(X_j)$:

$$\text{DIST}(X_i, X_j) = \sum_{x_i, x_j} P(x_i, x_j) \log \frac{P(x_i, x_j)}{P(x_i)P(x_j)}. \quad (5)$$

Wartość DIST będzie tym większa, im rozkład będzie bardziej odległy od rozkładu, zakładającego niezależność X_i i X_j . Stąd wynika idea takiego doboru poprzedników w sieci Bayesa, aby rozkłady warunkowe w węzłach były najmniej zbliżone do rozkładu zmiennych niezależnych, czyli najlepiej odzwierciedlały zależności pomiędzy zmiennymi.

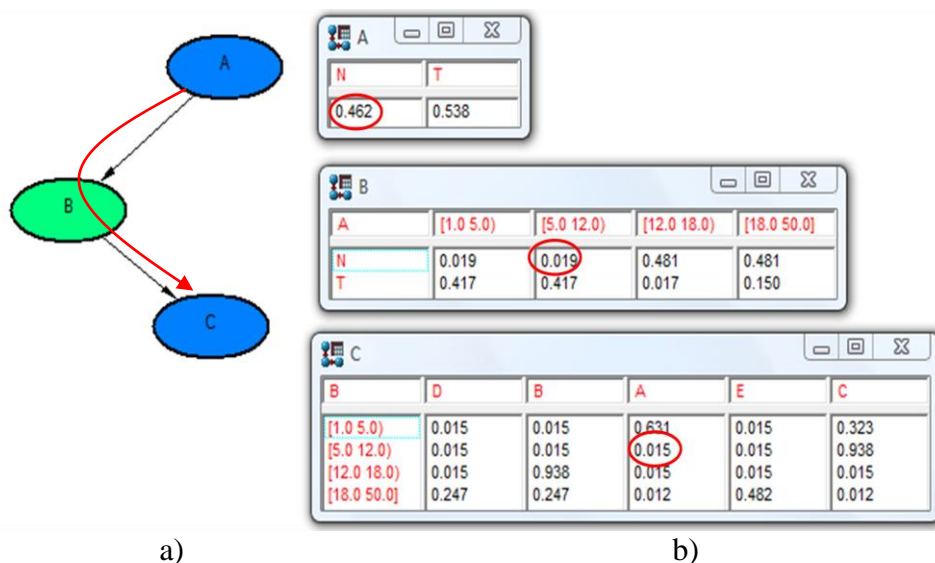
Po uwzględnieniu powyższych rozważań funkcja oceny doboru struktury sieci może być oparta na formule brzegowej funkcji logarytmicznego podobieństwa – ang. *marginal log-likelihood*:

$$\text{DIST}_{\log\text{-likelihood}} = \sum_{i=1}^N \sum_{x_i, \text{Pa}(x_i)} P(x_i, \text{Pa}(x_i)) \log P(x_i | \text{Pa}(x_i)). \quad (6)$$

Dobór struktury (znalezienie optymalnej sieci) polega na znalezieniu takiej struktury sieci, dla której $\text{DIST}_{\log\text{-likelihood}}$ przyjmuje wartość największą.

W wyniku analizy wszystkich możliwych struktur sieci dla trzech atrybutów A, B, C uzyskano optymalną sieć o strukturze pokazanej na rys. 3b.

W etapie drugim tworzenia następuje ilościowy opis sieci bayesowskiej – utworzenie tablic prawdopodobieństwa opisujących warunkowe rozkłady prawdopodobieństwa w węzłach. Na rys. 5b pokazano rezultat utworzenia tablic z opisem rozkładów warunkowych dla danych z tablicy bazy danych (zdefiniowanej na rys. 2). Zakłada się, że węzły wejściowe (niezależne) są opisane jedynie rozkładami brzegowymi (jak np. zmienna A na rys. 5b).



Rys. 5. Utworzona sieć Bayesa: a) struktura sieci i ścieżka przejścia przez sieć podczas wyznaczania selektywności, b) tablice warunkowych rozkładów prawdopodobieństwa w węzłach sieci

Fig. 5. Created Bayesian network: a) network structure and traversal through network during selectivity calculation, b) tables of conditional distributions in network nodes

2.2. Rozmiar reprezentacji łącznego rozkładu wartości atrybutów przy zastosowaniu sieci Bayesa

Celowość zastosowania sieci Bayesa ma się przejawiać w ograniczeniu zajętości pamięci na potrzeby reprezentacji wielowymiarowego rozkładu wartości atrybutów.

W bazach danych są przechowywane rozkłady brzegowe atrybutów. W rozpatrywanym przykładzie przechowywanie rozkładów brzegowych atrybutów A , B , C wymaga zapamiętania $2 + 4 + 5$ liczb (rys. 3a). Tyle danych wystarczy, by realizować niedokładną metodę wyznaczania selektywności opartą na regule AVI (niezależność atrybutów A , B , C).

Aby wykorzystywać sieć Bayesa, należy przechowywać tablice prawdopodobieństw warunkowych w węzłach. W rozpatrywanym przykładzie dotyczy to węzłów B i C , czyli wymaga zapamiętania wartości z dwóch tablic, tzn. $2 \times 4 + 4 \times 5 = 28$ liczb (rys. 5b).

Przechowywanie pełnej informacji o rozkładzie wielowymiarowym atrybutów A , B , C (wielowymiarowy histogram) wymaga przechowania aż $2 \times 4 \times 5 = 40$ liczb.

Reprezentacja rozkładu w postaci sieci Bayesa jest w rozpatrywanym przykładzie o 30% oszczędniejsza pod względem zajętości pamięci od wielowymiarowego histogramu.

Oczywiście, przewaga reprezentacji w postaci sieci Bayesa będzie się jeszcze bardziej uwidaczniać dla większej liczby wymiarów (niż $N = 3$ z rozpatrywanego przykładu).

2.3. Wyznaczanie selektywności zapytań z wykorzystaniem sieci Bayesa

Rozważmy zadanie wyznaczenia selektywności dla zapytania SQL o następującej treści:

```
SELECT * FROM Tablica WHERE A='N' AND 5 <= B <12 AND C ='A'
```

co sprowadzać się będzie do wyznaczenia prawdopodobieństwa wylosowania wiersza spełniającego kryteria selekcji, czyli prawdopodobieństwa zdarzenia $P(a, b, c)$, gdzie:

a: $A = 'N'$, b: $5 \leq B < 12$, c: $C = 'A'$.

Selektywność wyznaczona na podstawie reguły niezależności atrybutów (oznaczona jako *AVISelectivity*) wynosi $P(a, b, c) = P(a) P(b) P(c) = 0.462 * 0.233 * 0.158 = 0.017$. Wyznaczenie selektywności wymaga jedynie przemnożenia prawdopodobieństw odczytanych z rozkładów brzegowych (zaznaczonych elipsami na rys. 3a).

Selektywność wyznaczona na podstawie sieci Bayesa (oznaczona jako *BNSelectivity*) wynosi $P(a, b, c) = P(a) P(b|a) P(c|b) = 0.462 * 0.019 * 0.015 = 0.000132$. Wyznaczenie selektywności wymaga jedynie „przejścia przez sieć” (rys. 5a) i odczytu wartości prawdopodobieństw warunkowych (zaznaczonych elipsami na rys. 5b).

Wyznaczona w ten sposób selektywność – *BNSelectivity* – jest o ok. dwa rzędy dokładniejsza od *AVISelectivity*, ponieważ, jak łatwo zauważyć, dokładna selektywność rozpatrywanego zapytania wynosi 0 (w przykładowej tablicy z rys. 2b nie ma wierszy spełniających warunek $A = 'N'$ i $C = 'A'$).

3. Wykorzystanie systemu Weka do tworzenia sieci Bayesa

Weka (ang. *Waikato Environment for Knowledge Analysis*) [8] jest popularnym systemem wspomagającym rozwiązywanie zagadnień z zakresu uczenia maszynowego (ang. *machine learning*), utworzonym na Uniwersytecie Waikato. System jest napisany w języku Java, co umożliwia jego uruchomienie na wielu różnych systemach operacyjnych.

Weka może stanowić narzędzie m.in. wspomagające rozwiązywanie problemów wykorzystujących sieci Bayesa [9]. Możliwa jest praca z systemem w trybie interaktywnym (jest dostępny moduł Bayes Network GUI), ale system Weka można wykorzystać również jako bibliotekę pozwalającą na rozwój własnego oprogramowania.

W systemie Weka pracę z danymi wejściowymi rozpoczyna się od ich wczytania do pamięci komputera. Źródłem danych może być specjalnie sformatowany plik z rozszerzeniem *.arff (ang. *Attribute-Relation File Format*). Możliwe jest także pobranie informacji bezpośrednio za pośrednictwem serwera bazy danych [11]:

```
InstanceQuery query = new InstanceQuery();  
query.setUsername("dbuser");  
query.setPassword("dbpass");
```

```
query.setQuery("SELECT * FROM SampleTable");
Instances inputData = query.retrieveInstances();
```

Wszystkie wczytane dane są przechowywane w obiektach implementujących bazowy interfejs *weka.core.Instance*. Przed przystąpieniem do budowy sieci Bayesa często jest konieczne przeprowadzenie wstępnej obróbki danych, np. dyskretyzacji (choć zaimplementowany w Wece algorytm jest w stanie sam ją przeprowadzić). W tym celu można użyć filtra z klasy *weka.filters.unsupervised.attribute.Discretize*, co zapewni pełną kontrolę nad doborem przedziałów, ich liczebnością itp.

W pierwszym etapie budowy sieci Bayesa następuje wybór jej struktury. Odbywa się to przez użycie jednej z implementacji algorytmu wyboru postaci sieci na podstawie wartości lokalnej funkcji oceny (ang. *local score metrics*). Algorytmy są zaimplementowane w klasach pakietu *weka.classifiers.bayes.net.search.local*. W omawianym zastosowaniu przyjęto algorytm *GeneticSearch* (choć inne algorytmy też spełniały wymagania). Dodatkowo (niezależnie od rodzaju algorytmu), istnieje możliwość wyboru implementacji samej lokalnej funkcji oceny (ang. *scoreType*). W omawianym zastosowaniu wybrano funkcję oceny opartą na metryce MDL, wykorzystującą metrykę entropii (przybliżoną czytelnikowi w podrozdziale 2.1).

W kolejnym etapie są wyznaczone wartości warunkowych rozkładów prawdopodobieństwa w węzłach sieci. Odbywa się to przez użycie implementacji jednego z algorytmów uczenia tablic prawdopodobieństwa, dostępnych w klasach pakietu *weka.classifiers.bayes.net.estimate*. W omawianym zastosowaniu użyto klasy *SimpleEstimator*.

Funkcjonalność związana z utworzeniem reprezentacji rozkładu atrybutów w postaci sieci Bayesa została bezpośrednio zaimplementowana z użyciem bibliotek systemu Weka:

```
String options =
    "-D -Q weka.classifiers.bayes.net.search.local.GeneticSearch " +
    "-- -L 10 -A 100 -U 10 -R 1 -M -C -S BAYES " +
    "-E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5";

BayesNet bayesNet = new BayesNet();
bayesNet.setOptions(Utils.splitOptions(options));
bayesNet.buildClassifier(instances);
```

Wyznaczenie selektywności zapytań odbywa się przez przemnożenie wartości prawdopodobieństw rozkładów brzegowych pobranych metodą *getProbability()* z klasy *BayesNet*.

4. Rozszerzenie optymalizatora zapytań SZBD Oracle w zakresie metody wyznaczania selektywności definiowanej przez użytkownika

Moduł ODCIStats (**O**racle **D**ata **C**artridge **I**nterface **S**tatistics) [10] umożliwia rozszerzenie funkcjonalności SZBD Oracle w zakresie tworzenia/aktualizacji/usuwania statystyk dotyczących bazy danych oraz wyznaczania selektywności zapytań czy obliczania kosztu realizacji zapytań. Pozwala to na utworzenie własnych metod wyznaczania selektywności, których użycie jest w pełni zintegrowane z SZBD (użycie w optymalizatorze metod selektywności, definiowanych przez użytkownika, nie wymaga odzwierciedlenia tego w treści formułowanych zdań SQL). Tworzenie statystyk definiowanych przez użytkownika (specyficznych dla danej dziedziny) jest „przezroczyste” dla administratora, tzn. jest realizowane przez standardowe komendy, takie jak ANALYZE TABLE COMPUTE|ESTIMATE STATISTICS czy DBMS_STATS.GATHER_TABLE_STATS. Użycie ww. mechanizmu wymaga utworzenia obiektu implementującego wybrane metody interfejsu ODCIStats, tzn. co najmniej metod *ODCIStatsCollect* i *ODCIStatsSelectivity*.

4.1. Przykładowa tablica z dziedzinowym, złożonym typem danych

Na potrzeby ilustracji działań niezbędnych dla uruchomienia własnych statystyk zdefiniowano własny typ złożony:

```
CREATE TYPE SampleComplexType AS OBJECT (a CHAR(1), b NUMBER, c CHAR(1)).
```

oraz tablicę:

```
CREATE TABLE SampleTable (id NUMBER, ca SampleComplexType, name VARCHAR(100)).
```

Atrybut *id* jest kluczem głównym w tablicy *SampleTable*.

Stworzono również funkcję operującą na obiektach utworzonego typu:

```
CREATE FUNCTION SampleChckFnc (arg SomeComplexType,  
    aval CHAR(1), bleft NUMBER, bright NUMBER, cval CHAR(1)) RETURN NUMBER AS BEGIN  
    IF arg.a = aval and arg.b=>bleft and arg.b<bright and arg.c = cval  
        RETURN 1; ELSE RETURN 0; END  
END
```

sprawdzającą wartości obiektu typu *SampleComplexType*.

Dzięki temu zapytanie:

```
SELECT * FROM SampleTable WHERE ca.a='N' AND 5 <= ca.b <12 AND ca.c ='A'
```

można wyrazić następująco:

```
SELECT * FROM SampleTable WHERE SampleChckFnc (ca, 'N', 5, 12, 'A')=1.
```

Celem działań przedstawionych w kolejnych podrozdziałach jest takie rozszerzenie SZBD, aby optymalizator zapytań użył zaproponowanej (specyficznej) metody wyznaczania selektywności (opartej na sieci Bayesa), przy ewaluacji warunku selekcji opartym na funkcji *SampleChckFnc*.

4.2. Tworzenie reprezentacji wielowymiarowego rozkładu w postaci sieci Bayesa

Wielowymiarowy rozkład złożonego atrybutu *ca* w tablicy *SampleTable* będzie reprezentowany przez sieć Bayesa.

Utworzona dla atrybutu *schema_name.table_name.col_name* sieć Bayesa jest serializowana i zapamiętywana w systemowej tablicy *BNStatsTable* w atrybucie *BNet*:

```
CREATE TABLE BNStatsTable(id_stat NUMBER, schema_name VARCHAR(100),
table_name VARCHAR(100), col_name VARCHAR(100), BNet BLOB).
```

Atrybut *id_stat* jest kluczem głównym *BNStatsTable*.

Funkcja PL/SQL *BNStatsCreateFun*:

```
CREATE FUNCTION BNStatsCreateFun (schema_name varchar(100),
table_name varchar(100), col_name varchar(100))
RETURN NUMBER as language java name
'BNUtility.CreateBayesianNetwork (String, String, String) return int'.
```

pośrednio tworzy dla wskazanej kolumny statystykę w postaci sieci Bayesa, zapisuje jej reprezentację do *BNStatsTable* i zwraca unikalny identyfikator tej „statystyki”. Właściwa implementacja ww. funkcjonalności jest zrealizowana w Javie, w statycznej metodzie *CreateBayesianNetwork* klasy *BNUtility*. To w ramach *CreateBayesianNetwork* jest realizowana m.in. funkcjonalność ewentualnej dyskretyzacji zbioru wartości atrybutu (np. wykonywanej automatycznie, gdy liczba unikalnych wartości przekracza zadany próg), etap wyboru struktury sieci Bayesa, etap obliczania wartości prawdopodobieństw warunkowych w węzłach sieci. Funkcjonalności te zostały zrealizowane dzięki bezpośredniemu wykorzystaniu możliwości modułów systemu Weka, wspomnianych w rozdziale 3.

Głównym elementem rozszerzającym system jest *BNStatsObject* – obiekt implementujący interfejs *ODCIStats* w zakresie tworzenia statystyki (oraz wyznaczania selektywności):

```
CREATE TYPE BNStatsObject AS OBJECT (
  STATIC FUNCTION ODCIStatsCollect(col sys.ODCIColInfo,... ) ...
  STATIC FUNCTION ODCIStatsSelectivity(...,sel OUT NUMBER,... ,
  ...).
```

Statyczna funkcja *ODCIStatsCollect* wywołuje *BNStatsCreateFun* dla atrybutu określonego w parametrze *col*, tworząc odpowiednią statystykę (reprezentację sieci Bayesa).

Aby skojarzyć wybraną metodę tworzenia statystyki z konkretnym typem danych, należy wykonać komendę:

```
ASSOCIATE STATISTICS WITH TYPES SampleType USING BNStatsObject.
```

Ostatecznie, można już utworzyć automatycznie statystykę zdefiniowaną przez użytkownika za pomocą standardowej komendy:

```
ANALYZE TABLE SampleTable COMPUTE STATISTICS FOR COLUMNS ca, name.
```

Wyżej wymieniona komenda utworzy statystyki, tzn. reprezentacje rozkładów:

- standardowy histogram equiwidth dla jednowymiarowego rozkładu wartości atrybutu *name*,
- reprezentacja w postaci sieci Bayesa dla wielowymiarowego rozkładu wartości złożonego atrybutu *ca*.

4.3. Definicja własnej metody wyznaczania selektywności

Funkcja PL/SQL *BNSelectivityFun*:

```
CREATE FUNCTION BNSelectivityFun (id_stat NUMBER, a CHAR(1), bleft NUMBER,
    bright NUMBER, c CHAR(1)) RETURN NUMBER as language java name
    'BNUtility.CountSelectivity (int, String, int, int, String) return double'
```

dla zadanego identyfikatora istniejącej statystyki *id_stat* deserializuje strukturę zapisaną w *BNStatsTable* i wyznacza wartość selektywności na podstawie sieci Bayesa, dla warunku określonego w parametrach *a*, *bleft*, *brighth*, *c*. Funkcjonalność ta jest zrealizowana w Javie w ramach statycznej metody *CountSelectivity*.

Główny element rozszerzający system – *BNStatsObject* – implementuje interfejs *ODCIStats* w zakresie wyznaczania selektywności (oraz, jak już przedstawiono, w zakresie tworzenia statystyki):

```
CREATE TYPE BNStatsObject AS OBJECT (
    STATIC FUNCTION ODCIStatsCollect(col sys.ODCIColInfo,... ) ...
    STATIC FUNCTION ODCIStatsSelectivity(...,sel OUT NUMBER,... ,
    arg SomeCompleType, aval CHAR(1), bleft NUMBER, bright NUMBER, cval CHAR(1))...
    STATIC FUNCTION ODCIStatsSelectivity(...,sel OUT NUMBER,... ,) ...
...).
```

Styczna funkcja *ODCIStatsSelectivity* wywołuje *BNSelectivityFun* dla wyznaczenia wartości selektywności, zwracanej w parametrze *sel*.

Metoda *ODCIStatsSelectivity* może być przeciążana wielokrotnie. W związku z tym, że istnieje wersja metody *ODCIStatsSelectivity* o sygnaturze zgodnej (w zakresie końcowej listy argumentów) z sygnaturą funkcji *SampleChckFnc*, jest możliwe wymuszenie uruchomienia właściwej wersji metody *ODCISelectivity* przez optymalizator przy oszacowywaniu selektywności warunku opartego na *SampleChckFnc*. Aby wymusić takie działanie optymalizatora w fazie *prepare* realizacji zapytania, należy skojarzyć funkcję *SampleChckFnc* z obiektem *BNStatsObject*:

```
ASSOCIATE STATISTICS WITH FUNCTIONS SampleChckFnc USING BNStatsObject.
```

4.4. Praca optymalizatora zapytań z wykorzystaniem metody wyznaczania selektywności opartej na sieci Bayesa

Po wykonaniu czynności związanych z utworzeniem reprezentacji rozkładu w postaci sieci Bayesa (podrozdział 4.2) oraz implementacji sposobu wyznaczania selektywności (podrozdział 4.3) podczas przygotowania zapytania, optymalizator użyje zaproponowaną, dokładną metodę wyznaczania selektywności, opartą na sieci Bayesa. Po napotkaniu wywołania funkcji *SampleChckFnc* w zapytaniu o następującej treści:

```
SELECT * FROM SampleTable WHERE SampleChckFnc (ca, 'N', 5, 12, 'A')=1
```

optymalizator skorzysta z opisanego rozszerzenia SZBD.

Wyznaczona wartość selektywności może mieć bezpośredni wpływ na sposób realizacji zapytania – plan wykonania (ang. *execution plan*). Na potrzeby ilustracji takiego zachowania przyjmijmy istnienie jeszcze jednej tablicy bazy danych – *SecondSampleTable* – zdefiniowanej następująco:

```
CREATE TABLE SecondSampleTable (id_2 NUMBER, name_2 VARCHAR(100), id_1 NUMBER)
```

gdzie *id_2* jest kluczem głównym, a *id_1* jest kluczem obcym, stanowiącym odniesienie do tablicy nadrzędnej – *SampleTable*.

Rozważmy plan wykonania następującego zapytania ze złączeniem tablic:

```
EXPLAIN PLAN SET STATEMENT_ID = 'demo' FOR
SELECT count(*)
FROM SampleTable t1, SecondSampleTable2 t2
WHERE t1.id = t2.id_1
AND SampleChckFnc (t1.ca, 'N', 7, 12, 'Z').
```

Dla sytuacji, w której nie była zdefiniowana statystyka oparta na sieci Bayesa, plan wykonania miał następującą postać:

```
SELECT STATEMENT
  SORT AGGREGATE
    NESTED LOOPS
      TABLE ACCESS FULL SAMPLETABLE
      TABLE ACCESS FULL SECONDSAMPLETABLE.
```

Dla sytuacji, w której była zdefiniowana statystyka, plan wykonania miał następującą postać:

```
SELECT STATEMENT
  SORT AGGREGATE
    MERGE JOIN
      TABLE ACCESS FULL SECONDSAMPLETABLE
      BUFFER SORT
      TABLE ACCESS FULL SAMPLETABLE.
```

Pierwszy z planów zakładał mało efektywną metodę łączenia tablic typu *nested loop*, dlatego czas wykonania zapytania w pierwszej opisanej sytuacji był o rząd dłuższy od czasu

wykonania tego samego zapytania w drugiej sytuacji (wybór metody łączenia typu *merge join*, przy tej samej zawartości tablic w obu sytuacjach).

5. Rozwój systemu

„Tworzenie statystyk” oparty na sieci Bayesa może być procesem czasochłonnym (choć samo wyznaczanie wartości selektywności w trakcie optymalizacji zapytania jest szybkie). Nie umniejsza to jednak przydatności tego rozwiązania, gdyż tworzenie/aktualizacja statystyk jest na ogół przeprowadzana sporadycznie, np. w ramach konserwacji bazy danych (przykładowo, przy tworzeniu kopii zapasowej bazy danych czy defragmentacji bazy danych).

Jednak i w tym zakresie (tworzenie sieci Bayesa jako reprezentacji wielowymiarowego rozkładu prawdopodobieństwa wartości atrybutów tablicy) są możliwe usprawnienia – reprezentacja rozkładu może być wykonywana przyrostowo.

Propozycja rozwiązania polega na tym, żeby na podstawie przyrostów bazy danych (nowych wierszy wstawianych do tablicy po ostatniej czynności tworzenia statystyki) tworzyć dodatkową nową reprezentację rozkładu, w postaci sieci Bayesa, skonstruowanej tylko dla przyrostu bazy. Wtedy selektywność zapytania będzie liczona jako średnia ważona selektywności wyznaczanych na podstawie każdej przyrostowej statystyki. Wagami będą tutaj współczynniki wynikające z wielkości przyrostów bazy danych (liczba nowo wstawionych wierszy), na podstawie których przyrostowe statystyki były wykonywane.

Gdyby takie podejście (zakładające tworzenie statystyk przyrostowych) było jednak niewystarczające pod względem wydajnościowym, jest możliwe uproszczenie generowania sieci Bayesa w przyrostach. W takim podejściu, w ramach tworzenia kolejnych przyrostowych statystyk, byłyby tworzone jedynie tablice warunkowych prawdopodobieństw w węzłach sieci (w uproszczeniu zakłada się więc, że struktura sieci jest jednakowa dla wszystkich przyrostów, tzn. taka jak została utworzona dla pierwszego przyrostu).

6. Podsumowanie

Szacowanie selektywności zapytań jest kluczowym elementem w procesie wyznaczania efektywnego sposobu realizacji zapytań w module optymalizatora zapytań SZBD. Problem jest szczególnie trudny dla zapytań, których warunki selekcji są określone na kilku atrybutach tablicy. Dokładne szacowanie selektywności w takim przypadku może być zrealizowane jedynie na podstawie reprezentacji wielowymiarowego rozkładu wartości atrybutów, uwzględ-

niającego korelację pomiędzy atrybutami. Bezpośrednie zastosowanie wielowymiarowego histogramu częstości jako nieparametrycznego estymatora wielowymiarowej funkcji gęstości jest na ogół niepraktyczne z powodu dużej zajętości pamięci takiej reprezentacji. Problem zajętości staje się uciążliwy dla dużej liczby wymiarów (tzw. przekleństwo wymiarowości). Stąd pojawiają się podejścia, w których proponuje się inne, przybliżone reprezentacje rozkładów wielowymiarowych, w których informacja o korelacji atrybutów byłaby uwzględniona.

Sieć Bayesa może zostać wykorzystana jako oszczędna pod względem zajętości reprezentacja wielowymiarowego rozkładu [1]. Tworzenie sieci Bayesa jest zadaniem czasochłonnym, ale algorytm wyliczania selektywności zapytania, przy wykorzystaniu sieci Bayesa, jest prosty (a więc i wydajny).

W artykule przedstawiono wybrane własności sieci Bayesa, wykorzystane przy wyznaczaniu selektywności zapytań.

Wskazano, które elementy systemu Weka (klasy) zostały użyte do implementacji funkcji tworzenia sieci Bayesa.

Na przykładach przedstawiono użycie zaproponowanej implementacji funkcjonalności, rozszerzającej SZBD Oracle w zakresie:

- tworzenia statystyki, będącej reprezentacją rozkładu prawdopodobieństwa, w postaci sieci Bayesa,
- wyznaczania selektywności wg metody, opartej na sieci Bayesa.

Dzięki wykorzystaniu modułu ODCISat [10] rozszerzono funkcjonalność optymalizatora zapytań, dzięki czemu optymalizator może zastosować zaproponowaną metodę wyznaczania selektywności, opartą na sieci Bayesa.

Dalsze prace będą koncentrować się na ilościowej ocenie omówionego rozwiązania (w tym również porównania z innymi metodami wyznaczania selektywności i innymi reprezentacjami wielowymiarowego rozkładu wartości atrybutów). W szczególności, zostaną przeprowadzone testy, dotyczące wpływu typu rozkładów wartości atrybutów i wymiarowości rozkładów na czas tworzenia i zajętość reprezentacji rozkładów.

BIBLIOGRAFIA

1. Getoor L., Taskar B., Koller D.: Selectivity estimation using probabilistic modes. Proc. of ACM SIGMOD Int. Conf. on Management of Data. ACM, New York 2001.
2. Possala V., Ioannidis Y. E.: Selectivity Estimation without the Attribute Value Independence Assumption. Proc. of the 23rd Int. Conf. on Very Large Databases, The VLDB Journal, Athens 1997.

3. Gunopulos D., Kollios G., Tsotras V.J.: Approximating Multi-Dimensional Aggregate Range Queries Over Real Attributes. ACM SIGMOD 2000, Dallas 2000.
4. Lee. J., Deok-Hwan K., Chin-Wan Ch.: Multi-dimensional Selectivity Estimation Using Compressed Histogram Estimation Information. Proc. of ACM SIGMOD Int. Conf. on Management of Data. ACM, Philadelphia 1999.
5. Chakrabarti K., Garofalakis M., Rastogi R., Shim K.: Approximate Query Processing Using Wavelets. VLDB Journal. vol. 10, no. 2-3, Springer-Verlag, New York 2001.
6. Bruno N., Chaudhuri S., Gravano L. (2001) STHoles: a multidimensional workload-aware histogram. Proc. of ACM SIGMOD Int. Conf. on Management of Data 30(2). ACM, New York 2001.
7. Kullback–Leibler divergence – Wikipedia (2010). http://en.wikipedia.org/wiki/Kullback-%E2%80%93Leibler_divergence.
8. Witten I.H., Frank E.: Data Mining: Practical Machine Learning Tools and Techniques (2 edition), Morgan Kaufmann, 2005.
9. Weka 3 – Data Mining with Open Source Machine Learning Software in Java (2010). <http://www.cs.waikato.ac.nz/ml/weka>.
10. Bouckaert R.R.: Bayesian Network Classifiers in Weka for Version 3-5-8. University of Waikato 2008.
11. Accessing Databases from WEKA: <http://weka.wikispaces.com/Databases>
12. Oracle 10g. Using extensible optimizer (2010). http://download.oracle.com/docs/cd/B14117_01/appdev.101/b10800/dciextopt.htm.

Recenzent: Dr inż. Arkadiusz Sochan

Wpłynęło do Redakcji 21 grudnia 2010 r.

Abstract

A selectivity for a simple single-table query is a fraction, where the numerator is the number of rows satisfying the given query criteria and the denominator is the total number of table rows. A query selectivity value allows to estimate size of result data. This information is required by a query optimizer which can choose the best method of the query execution. Obtaining selectivity requires an estimator of an attribute value distribution (commonly a one-dimensional histogram).

Accurate selectivity calculations for complex queries with selection condition based on many table attributes require a non-parametric estimator of multidimensional distribution of attributes values. Using multidimensional histogram as the estimator of multidimensional distribution is impractical because it is too much memory-consuming (especially for high dimensions, what is known as curse of dimensionality). Bayesian Network can be used as a memory-efficient representation of multidimensional distribution of attributes values [1].

The paper presents proposed modules supporting query selectivity estimation based on Bayesian network, fully integrated with DBMS Oracle. The article shows selected properties of Bayesian network useful in selectivity estimation. The paper enumerates those classes from Weka library which were used in the proposed implementation of Bayesian Network creation functionality. The paper presents methods of selectivity estimation realized by Bayesian Network traversal.

Finally the paper shows how to integrate with Oracle DBMS such Java-based functionality:

- creating Bayesian Network representation (so-called database statistics),
- enabling usage the Bayesian-Netwok-based selectivity method to the query optimizer.

ODCISat module allows to extend Oracle DBMS functionality. Future works may concentrate on quantitative analysis of the described solution.

Adresy

Dariusz Rafał AUGUSTYN: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, draugustyn@polsl.pl.

Łukasz WARCHAŁ: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, lukasz.warchal@polsl.pl.