Akademia Górniczo-Hutnicza im. Stanisława Staszica
Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki
Katedra Informatyki

Autoreferat rozprawy doktorskiej

Pozyskiwanie wiedzy z dużych zbiorów danych z zastosowaniem adaptacyjnych procedur generowania zapytań[1]

mgr inż. Bartosz Jędrzejec

PROMOTOR: Prof. zw. dr hab. inż. Edward Nawarecki

Kraków, 2009

1. Wstęp

Wraz z rozwojem systemów informatycznych następuje stały wzrost ilości generowanych, gromadzonych i przetwarzanych danych. Duża objętość zasobów przechowywanych danych, często rzędu wielu tysięcy, a nawet milionów rekordów, powoduje trudności w wydobywaniu z tych danych informacji użytecznych z punktu widzenia potencjalnych użytkowników (lekarzy, menadżerów, ekonomistów, technologów). Z tego względu, w ostatnich latach wzrosło zapotrzebowanie na metody i narzędzia do efektywnego pozyskiwania informacji ukrytych w bazach i hurtowniach danych.

Jednym z najważniejszych problemów, jaki należy w tym zakresie rozwiązać, jest opracowanie odpowiednich modeli drążenia danych oraz dobór wartości parametrów tych modeli. Model zbyt skomplikowany powoduje utrudnienia w odkrywaniu wiedzy istotnej z punktu widzenia analityka. Natomiast model uproszczony jest wprawdzie łatwy w analizie, lecz pewne istotne dla analityka dane i informacje mogą zostać w nim nie uwzględnione. Z tego powodu poszukuje się nowych metod umożliwiających generowanie modeli uproszczonych, lecz zorientowanych na opis tylko wybranych zależności interesujących analityka dziedzinowego (użytkownika systemu). Uzyskanie uproszczenia modelu jest możliwe poprzez:

1. nałożenie ograniczeń przed stworzeniem modelu, lub
2. dokonanie uproszczeń w modelu już wygenerowanym.

Celem niniejszej pracy było opracowanie metodyki i zaprojektowanie procedur analizy dużych, wygenerowanych w oparciu o bazę danych, modeli asocjacyjnych. Modele te, zapisane w wywodzącym się z XML standardzie PMML, mogą być w łatwy sposób analizowane przy pomocy zapytań w języku XQuery. Prawidłowe sformułowanie zapytania w języku zbliżonym do naturalnego powinno generować właściwy podzbiór reguł, który analityk może wykorzystać w procesie wnioskowania na poparcie postawionej przez siebie hipotezy.

W pracy zaproponowano metodę, która w oparciu o algorytm programowania genetycznego poszukuje coraz to lepszych zapytań do modelu reguł asocjacyjnych. Dzięki temu podzbiory reguł zwracane przez zapytania są ograniczane jedynie do tych, które są interesujące z punktu widzenia celu przeprowadzanej analizy. Dodatkowo liczba zwracanych reguł jest na tyle mała, że są one łatwe do szybkiego przeanalizowania.

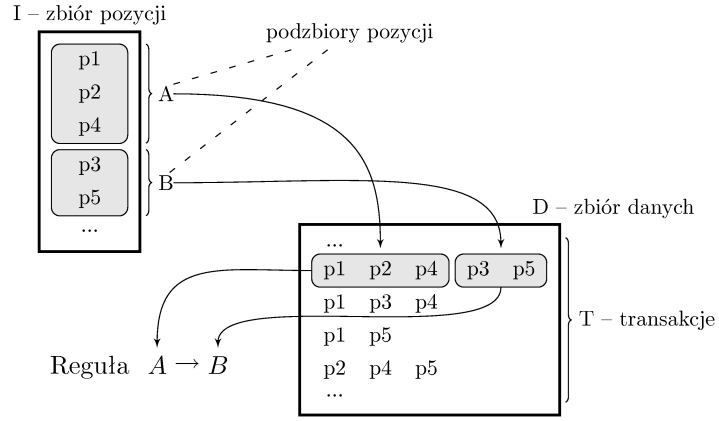
---

1 Praca powstała przy wsparciu finansowym Ministerstwa Nauki i Szkolnictwa Wyższego w postaci projektu badawczego promotorskiego nr N516 026 31/2545.

Tezą niniejszej pracy jest następujące stwierdzenie:
Zastosowanie adaptacyjnych procedur formułowania zapytań umożliwia skuteczne pozyskiwanie problemowo zorientowanej wiedzy z dużych zbiorów danych analitycznych.

2. Reguły asocjacyjne

Model oparty na wykorzystaniu reguł asocjacyjnych jest jednym z ważniejszych typów modeli drążenia danych. Reguły asocjacyjne pozyskiwane są poprzez analizę zbioru danych D (rys. 1), często nazywanego zbiorem transakcji, który składa się z rekordów T. Każda transakcja składa się z pozycji ze zbioru I. Dla każdej transakcji T i podzbioru pozycji A, mówimy, że T zawiera A, gdy $A \subseteq T$.



Rys. 1. Pozyskiwanie reguł asocjacyjnych

Regułę asocjacyjną można przedstawić w postaci $A \rightarrow B$, przy czym $A \cap B = \varnothing$ (A, B – podzbiory pozycji, które pochodzą z pewnego zbioru pozycji I).
Dla określenia w jakim stopniu dana reguła $A \rightarrow B$ jest interesująca, najczęściej stosuje się miary tzw. wsparcia i zaufania reguły.

3. Problem redukcji modeli

Podczas realizacji etapu budowy modelu drążenia, aspekt doboru odpowiednich parametrów tego modelu jest bardzo istotny. Dla modelu asocjacyjnego są to wartości progowe wsparcia i zaufania dla reguł asocjacyjnych. Założenie wysokich wartości tych parametrów skutkuje wygenerowaniem modelu o małej ilości reguł, a co za tym idzie modelu przejrzystego - łatwego w analizie. W takim przypadku jednak istotne dane, które mogą interesować analityka, mogą zostać pominięte, a w konsekwencji uzyskana wiedza jest niepełna. Gdy wartości progowe parametrów są zbyt małe - generowana jest duża liczba reguł, co powoduje znaczną złożoność modelu oraz trudność w wydobywaniu najbardziej istotnej dla analityka wiedzy dziedzinowej. Tutaj nie następuje utrata interesujących informacji, lecz są one ukryte i trudno dostępne dla użytkownika. W takich przypadkach zastosowanie metod wydobywania istotnych dla danego zadania reguł staje się sprawą kluczową.
W znanych autorowi pracy publikacjach, opisujących problematykę redukcji modelu z wykorzystaniem języków zapytań, występują trzy istotnie różniące się podejścia do rozwiązania tego problemu:
– redukcja poprzez ograniczenie danych źródłowych (np. język DMQL, Operator MINE RULE, język Mine-SQL),
– redukcja poprzez ograniczenia dla generowanych reguł (np. język MineSQL, język MSQL),
– redukcja przy użyciu zapytań do modelu (np. język MSQL, SMARTSKIP System).

Porównując przedstawione metody redukcji (tab. 1), można stwierdzić, iż metoda przy użyciu zapytań do modelu jest bardziej obiecująca niż pozostałe, ze względu na możliwość wielokrotnego zadawania zapytań i badania podzbiorów reguł, bez potrzeby ponownego budowania modelu. Należy zwrócić uwagę, ze żadne ze znanych rozwiązań nie umożliwia automatycznego zadawania zapytań w celu uzyskiwania odpowiedzi adekwatnych do potrzeb analityka. Nie są to również rozwiązania uniwersalne, umożliwiające wykorzystanie ich dla różnych platform sprzętowych i programowych. Dlatego też, stworzenie metodyki umożliwiającej analizę dużych zbiorów reguł asocjacyjnych, wraz z zestawem narzędzi programowych dających możliwość

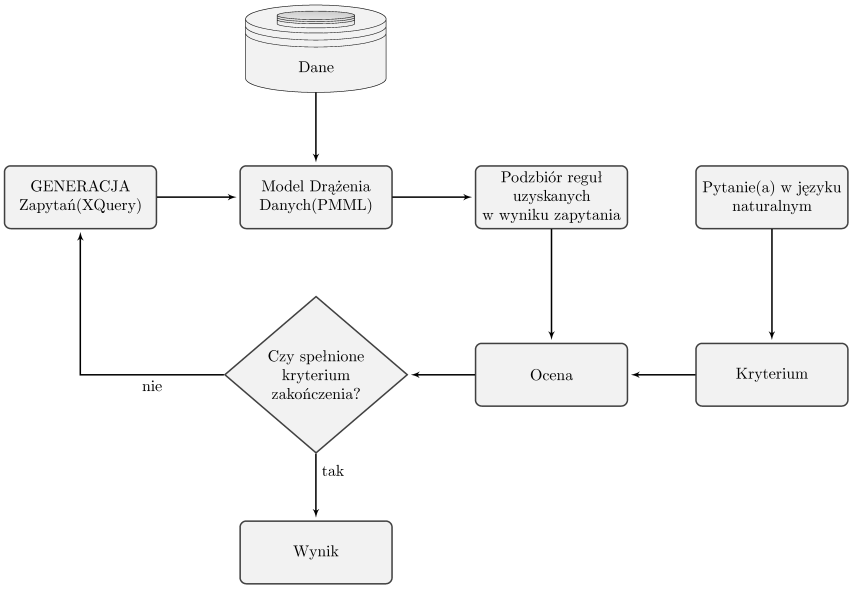
realizacji tej metodyki, wydaje się zarówno z badawczego jak i praktycznego punktu widzenia zadaniem zasadnym i interesującym.

Tabela 1 Porównanie rozwiązań redukcji modeli drążenia

Typ redukcji	Zalety	Wady
ograniczenie danych wejściowych	– budowa modelu odbywa się tylko dla wyspecyfikowanych danych – liczba wygenerowanych reguł jest mniejsza niż w przypadku analizy całości danych	– zmiana założeń analizy wymusza konieczność budowy nowego modelu – analiza danych cząstkowych może doprowadzić do niepoprawnych wyników i wniosków
ograniczenie dla generowanych reguł	– we wszystkich regułach modelu występują wskazane przez użytkownika pozycje – łatwość analizy małej (zazwyczaj) liczby reguł	– zmiana założeń analizy wymusza konieczność budowy nowego modelu
ograniczenie przez użycie zapytań	– generowane są wszystkie reguły (powyżej progu wsparcia i zaufania) - nie ma potrzeby budowy nowego modelu przy zmianie założeń analizy	– trudność doboru odpowiedniego zapytania zwracającego oczekiwany wynik

4. Metoda automatycznego generowania zapytań

Z rozważań przeprowadzonych w rozprawie można wysnuć wniosek, ze najkorzystniejszą metodą analizy modelu reguł asocjacyjnych jest zadawanie zapytań do zbioru reguł, który został zbudowany bez ograniczeń dotyczących danych źródłowych oraz generowanych reguł. Jednak z powodu dużej liczby reguł zadanie prawidłowego zapytania nie jest rzeczą łatwą i często uzyskany efekt jest niezadowalający. Prowadzi to do sytuacji. w której otrzymanie oczekiwanego wyniku wymaga zadania wielu zapytań, co realizowane manualnie jest zajęciem czasochłonnym i często niewykonalnym. Dlatego też, zadaniem proponowanej metody jest automatyczne generowanie coraz lepszych zapytań do bazy reguł, a co za tym idzie poszukiwanie optymalnego rozwiązania pod względem zadanego kryterium. Schemat opracowanej metody przedstawiono na rysunku 2.



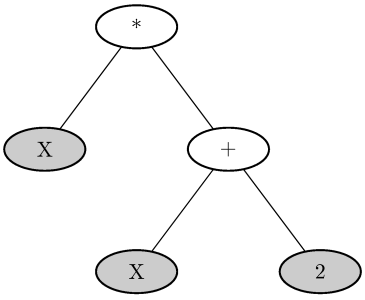
Rys. 2. Schemat metody automatycznego generowania zapytań

W pierwszej kolejności budowany jest model zdefiniowany przez zbiór reguł asocjacyjnych. Zapisywany jest on w standardzie PMML, a następnie zadawane jest zapytanie do modelu w języku XQuery. Zapytanie to zwraca pewien podzbiór reguł, który następnie poddawany jest ocenie względem przyjętego wcześniej kryterium. Jeśli warunek zakończenia jest spełniony, zwracane jest uzyskane rozwiązanie, w przeciwnym wypadku generowane jest kolejne zapytanie i cały cykl powtarza się. W przypadku tworzenia kolejnego

zapytania podstawowym zadaniem jest znalezienie takiego nowego zapytania, aby generowało ono podzbiór reguł lepszy od poprzedniego.
Drugim problemem zaproponowanej metody jest dobranie takiego kryterium oceny, które musi łączyć w sobie elementy wskazane przez użytkownika oraz mechanizmy ograniczające liczebność zbioru reguł wynikowych do asocjacji najważniejszych z punktu widzenia prowadzonej analizy.
W pracy przedstawione zostały rozwiązania obydwóch w/w problemów przy wykorzystaniu algorytmów programowania genetycznego oraz zdefiniowanego kryterium oceny.

5. Programowanie genetyczne

Programy ewolucyjne stały się w ostatnim czterdziestoleciu jedną z bardzo szybko rozwijających się gałęzi wiedzy w dziedzinie sztucznej inteligencji. Pod tym pojęciem należy rozumieć całą klasę systemów opartych na mechanizmie doboru naturalnego i dziedziczenia.
Mechanizm ten polega na stworzeniu populacji osobników, która jest pewnym zbiorem rozwiązań zadanego problemu. Każdy z nich poddawany jest ocenie, która określa stopień, w jakim dane rozwiązanie spełnia założenia funkcji celu. Następnie na podstawie reprodukcji i modyfikacji najlepszych osobników tworzona jest kolejna populacja, w której spodziewamy się znaleźć rozwiązanie lepsze od dotychczas uzyskanych.
W metodzie programowania genetycznego, przedstawionej przez Johna R. Kozę na początku lat dziewięćdziesiątych, zaproponowane zostało, aby zamiast budowania jednego programu ewolucyjnego rozwiązującego zadany problem, przeszukiwać przestrzeń programów. Programy te traktowane są jako populacja osobników, które ewoluują w kolejnych pokoleniach w celu znalezienia jak najlepszego, w sensie przyjętego kryterium, rozwiązania.
Struktura osobnika w programowaniu genetycznym, która podlega procesowi adaptacji ma formę drzewa. Drzewo to reprezentuje program komputerowy, w którym funkcje programu znajdują się w węzłach, natomiast argumenty funkcji (tzw. terminale) w liściach drzewa. Rolę funkcji mogą spełniać, w zależności od zastosowanego języka programowania, operatory: matematyczne, warunkowe, logiczne lub funkcje: matematyczne, iteracyjne i inne. Argumentami funkcji mogą być zmienne, stałe, struktury, obiekty itp.
Na rysunku 3 przedstawiono przykład drzewiastej struktury osobnika. W pokazanym przykładzie osobniki populacji są wzorami funkcji zmiennej X. Ze zbioru operatorów matematycznych F={*,/,+,-} (mnożenie, dzielenie, dodawanie i odejmowanie) oraz zbioru terminali T={stała ∈ N, zmienna ∈ {X}} (stała należąca do liczb naturalnych oraz X należące do zbioru zmiennych) tworzone jest drzewo, które jest odpowiednikiem wyrażenia matematycznego X*(X+2).



Rys. 3. Drzewiasta struktura osobnika w programowaniu genetycznym

Do głównych operacji ewolucyjnych należą: reprodukcja, krzyżowanie i mutacja. Zasada działania operacji ewolucyjnych w programowaniu genetycznym jest podobna do tych stosowanych w innych metodach ewolucyjnych. Różnica polega na tym, ze w tym przypadku operacje modyfikacji wykonywane są na strukturach drzewiastych.

6. Standard PMML oraz język zapytań XQuery

PMML (*Predictive Model Markup Language*) jest opartym na języku XML, standardem tekstowego zapisu modeli statystycznych i drążenia danych. Język ten został wprowadzony przez Data Mining Group – organizację, która definiuje nowe standardy w dziedzinie drążenia danych. W jej skład wchodzi wiele wiodących firm takich jak: IBM, Microsoft, Oracle, SAS, SPSS i inne. Standard PMML, podobnie jak i inne oparte na bazie XML, zdobywa w ostatnim czasie coraz większą popularność i przewiduje się, że tego typu standardy staną się w najbliższym czasie podstawą systemów drążenia danych i pozyskiwania wiedzy.

Przykład zapisu reguły w języku PMML:

<AssociationRule confidence=”0.58821” support=”0.01477”consequent=”1” antecedent=”25” />

Tak zapisaną regułę można zinterpretować następująco: reguła asocjacyjna, której poprzednik (zbiór częsty) ma identyfikator 25, a następnik identyfikator 1 o wartości parametru wsparcia równej 0.01477 i zaufania =0.58821.

Pozostałe elementy modelu takie jak pozycje, zbiory częste są również definiowane w modelu za pomocą odpowiednich znaczników zdefiniowanych w tym standardzie.

Pierwsza wersja języka XQuery, jako tzw. working draft, została przedstawiona w lutym 2001 roku. Od tego czasu trwały prace nad rozwojem tego języka, aby w styczniu 2007 stał się on oficjalnym standardem, wskazywanym przez organizację World Wide Web Consortium, jako język zapytań dla dokumentów XML.

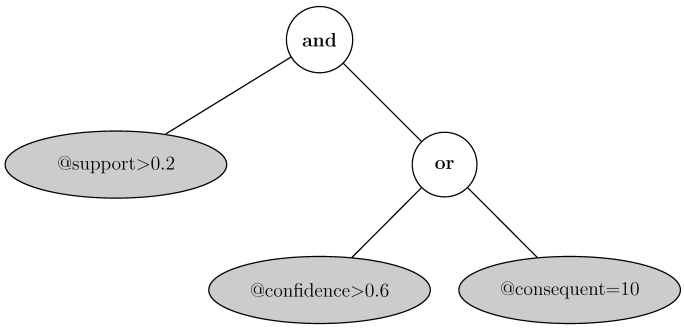
Główną zaletą języka XQuery jest zwięzła, przypominająca stylem język zapytań SQL, struktura zapytania nazywana w skrócie „FLWOR". Skrót ten pochodzi od podstawowych instrukcji występujących w zapytaniach XQuery: **for** (instrukcja pętli), **let** (instrukcja przypisania wartości), **where** (instrukcja warunku), **order by** (instrukcja sortowania) i **return** (instrukcja zwracająca wynik). Oprócz wymienionych wyrażeń język XQuery wyposażony jest również w instrukcję warunkową oraz wiele wbudowanych funkcji bibliotecznych, które można uzupełnić o deklaracje własnych procedur.

```
for $i in doc(“model.xml”)//AssociationRule
where $i/@support>0.2
order by $i/@confidence descending
return $i
```

Przedstawione powyżej zapytanie wyselekcjonuje wszystkie reguły z modelu zapisanego w zbiorze model.xml, których parametr wsparcia przekracza wartość 0.2 oraz wynik zapytania jest uporządkowany malejąco według parametru zaufania.

7. Postać osobnika i kryterium jego oceny

Analizując zapytanie XQuery wyszukujące reguły asocjacyjne z modelu zapisanego w języku PMML, można stwierdzić, że podstawą selekcji reguł ze zbioru danych jest określenie warunku w klauzuli **where**. Zmiana warunku selekcji powoduje wybór innego podzbioru reguł, dlatego też w proponowanej metodzie ten element zapytania jest poddawany zmianom ewolucyjnym. W tym celu zdefiniowano zbiór funkcji F={and,or,not} oraz składający się z jednego elementu zbiór terminatorów T={wyrażenie}. Każde wyrażenie składa się z 3 składników: nazwy atrybutu (@support, @confidence, @consequent lub @antecedent), operatora porównania oraz wartości rzeczywistej lub całkowitej.




```
for $i in doc(“model.xml”)//AssociationRule
where [@support>0.2 and (@confidence>0.6 or @consequent=10)]
order by $i/@confidence descending
return $i
```

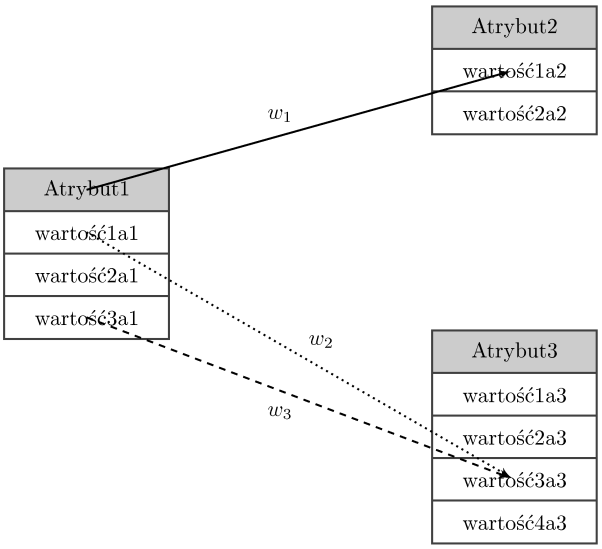
Rys. 4. Rozbudowany warunek zapytania w strukturze drzewiastej

Na rysunku 4 przedstawiono rozbudowany warunek zapytania oraz jego reprezentację w postaci struktury drzewiastej. W pracy opracowano odpowiednie algorytmy dla potrzeb budowy osobnika oraz wyrażenia.

Dla prawidłowego działania algorytmu genetycznego niezbędny jest dobór odpowiedniego kryterium oceny osobników populacji. W proponowanej metodzie automatycznego generowania zapytań, zadaniem kryterium jest ocena podzbioru reguł asocjacyjnych zwróconych przez zapytanie.

Zaproponowane w pracy kryterium łączy w sobie trzy elementy: miarę subiektywną, miarę obiektywną oraz parametry wyszukiwania, które zagregowane wykorzystane są do oceny zapytania.

Miara subiektywna jest określana przez użytkownika i wyznaczenie jej realizowane jest w kilku etapach:
1. zdefiniowanie zbioru wag *W* – czyli rosnącego ciągu liczb całkowitych, których i-ty element określa stopień ważności powiązania
2. wybór atrybutów z bazy danych, które będą brać udział w analizie, a następnie zdefiniowanie powiązań pomiędzy wartościami tych atrybutów (rys. 5) – przy czym powiązania mogą być zdefiniowane pomiędzy dwoma pojedynczymi wartościami atrybutów (siłę tego powiązania określają wagi w_2, w_3) lub pomiędzy atrybutem (wszystkimi jego wartościami), a wartością pojedynczą atrybutu lub odwrotnie (waga w_1)



Rys. 5. Definicja wzorca kryterium; w_1,w_2,w_3 – wagi powiązań, wartość j a x – j-ta wartość atrybutu x

Na podstawie tak zdefiniowanego wzorca, każdej pozycji w modelu przypisana jest znormalizowana wartość wagi odpowiednio dla poprzednika i następnika reguły. Wartością miary subiektywnej (P_s) jest średnia wag pozycji dla reguł zwróconych przez zapytanie.
Następnie użytkownik określa jakiego typu reguły mają być poszukiwane – potwierdzające wzorzec lub nowe wcześniej nieznane, ale powiązane ze zdefiniowanym wzorcem.

Miara obiektywna (P_o) tworzona jest natomiast poprzez uśrednienie wartości *J-miary* (lub *J'-miary* w celu wyszukiwania reguł nowych wcześniej nieznanych) dla wszystkich reguł zwróconych przez zapytanie.

Dodatkowo, w celu uszczegółowienia rodzaju oczekiwanych rezultatów, określa się 2 parametry:
- oczekiwaną liczbę reguł (maksymalna liczba reguł), zwróconą przez zapytanie N_O, na podstawie którego obliczany jest współczynnik oczekiwanej liczby reguł D_N, oraz
- współczynnik oceniający liczbę reguł znaczących D_R (czyli takich, które mają przynajmniej 2 pozycje wskazane przez użytkownika we wzorcu kryterium dla reguł potwierdzających wzorzec lub dokładnie 1 pozycję dla poszukiwania nowych reguł powiązanych ze wzorcem).

W rezultacie funkcja przystosowania definiowana jest następująco:

$$F = \frac{(P_S + P_O)}{2} \cdot D_N \cdot D_R$$

Parametry równania zdefiniowano powyżej.

8. Implementacja

Dla potrzeb testowania metody automatycznego generowania zapytań, została opracowana oraz zaimplementowana aplikacja komputerowa o nazwie GAZdRA (Generowanie Automatyczne Zapytań do Reguł Asocjacyjnych), dzięki której możliwe było sprawdzenie poprawności działania opracowanej metody na danych rzeczywistych. Oprogramowanie składa się z 5 modułów:
– moduł pobierania danych,
– moduł definicji kryterium,
– moduł budowy modelu reguł asocjacyjnych,
– model procesu programowania genetycznego,
– moduł wizualizacji wyników.

Przy tworzeniu oprogramowania w języku Java wykorzystano środowisko programistyczne NetBeans IDE w wersji 6.0, wybrane biblioteki programistyczne udostępnione na zasadzie licencji Open Source oraz zaimplementowane rozwiązania własne.

9. Wyniki badań

W celu sprawdzenia skuteczności opracowanej metody automatycznego generowania zapytań przeprowadzono serię badań eksperymentalnych, wykorzystując aplikację GAZdRA. Do budowy modeli drążenia koniecznym było pozyskanie danych rzeczywistych, które następnie poddawane były analizie. Do badań udało się pozyskać dane z trzech źródeł:
– dane o zużyciu energii elektrycznej, udostępnione przez Zakład Energetyczny,
– dane medyczne z badań alergologicznych, przeprowadzonych przez Zakład Alergologii Szpitala Uniwersyteckiego w Krakowie,
– dane o wypadkach samochodowych w Stanach Zjednoczonych (ogólnie dostępne w sieci internet).

Przeprowadzona analiza wyników badań prowadzi do stwierdzenia, że metoda automatycznego generowania zapytań, w kolejnych krokach procesu ewolucyjnego pozwalała znaleźć takie zapytania do modelu, aby zwrócony przez nie podzbiór reguł był łatwiejszy w analizie niż liczące setki czy nawet tysiące reguł modele analizowane we wstępnym etapie badań. Liczba reguł pozyskanych w wyniku zadania „optymalnego'' zapytania do modelu była zredukowana do wartości oczekiwanej przez użytkownika (eksperta dziedzinowego) lub nieznacznie się od niej różniła. Dzięki temu specjaliści dziedzinowi mają ułatwione zadanie podczas analizy, często bardzo dużych zbiorów danych.
Podzbiór reguł, zwracanych przez najlepsze zapytanie znalezione przez algorytm, zawierał w większości przypadków tylko reguły znaczące, które na podstawie oświadczenia użytkownika okazały się dla niego interesujące.

10. Podsumowanie

Jako najważniejsze i oryginalne osiągnięcia wynikające z przeprowadzonych badań, zdaniem autora, można wskazać:
– stworzenie koncepcji zwiększenia efektywności pozyskiwanej wiedzy z dużych zbiorów danych, opartej na automatycznym generowaniu zapytań, z zastosowaniem programowania genetycznego,
– skonstruowanie kryterium oceny otrzymanych rozwiązań, stanowiącego połączenie kryterium subiektywnego (definiowanego przez użytkownika) z kryterium obiektywnym opartym na zastosowaniu *J-miary*,
– implementację zaproponowanej metody w aplikacji, stanowiącej narzędzie umożliwiające zarówno weryfikację samej metody, jak też rozwiązywanie określonej klasy zadań praktycznych (co potwierdzają przeprowadzone eksperymenty numeryczne).

Publikacje doktoranta:

1. K. Świder, B. Jędrzejec: A Query-Driven Exploration of Association Rule Models in PMML. In R. Tadeusiewicz, A. Ligeza, and M. Szymkat, editors, Proc. 5th Int. Conference Computer Methods and Systems, pages 409–414, Oprogramowanie Naukowo-Techniczne, 2005.
2. B. Jędrzejec, E. Czarnobilska, G. Porębski, K. Obtułowicz, E. Nawarecki: The Estimation of Reliability In The Preventive Examination Of Allergic Diseases With Knowledge Discovery Methods. In Computers in Medical Activity, 2008.
3. B. Jędrzejec, E. Nawarecki: Discovery of knowledge in electrical Power systems. In 9th IFAC Workshop on Intelligent Manufacturing Systems (IMS'08), 75–80, 2008.
4. K. Świder, B. Jędrzejec, M. Wysocki: A Query Driven Exploration of Multiple Association Rules. In C. Cotta, S. Reich, R. Schaefer, and A. Ligeza, editors, Knowledge-Driven Computing, Knowledge Engineering and Intelligent Computations Series: Studies in Computational Intelligence, volume 102, pages 283–288, 2008.