WYŻSZA SZKOŁA
INFORMATYKI STOSOWANEJ
I ZARZĄDZANIA

ANALIZA SYSTEMOWA W FINANSACH I ZARZĄDZANIU

Wybrane problemy

Pod redakcją
Jerzego HOŁUBCA

WYŻSZA SZKOŁA
INFORMATYKI STOSOWANEJ
I ZARZĄDZANIA

ANALIZA SYSTEMOWA W FINANSACH I ZARZĄDZANIU
Wybrane problemy

Pod redakcją
Jerzego HOŁUBCA

Warszawa 1999

Wykaz opiniodawców artykułów zamieszczonych w tomie:

prof. dr hab. Jerzy **HOŁUBIEC**

prof. dr hab. Janusz **KACPRZYK**

prof. dr hab. Tadeusz **NOWICKI**

prof. dr hab. Stanisław **PIASECKI**

prof. dr hab. Piotr **SZCZEPANIAK**

prof. dr hab. Tadeusz **TRZASKALIK**

doc. dr hab. Sławomir **WIERZCHOŃ**

doc. dr hab. Leszek **ZAREMBA**

Warszawa 1999

ISBN 83-85847-24-3

Przedmowa

Na niniejszą publikację składa się zbiór prac doktorantów Zaocznych Studiów Doktoranckich "Informatyka w zarządzaniu i finansach" działających przy ***Instytucie Badań Systemowych Polskiej Akademii Nauk***.

Prace te były referowane na konferencji BOS'98 "Rozwój średnich i małych miast w XXI wieku w Polsce: Rola badań operacyjnych i systemowych", Kutno, 8-10 czerwca 1998 r.[1], a także na seminariach Studiów Doktoranckich "Informatyka w zarządzaniu i finansach". Nad stroną merytoryczną publikacji czuwał Pan Prof. dr hab. Jerzy Hołubiec oraz grono recenzentów i opiekunów naukowych doktorantów.

Prace dotyczą głównie problemów analizy systemowej oraz jej zastosowań w dziedzinie finansów, a zwłaszcza - teorii portfela, obligacji i problemów inwestycyjnych. Niektóre prace przy analizie finansowej posługują się tzw. algorytmami genetycznymi i sieciami neuronowymi, a także modelowaniem rozmytym i strukturami fraktalnymi. Część prac dotyczy zarządzania i sterowania produkcją.

Wypada zauważyć, iż doktoranci Studiów atakują w swych pracach tematy nowoczesne i znajdujące się w obszarze tzw. frontu badawczego analizy systemów. Wypada im życzyć sukcesów i wytrwałości w pracy, która winna zakończyć się obronioną pracą doktorską.

---

[1] Głównymi organizatorami konferencji było Polskie Towarzystwo Badań Operacyjnych i Systemowych oraz Instytut Badań Systemowych PAN.

Wypada także zaznaczyć, iż wydanie niniejszej publikacji stało się możliwe dzięki wsparciu finansowemu ze strony ***Wyższej Szkoły Informatyki Stosowanej i Zarządzania***, działającej w ramach Fundacji Krzewienia Nauk Systemowych. Fundacja ta została założona w 1991 roku z inicjatywy Prof. L. Kuźnickiego, wówczas Sekretarza Naukowego Polskiej Akademii Nauk. Do zadań Fundacji należy, między innymi, wspieranie i promocja prac młodych pracowników nauki, a zwłaszcza prac doktorantów.

Mamy nadzieję, iż publikacja niniejsza zostanie życzliwie przyjęta przez specjalistów działających w obszarze nauk systemowych.

Rektor WSISiZ

Prof. Roman Kulikowski

LINGWISTYCZNE PODSUMOWANIA BAZ DANYCH Z UŻYCIEM LOGIKI ROZMYTEJ I ALGORYTMÓW GENETYCZNYCH

Paweł STRYKOWSKI
Zaoczne Studia Doktoranckie IBS PAN

Rozpatruje się problem podsumowania zawartości bazy danych (krótkim) stwierdzeniem w języku (quasi)naturalnym. Proponuje się algorytm znajdowania takich podsumowań. W sformułowaniu zadania i algorytmie stosuje się logikę rozmytą z kwantyfikatorami lingwistycznymi. Pozwala to na formalne ujęcie nieprecyzyjności lingwistycznym określeń wartości atrybutów oraz częstości występowania określonych wartości atrybutów. Podano wskaźniki jakości takich podsumowań lingwistycznych. Zastosowano algorytmy genetyczne do efektywnego wyboru najbardziej trafnych (o największej wartości przyjętych wskaźników jakości) podsumowań.

1. Wstęp

Obliczono, że ilość informacji zapisywanych na nośnikach komputerowych podwaja się co 20 miesięcy. Wielkość oraz liczba baz danych prawdopodobnie zwiększa się jeszcze szybciej. Automatyzacja zapisu większości procesów dotyczących działalności gospodarczej powoduje rosnący strumień informacji wpisywanych do komputera. Przykładem mogą być tak proste operacje, jak np. rozmowy telefoniczne, użycie karty kredytowej, wykonanie testu medycznego, czy sprzedaż towaru. Istnieje jednak duża różnica między dysponowaniem

danymi a ich właściwym zrozumieniem i sensownym wykorzystaniem. Właściwie nie same dane są tu najważniejsze, ale reprezentowane przez nie zależności, czyli tzw. *wiedza*.

Tę wiedzę trzeba jednak najpierw z danych wydobyć (odkryć). *Wydobywanie* (*odkrywanie*) *wiedzy* z ogromnych baz danych nie jest procesem trywialnym i prostym, często zależnym od specyfiki informacji. Powstało wiele metod i technik wspomagające wydobywanie wiedzy, takich jak: metody probabilistyczne i statystyczne, zbiory (logika) rozmyte, sieci neuronowe, zbiory przybliżone, algorytmy uczenia maszynowego itp. Badania w tej dziedzinie obejmują m.in. analizy statystyczne, generowanie odpowiedzi na zapytania o wysokim poziomie złożoności, interaktywne uzyskiwanie wiedzy na podstawie zapytań do użytkownika, zarządzanie niedokładnością i niepewnością, poszukiwanie korelacji między danymi, podsumowanie lingwistyczne itp. Podsumowywanie bazy danych polega na automatycznym bądź interaktywnym generowaniu zdań w formie jak najbardziej zrozumiałej dla użytkownika. Powinno być odpowiedzią na zapytanie użytkownika wyrażone jak najprostszym sposobem. Podsumowanie to powinno być możliwie krótkie i powinno zawierać jak najważniejsze informacje o zależnościach znalezionych w danej bazie danych.

Podsumowanie składać się może z zdań typu [2, 10, 11, 14]:

Określenie liczności + określenie podmiotu + określenie cechy

W niniejszej pracy przedstawia się nową metodę opartą na logice rozmytej. Użytkownik, bądź ekspert, definiuje wartości atrybutów występujących w bazie danych w języku naturalnym, w odpowiedniej formie i przypadku. Te wartości lingwistyczne określa się jako odpowiednie zbiory rozmyte. Przy definiowaniu wartości atrybutów i odpowiadających im wartości rozmytych można korzystać z funkcji transformujących wielkości nienumeryczne na numeryczne.

Zdefiniowane zostały wskaźniki jakości zdania składającego się na ostatecznie podsumowanie. Podstawowe wskaźniki oceniają prawdziwość zdania, liczność (tzn. jak wiele rekordów takie zdanie adekwatnie reprezentuje) i trafność. Podsumowania zawierają ograniczoną liczbę możliwych zdań. Zdania są związane są ze sformułowanym

przez użytkownika zapytaniem opierającym się na wcześniej określonych atrybutach i wartościach rozmytych. Program wybiera najlepsze zdania, czyli takie, dla których sumaryczny wskaźnik jakości przyjmuje największą wartość. Sprawdzane są wszystkie możliwe podsumowania (zdania). Przy przeszukiwaniu stosuje się algorytmy genetyczne, których idea jest oparta na procesach obserwowanych w ewolucji naturalnej.

Omawia się oprogramowanie implementujące zaproponowaną metodę i algorytm. Oprogramowanie to jest testowane na rzeczywistej bazie danych związanej ze sprzedażą w firmie handlującej sprzętem komputerowym. Podano przykłady otrzymanych podsumowań i ich analizę.

2. Podsumowania lingwistyczne

Załóżmy, że dany jest duży zbiór danych (baza danych), i niech V będzie atrybutem, który przyjmuje wartości w zbiorze $X = \{x_1, x_2, ...\}$. Na przykład, V może być „wiekiem", „doświadczeniem" itp. Niech $Y = \{y_1, ..., y_n\}$ będzie zbiorem obiektów (np. ludzi) dla których można określić wartości atrybutu V. Niech $V(y_i)$ wskazuje wartości atrybutu dla obiektu y_i.

Tak więc, dane które chcemy podsumować są zbiorem:

$$D = \{ V(y_1), V(y_2), ..., V(y_n)\}.$$

gdzie, jeśli np. Y jest grupą n ludzi i V jest wynagrodzeniem, to D jest zbiorem określającym wynagrodzenia poszczególnych osób.

Lingwistyczne podsumowanie bazy danych D (w sensie Yagera [14]) składa się z:

- określenia grupy obiektów S,
- określenia liczności Q,
- współczynnika jakości (zgodności, trafności, ...) T.

Na przykład, jeśli D zawiera zbiór ludzi (np. pracowników), to podsumowaniem może być:

„Wielu ludzi ma wysokie wynagrodzenie (0.78)"

gdzie „ludzie mający wysokie wynagrodzenie" to określenie grupy obiektów S. „Wielu" to określenie liczności Q, a „0.78" to wartość współczynnika jakości (zgodności, trafności, ...) podsumowania T określonego według ustalonego algorytmu.

Wartość T można obliczyć stosując rachunek zdań z rozmytymi kwantyfikatorami lingwistycznymi zaproponowany przez Zadeha [16]. Zgodnie z tą metodą, jeśli V jest zbiorem rozmytym w X, to określenie S jest też zbiorem rozmytym w X. Ponadto jeśli Q będzie rozmytym słownym (względnym) kwantyfikatorem (wg. definicji Zadeha [16]), takim że Q: [0,1] $\rightarrow$ [0,1], wtedy T możemy obliczyć według następującego algorytmu:

1. dla każdego $d \in D$ obliczamy $S(d_i) \in [0,1]$, czyli stopień w jakim d_i spełnia określenie S,
2. obliczamy, jaka część D spełnia S:

$$r = \frac{1}{n} \sum_{i=1}^{n} S(d_i) \tag{1}$$

3. obliczamy T jako wartość funkcji przynależności określenia liczności (kwantyfikatora lingwistycznego) Q dla wartości r, tzn.

$$T = Q(r) \tag{2}$$

Przypuśćmy że nasz zbiór danych posiada więcej atrybutów. Niech $V=\{V_1,V_2,..., V_m\}$ będzie zbiorem tych atrybutów. Niech $V_j(y_i)$ określa wartość atrybutu V_j dla obiektu y_i.

Tak więc dane, które chcemy podsumować, są zbiorem:

$$D=\{d_1, d_2, ...,d_n\} = \{V_1(y_1), V_2(y_1),..., V_m(y_1),$$

$$V_1(y_2), V_2(y_2),..., V_m(y_2),$$

$$V_1(y_n), V_2(y_n),..., V_m(y_n)\} \quad (3)$$

Określenie grupy obiektów S jest rodziną zbiorów rozmytych $S=\{s_1, s_2,,... s_m,\}$, gdzie s_i jest zbiorem rozmytym w X_1.

Wtedy $S(d_i)$ może być określane następująco:

$$S(d_i)= \min_{j=1, 2, ..., m} (s_j(V_j(y_i))).$$

3. Algorytm generacji podsumowania lingwistycznego

Potrafimy już więc obliczyć wartość współczynnika jakości T dla zaproponowanego podsumowania – zdania składającego się z określenia liczności i określenia grupy obiektów. Poszukiwanie najlepszego podsumowania (pojedynczego zdania) sprowadza się do sprawdzenia wszystkich możliwych do wygenerowania zdań i wybrania najlepszego. Dla podsumowań i baz składających się z wielu atrybutów jest to zadanie dosyć złożone w sensie obliczeniowym. Dla i atrybutów oraz j wartości rozmytych dla każdego z nich należało by więc sprawdzić j^i zdań. Za każdym razem sprowadza się to do przejrzenia całej bazy danych, co może być dosyć długotrwałe. Dlatego też należy ograniczyć liczbę dopuszczalnych kombinacji.

Można to zrobić na cztery sposoby:

- można określić interesujące użytkownika podsumowanie poprzez zapytanie (por. [4]-[12]) – wskazując obiekt (wartość rozmytą),
- można ograniczyć liczbę atrybutów, które są dla użytkownika interesujące,
- można określić więcej współczynników jakości i ich dopuszczalne zakresy,
- można wykorzystać jakąś metodę ograniczonego przeglądu, np. algorytmy genetyczne.

Użytkownik może zdefiniować zapytanie związane z wartością jednego z atrybutów. Może zapytać np. o młodych pracowników. Odpowiedzi (zdania podsumowujące) będą dotyczyć więc młodych pracowników. Zapytanie w_g jest zbiorem rozmytym w X_g związanym z atrybutem V_g.

Zapytanie można zaimplementować na dwa sposoby:

- możemy ograniczyć sprawdzane rekordy tylko do tych, dla których funkcja przynależności zapytania przyjmuje wartości większe od przyjętej stałej (progu),
- możemy zmienić wzór na T (2), tak aby zależał on od wartości funkcji przynależności zapytania, np.:

$$S(d_i)= \min_{j=1,\ldots,m}(s_j(V_j(y_i)))* w_g(V_g(y_i))$$

$$r = \frac{\sum_{i=1}^{n} S(d_i)}{\sum_{i=1}^{n} w_g(V_g(y_i))} \qquad (4)$$

Dotychczas ograniczaliśmy się do jednego współczynnika jakości podsumowania (zdania). Nie wskazywał on jednak na jego inne cechy, jak np. trafność, ogólność, czy stopień skomplikowania. Dlatego przy wartościowaniu zdań warto posłużyć się także innymi wskaźnikami. Korzystamy tu z propozycji Traczyka [13].

Takimi dodatkowymi wskaźnikami jakości podsumowań mogą być:

- wskaźnik rozmytości, którego zakres może być założony przez użytkownika, jego wartość może być niezależna od bazy danych, a tylko od sformułowania zdania,
- wskaźnik liczności określający, jaka część obiektów spełnia zapytanie,
- wskaźnik trafności mówiący, jak (do jakiego stopnia) charakterystyczną zależność określa dane zdanie dla wybranej bazy,
- wskaźnik długości zdania, który może wskazywać na jego zrozumiałość i czytelność.

Załóżmy więc, że $S(d_i)$ jest wartością funkcji przynależności dla i – tego rekordu, daną jako

$$S(d_i)= \min(s_j(V_j(y_i))) \quad j=1 \ldots m$$

i niech

$$r = \frac{\sum_{i=1}^{n} S(d_i) * w_g(V_g(y_i))}{\sum_{i=1}^{n} w_g(V_g(y_i))}$$

Wtedy te powyższe wskaźniki jakości są określone jako:

- T_1, pierwotny wskaźnik jakości Yagera (2), o którym zakłada się, że powinien dominować, określony jako:

$$T_1 = Q(r) \tag{5}$$

- T_2, wskaźnik rozmytości (por. Kacprzyk [3]), określony jako:

najpierw dla s_i;

$$in(s_j) = \frac{\left|\{x \in X_j : s_j(x) > 0\}\right|}{\left|X_j\right|} \tag{6}$$

a potem:

$$T_2 = 1 - \sqrt[m]{\prod_{j=1-m} in(s_j)} \tag{7}$$

- T_3, wskaźnik liczności, określony jako:

$$T_3 = \frac{\sum_{i:S(d_i)>0 \wedge w_g(V_g(y_i))>0} 1}{\sum_{i:w_g(V_g(y_i))>0} 1} \tag{8}$$

- T_4, wskaźnik trafności; określony jako:

najpierw dzielimy podsumowanie S na m podsumowań $S_1,S_2,...S_m$, gdzie każde związane jest tylko z jedną wartością rozmytą, $S_1=\{s_1\}$, $S_2=\{s_2\}$, ...,$S_m=\{s_m\}$, wtedy:

$$S_j(d_i)=s_j(V_j(y_i))$$

$$r_j = \frac{\sum_{s_j(d_i)>0} 1}{n}$$

w końcu:

$$T_4 = abs(\prod_{i=1..m} (r_j) - T_3) \tag{9}$$

- T_5 – wskaźnik długości podsumowania, określony jako:

$$T_5 = (0.5|S|*2 = \frac{2}{2^m} \tag{10}$$

4. Implementacja algorytmu – opis programu

Opisany powyżej algorytm został zaimplementowany w języku Delphi, w wersji 1.0. Opracowane zostały dwa programy. Pierwszy sprawdzał zachowanie się uproszczonej wersji algorytmu dla bazy z pracownikami. Przetestowane zostały algorytmy genetyczne. Drugi program, będący rozwinięciem pierwszego, umożliwia analizę dowolnej bazy danych sprowadzonej do pojedynczego pliku w formacie dbf.

Struktura analizowanej bazy danych jest następująca:

Nazwa atrybutu	Rodzaj atrybutu	Opis
Data	Data	Data sprzedaży
Czas	Czas	Czas sprzedaży
Nazwa	Napis	Nazwa towaru
Ilość	Numeryczny	Ilość sprzedanych sztuk w danej transakcji
Cena	Numeryczny	Cena jednostkowa
Marża	Numeryczny	Marża w procentach
Wartość	Numeryczny	Wartość transakcji = ilość x cena
Rabat	Numeryczny	Wartość rabatu w procentach
Grupa	Napis	Grupa z której pochodzi towar
Wartość dokumentu	Numeryczny	Wartość całej transakcji w dokumencie
Obrót z klientem	Numeryczny	Wartość całkowitej sprzedaży dla klienta (występującego w danej transakcji) w danym roku podatkowym.
Częstość klienta	Numeryczny	Liczba zakupów dokonana przez klienta w danym roku kalendarzowym
Miasto	Napis	Miasto z którego pochodzi klient

Tabela 1. Struktura analizowanej bazy danych

Użytkownik definiuje atrybuty i wartości rozmyte opierając się na pierwotnych atrybutach typu numerycznego, zawierających napisy lub datę i czas.

Program został przetestowany i sprawdzony przy użyciu baz danych z transakcjami sprzedaży. Moduł importu pobiera dane z jednego z programów magazynowych (PRO) i sprowadza je do jednego pliku w formacie dbf. Umożliwia to sprawdzenie zachowania się algorytmów na wielu realnych bazach danych związanych ze sprzedażą rzeczywistych firm handlowych.

Baza zawiera informacje, które zebrane zostały poprzez bezpośrednie przepisanie danych, ale także w wyniku obliczeń wykonanych na zbiorze baz powiązanych ze sobą relacjami. Wymaga to zwykle przygotowania funkcji eksportującej dane korzystającej z pewnej wiedzy o strukturze i relacjach występującej w zbiorze baz źródłowych. Program nie ogranicza ilości analizowanych atrybutów, wielkości bazy ani wartości atrybutów. Użytkownik może przeglądać, uzupełniać i poprawiać bazę danych.

Wprowadzone zostały okna umożliwiające definicję atrybutów i wartości rozmytych dla konkretnej bazy danych, sprawdzenie dopasowania wartości rozmytych do elementów opisanych przez rekordy z relacji oraz tworzenie zapytań do bazy w oparciu o zbudowany język.

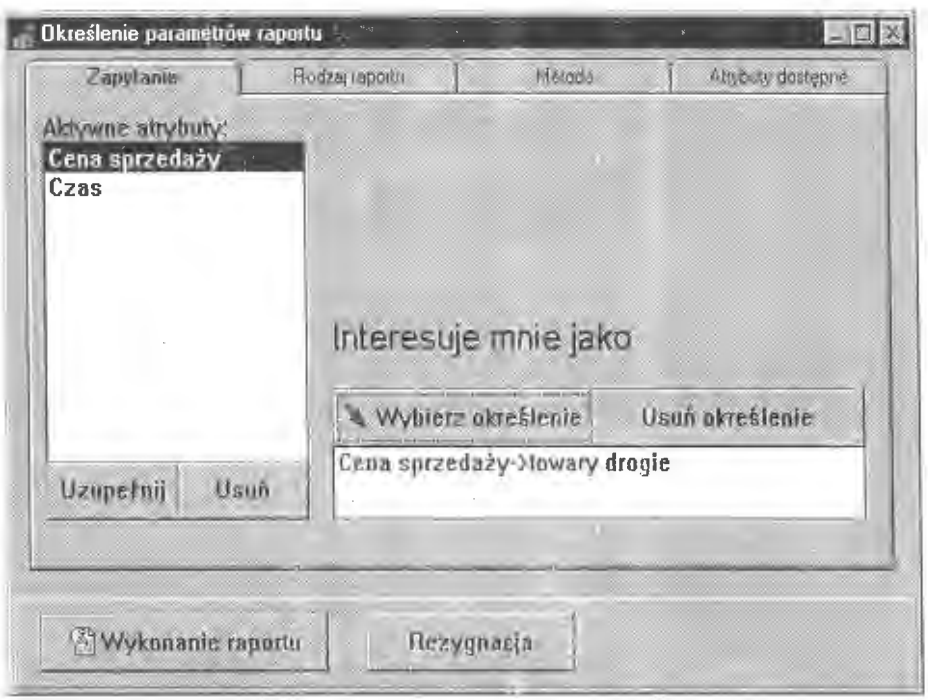
Po zdefiniowaniu atrybutów i wartości rozmytych, wskazaniu bazy danych i przetestowaniu zapytań do baz, można przejść do najważniejszej części programu, czyli modułu generowania podsumowań. Moduł ten umożliwia stworzenie zbioru zdań najlepiej pasujących do bazy danych.

Po uruchomieniu opcji „Analiza - Wykonanie podsumowania" pojawia się okno z stronicowaną konfiguracją oraz dwoma przyciskami, jak to pokazano na rys. 1.

Pierwszym krokiem jest określenie parametrów podsumowania, następnie uruchamia się procedurę liczącą, a ostatecznie na ekranie pojawia się okno wyników, które możemy przeglądać na różne sposoby.

Parametry generowania podsumowań są podzielone na trzy grupy:

- „Zapytanie" – określenie atrybutów i podmiotu,
- „Rodzaj raportu" – określenie sposobu przedstawienia wyników,
- „Metoda" – ustalenie parametrów metody.



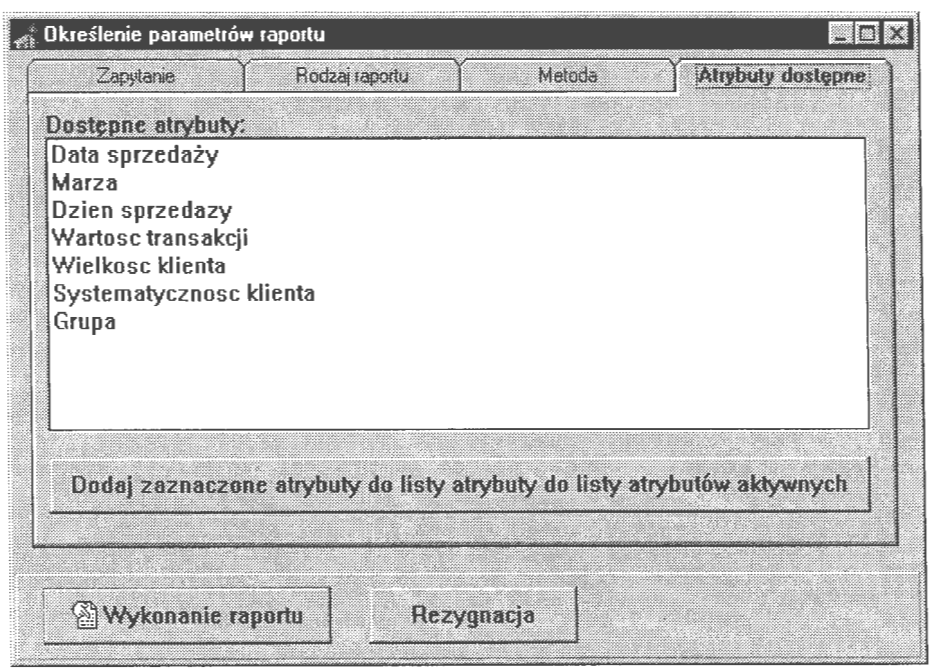
Rys. 1. Okno wykonania podsumowania

Grupa parametrów „Zapytanie" pozwala ustalić jakie atrybuty mają występować w zdaniach oraz jaki ma być podmiot. Naciśnięcie klawisza „Uzupełnij" powoduje przejście do strony z dostępnymi atrybutami. Pojawia się lista atrybutów, które zaznaczamy i wybieramy klawiszem „Dodaj zaznaczone...".

Wybranie atrybutów jest równoważne z zadaniem zapytania typu:

„co można powiedzieć o zależnościach między atrybutami ..."

Na przykład, jeśli wybierzemy atrybuty „marża" oraz „wielkość klienta", to otrzymamy informacje typu, jakie marże otrzymuje mały, średni i duży klient, a także, jaka była sprzedaż dla dużych klientów, lub jaka część sprzedaży jest z niską marżą. O czasie obliczeń decyduje przede wszystkim liczba aktywnych atrybutów (patrz rys. 2). Rozsądne jest analizowanie równocześnie najwyżej 8 atrybutów, z tym, że szybko otrzymamy wyniki tylko dla dwóch lub trzech.



Rys. 2. Okno wyboru atrybutów aktywnych

W oknie przedstawionym na rys. 2 możemy jeszcze określić podmiot, jaki dopuszczamy w podsumowaniach. Domyślnym podmiotem jest podmiot ogólny, jak np. sprzedaż. Ale możemy narzucić podmiot i stwierdzić, że np. interesują nas zależności dotyczące tylko sprzedaży towarów drogich.

W ten sposób bardzo łatwo możemy oczekiwać odpowiedzi, które odpowiadają na pytania:

- jakiego typu towary kupowali systematyczni klienci,
- jakie grupy towarów występowały na dużych fakturach.,
- kto kupuje po godzinie 16,
- jakie są zależności między marżą, a grupą towarową,
- jakie godziny sprzedaży charakteryzują kolejne pory roku.

Kolejną grupą parametrów jest „Rodzaj raportu". Tutaj ustalamy, jak długie ma być podsumowanie, jakie współczynniki mają być wypisane oraz rodzaj podmiotu (rys. 3).

Długość podsumowania (z ilu zdań się składa) wpływa na przejrzystość wyniku, ale także na czas obliczeń. Czym dłuższego podsumowania szukamy tym dłuższy czas oczekiwania na wynik (czas wydłuża się o parę procent).

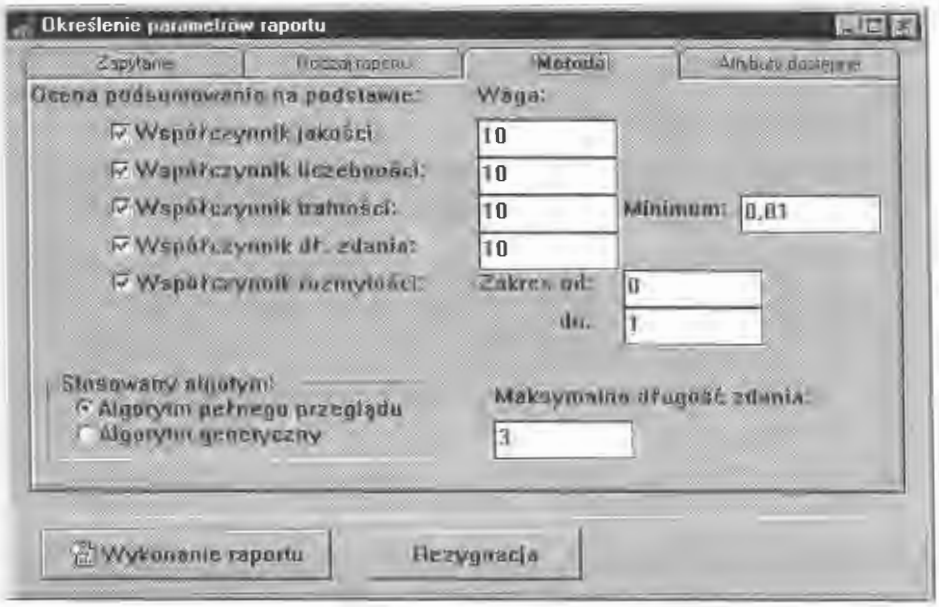
Bardzo duży natomiast jest wpływ rodzaju podmiotu na czas obliczeń. Wybranie podmiotu ogólnego powoduje, że komputer podczas sprawdzania wszystkich dopuszczalnych zdań ustala podmiot na stałe, co radykalnie zmniejsza liczbę sprawdzanych przykładów zwykle 7-8 razy.

Ostatnią grupą parametrów jest „Metoda", w której ustalamy parametry generowania podsumowań (rys. 4). Wpływają one na sens, wygląd i kolejność generowanych zdań.

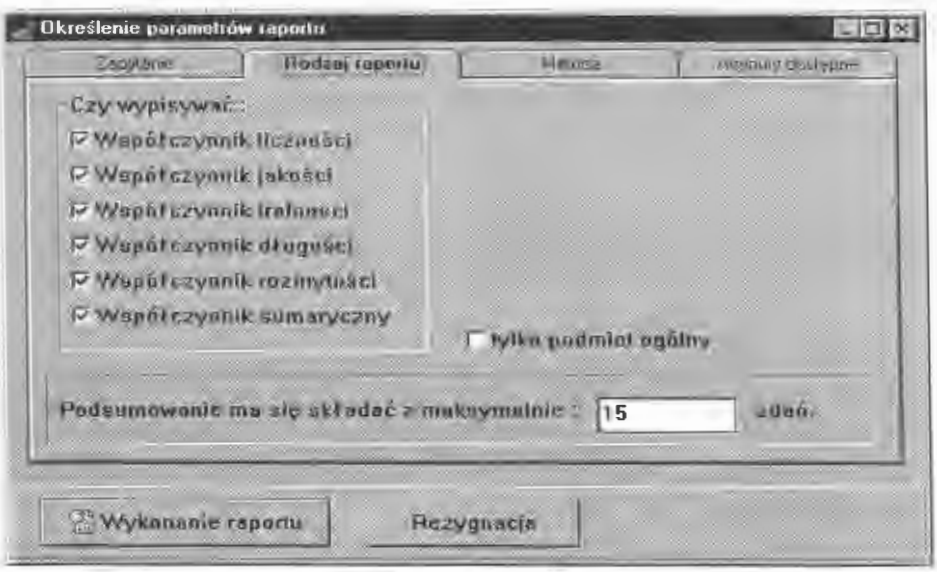
Ostateczna ocena jakości danego zdania jest średnią ważoną wybranych wskaźników, z możliwością odrzucenia zdań po niespełnieniu warunków związanych z zakresem ich wartości.

Możemy wybrać algorytm pełnego przeglądu lub algorytm genetyczny. Pierwszy znajdzie rozwiązania optymalne, ale będzie trwał dłużej. Na czas obliczeń duży wpływ ma maksymalna długość zdania. Wskazuje ona na liczbę określeń i powinna być ograniczona, ponieważ czym więcej, określeń tym dłuższy czas obliczeń oraz mniejsza czytelność zdań.

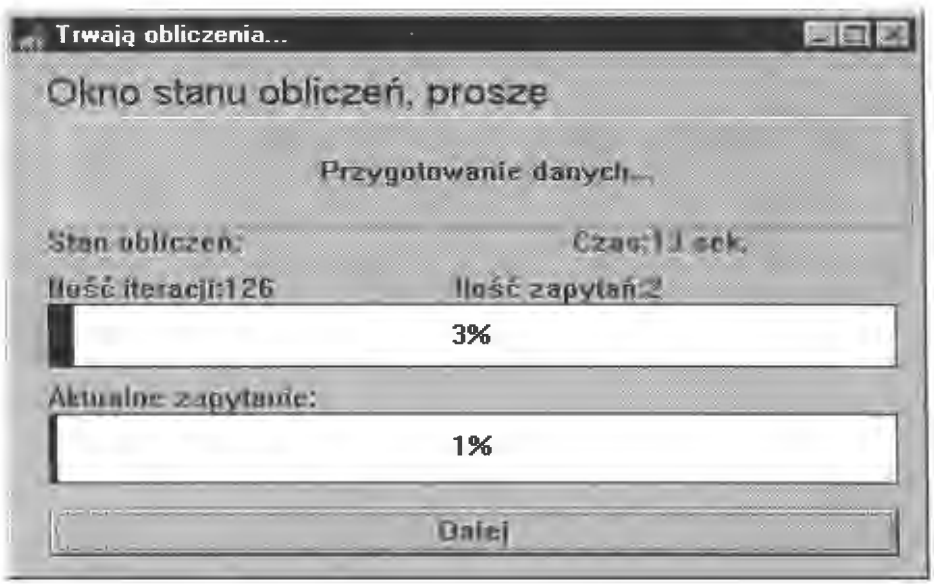
Po ustaleniu wszystkich parametrów należy nacisnąć klawisz „Wykonanie raportu". Powoduje to otwarcie okna obliczeń pokazanego na rys. 5.



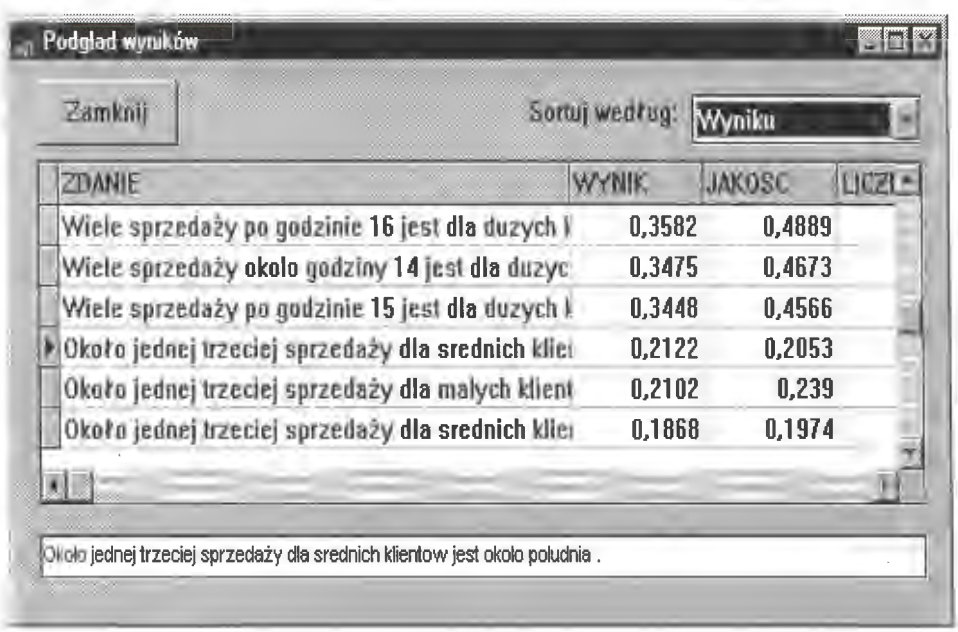
Rys. 3. Okno określające wygląd raportu



Rys. 4. Okno określające parametry generowania podsumowania



Rys. 5. Okno zaawansowania obliczeń



Rys. 6. Okno z wynikami

Górna linia wskazuje na postęp w przeglądzie zapytań, dolna na postęp w zapytaniu aktualnym. Dodatkowo wyświetlone są informacje o liczby zdań do sprawdzenia oraz czasie obliczeń. Po zakończeniu obliczeń przechodzimy do okna przeglądu wyników.

Mamy możliwość przeglądu wszystkich wygenerowanych zdań oraz przypisanych im wskaźników. Zdania możemy przeglądać według wartości wybranego wskaźnika lub według globalnej oceny (domyślnie) – por. rys. 6.

W zależności od wielu parametrów tematyka zdań, które występują w podsumowaniu dotyczyć może różnych atrybutów. Zwykle analizie poddaje się dwa lub trzy atrybuty, otrzymując najlepsze zdania opisujące zależności pomiędzy atrybutami. Dla bazy danych zawierającej informacje o sprzedaży firmy handlowo – usługowej za rok 1997 otrzymano między innymi następujące zdania pokazane w tab. 2.

Zdanie oraz wartość rozmyta	Współczynnik trafności	Współczynnik liczebności	Współczynnik jakości
	Współczynnik rozmytości	Wynik	
Około połowy sprzedaży ze strata jest dla klientów stałych.	0.2136 0.7520	0.5385 0.3851	0.2136
Około jednej trzeciej sprzedaży w bardzo małych porcjach jest z wysoka marżą.	0.2279 0.1872	0.4151 0.3085	0.3411
Mało sprzedaży ze strata jest dla klientów jednorazowych.	0.1090 0.0641	0.1731 0.1763	0.1090

Około połowy sprzedaży elementów sieci jest z wysoka marzą.	0.2329 0.1872	0.4202 0.3165	0.3630
Około połowy sprzedaży komputerów jest z średnia marzą.	0.2045 0.3453	0.5498 0.3699	0.4753
Wiele sprzedaży z wysoka marzą jest akcesorii.	0.1684 0.4095	0.5779 0.3919	0.5713
Około połowy sprzedaży z bardzo dużych faktur jest pod koniec tygodnia.	0,0412 0,4118	0,4529 0,0828	0,3320
Około jednej trzeciej sprzedaży z bardzo dużych faktur jest po godzinie 15.	0,0166 0,2559	0,2725 0,0452	0,1266
Około jednej trzeciej sprzedaży komputerów jest w końcu roku.	0,0999 0,2010	0,3009 0,1274	0,2801
Około połowy sprzedaży w okresie jesiennym jest akcesorii.	0,0642 0,4095	0,4737 0,1143	0,4790
Około jednej trzeciej sprzedaży elementów sieci jest na początku roku.	0,0733 0,2124	0,2857 0,0982	0,1957
Około jednej trzeciej sprzedaży po godzinie 15 jest w okresie wiosennym dla dużych klientów.	0.0358 0.2186	0.2544 0.0528	0.1514

Bardzo mało sprzedaży po godzinie 15 jest w końcu roku.	0.0512 0.2010	0.1498 0.0750	0.1382
Wiele sprzedaży w sobotę jest dla dużych klientów.	0,0094 0,8656	0,8750 0,0980	0,8319
Około połowy sprzedaży w godzinach rannych jest pod koniec tygodnia.	0,0472 0,4118	0,4590 0,0985	0,3588
Około jednej trzeciej sprzedaży w godzinach rannych jest w okresie letnim.	0,0516 0,2114	0,2630 0,0857	0,1845
Około jednej trzeciej sprzedaży dla średnich klientów jest z niska marżą.	0,3186 0,5837	0,2650 0,3013	0,2158
Około połowy sprzedaży dla małych klientów jest z wysoka marżą.	0,2484 0,1872	0,4356 0,2716	0,3611
Wiele sprzedaży w sobotę jest około południa z niska marża.	0,3843 0,2748	0,6591 0,3863	0,3951
Wiele sprzedaży dla małych klientów jest dla klientów przypadkowych.	0,6250 0,1458	0,7709 0,5986	0,5105

Tabela 2. Wybrane współczynniki i zdania określające zależności pomiędzy atrybutami z bazy sprzedaży

5. Wnioski

Lingwistyczne podsumowania pozwalają na szybkie zdobycie trafnych i ogólnych informacji o obiektach opisywanych w bazie danych. Podana metoda nie jest ograniczona przez pochodzenie informacji, rodzaj obiektu oraz wielkość bazy danych.

Przebieg analizy bazy danych pozwala na otrzymanie zdań dotyczących wybranej grupy obiektów lub wszystkich. Istnieje możliwość stworzenia zapytania w języku naturalnym ograniczającego atrybuty, poziom ogólności poszukiwanych zdań oraz czas oczekiwania. W wyniku działania algorytmu pojawia się grupa zdań oznajmujących w języku naturalnym, które są krótkie i przez to łatwiej zrozumiałe.

Zaproponowana metoda została sprawdzona na specjalnie opracowanym oprogramowaniu „GP" napisanego przez autora pracy. Oprogramowanie to zostało sprawdzone na danych fikcyjnych dotyczących pracowników oraz danych pochodzących z rzeczywistej firmy usługowo – handlowej i dotyczących sprzedaży w roku podatkowym 1997.

Istnieje wiele kierunków rozwoju omawianej metody. Dotyczą one rozszerzenia metody oraz samej implementacji algorytmu. Rozbudowanie algorytmu tak, aby wynikiem analizy mogły być zdania złożone oraz łączenie zdań w logiczną całość tworzącą zwarte streszczenie, z pewnością zwiększyłyby atrakcyjność omawianej metody. Przyspieszenie algorytmu poszukiwań, dodanie bardziej jasnych i łatwych do analizy współczynników oraz poprawienie interfejsu programu komputerowego podniosłoby użyteczność programu.

Literatura

[1] Bosc P. and Kacprzyk J., eds.. (1995) Fuzziness in Database Management Systems. Physica-Verlag, Heidelberg and New York.

[2] George R. and Srikanth R. (1996) Data summarization using genetic algorithms and fuzzy logic. In: F. Herrera and J.L. Verdegay,

eds.:Genetic Algorithms and Soft Computing. Physica-Verlag, Heidelberg and New York, 599-611.

[3] Kacprzyk J. (1986) Zbiory rozmyte w analizie systemowej, PWN, Warszawa.

[4] Kacprzyk J., Zadrożny S. and Ziółkowski A. (1989) FQUERY III+: a 'human-consistent` database querying system based on fuzzy logic with linguistic quantifiers. Inf. Systems 6, 443-453.

[5] Kacprzyk J. and Zadrożny S. (1994a) Fuzzy querying for Microsoft Access, Proc. of FUZZ - IEEE'94 (Orlando, USA), 1994, Vol. 1, 167-171.

[6] Kacprzyk J. and Zadrożny S. (1995a) Fuzzy queries in Microsoft Access v. 2 , Proc. of FUZZ-IEEE/IFES '95 (Yokohama), Workshop on Fuzzy Database Systems and Information Retrieval, 61-66.

[7] Kacprzyk J. and Zadrożny S. (1995b) Fuzzy queries in Microsoft Access v.2, Proc. of 6th IFSA Congress (Sao Paolo), Vol. II,. 341-344.

[8] Kacprzyk J. and Zadrożny S. (1995c) FQUERY for Access: fuzzy querying for a Windows-based DBMS. In: P. Bosc and J. Kacprzyk, eds.: Fuzziness in Database Management Systems. Physica-Verlag: Heidelberg,. 415-433.

[9] Kacprzyk J. and Zadrożny S. (1996) A fuzzy querying interface for a WWW-server-based relational DBMS. Proc. of 6th IPMU Conference (Granada), Vol. 1, 19-24.

[10] Kacprzyk J. and Zadrozny S. (1998a) On sumarization of large datasets via a fuzzy-logic-based querying add-on to Microsoft Access. In: Intelligent Information Systems VII. Proceedings of Workshop held in Malbork, IPI PAN, Warszawa, 249-258.

[11] Kacprzyk J. and Zadrożny S. (1998b) Data Mining via Linguistic Summaries of Data: An Interactive Approach. W: T. Yamakawa i G. Matsumoto, eds.: Methodologies for the Conception, Design and Application of Soft Computing, Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems – IIZUKA'98, World Scientific, Singapore, 668-671.

[12] Zemankova M. and Kacprzyk J., eds. (1993) Integrating Artificial Intelligence and Database Technologies, Special Issue of *Journal of Intelligent Information Systems*, Vol. 2, No 4.

[13] Traczyk W. (1997) Evaluation of Knowledge Quality, *System Science*, vol.23, 201-225.

[14] Yager R.R. (1991) On linguistic summaries of data. W: W. Frawley and G. Piatetsky-Shapiro, eds.: Knowledge Discovery in Databases. AAAI/MIT Press, 347-363.

[15] Yager R.R. and Kacprzyk J. (1997) The Ordered Weighted Averaging Operators: Theory and Applications. Kluwer, Boston.

[16] Zadeh L.A. (1983) A computational approach to fuzzy quantifiers in natural languages. *Computers and Maths with Appls*. 9, 149-184.

[17] Zadeh L.A. and J. Kacprzyk, eds. (1992) Fuzzy Logic for the Management of Uncertainty, Wiley, New York.

[18] Zadeh L.A. and J. Kacprzyk, eds. (1999) Computing with Words in Information/Intelligent Systems. Physica-Verlag, Heidelberg and New York.

WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ I ZARZĄDZANIA

działa pod auspicjami
Polskiej Akademii Nauk

ZAŁOŻYCIELEM
Wyższej Szkoły Informatyki Stosowanej i Zarządzania
jest
FUNDACJA KRZEWIENIA NAUK SYSTEMOWYCH
powołana z inicjatywy
Prezesa
POLSKIEJ AKADEMII NAUK

FUNDATOREM
Fundacji Krzewienia Nauk Systemowych
jest
POLSKA AKADEMIA NAUK

ORGANEM
sprawującym nadzór
jest
MINISTERSTWO EDUKACJI NARODOWEJ

Wyższa Szkoła Informatyki Stosowanej i Zarządzania
prowadzi studia wyższe na kierunkach:
INFORMATYKA
ZARZĄDZANIE I MARKETING

SIEDZIBA
Instytut Badań Systemowych
Polskiej Akademii Nauk

ISBN 83-85847-24-3