TECHNIKI POZYSKIWANIA WIEDZY W HURTOWNI DANYCH

Małgorzata NYCZ

Streszczenie: Artykuł poświęcony jest pozyskiwaniu wiedzy z hurtowni danych. Składa się z czterech części. Po krótkim wstępie zaprezentowano techniki przetwarzania struktur danych w ramach hurtowni danych. Następnie przedstawiono techniki pozyskiwania wiedzy z danych zebranych w hurtowni. Artykuł kończy podsumowanie przeprowadzonych rozważań.

Słowa kluczowe: hurtownia danych, techniki odkrywania wiedzy.

1. Wprowadzenie

Żyjemy w czasach, które charakteryzują się m.in. tym, że ilość informacji wzrasta lawinowo. Przedsiębiorstwa stają wobec problemu posiadania, przetwarzania i zbierania coraz większych ilości danych. Dane pochodzą najczęściej z heterogenicznych źródeł. Problem integracji danych rozwiązuje wykorzystanie np. hurtowni danych. Przetwarzanie struktur danych zgromadzonych w hurtowni danych , jak też zastosowanie data mining może dostarczyć decydentowi odpowiedzi na zapytania biznesowe; odpowiedzi, które mu pomogą w podjęciu decyzji biznesowej. Artykuł przedstawia techniki przetwarzania struktur danych w ramach hurtowni, a następnie prezentuje techniki data mining służące do odkrywania wiedzy z danych zawartych w hurtowni danych.

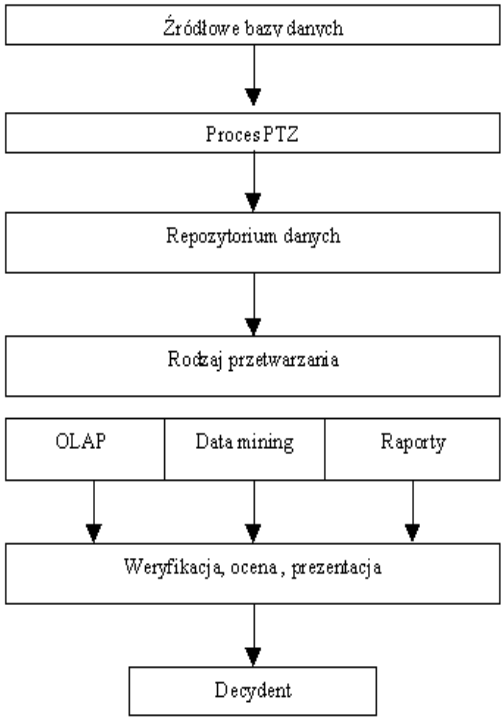
2. Techniki przetwarzania struktur danych w hurtowni danych

Dane w hurtowni danych mogą być, generalnie rzecz ujmując, wykorzystywane w ramach trzech rodzajów ich przetwarzania. Są nimi wielowymiarowa analiza danych, data mining oraz raportowanie. Wyniki są weryfikowane oceniane, i prezentowane jako raport w postaci wygodnej dla użytkownika. Schematycznie operacje wykonywane na danych zgromadzonych w HD przedstawia rysunek 1.

Podstawowych (ale nie jedynym: innym, często spotykanym jest klasyczne, relacyjne przetwarzanie danych) rodzajem przetwarzania danych w hurtowni danych jest przetwarzanie analityczne OLAP (ang. On Line Analytical Processing), a wyniki mogą być prezentowane w postaci raportów o, na przykład trendach, osiągnięciach lub niepowodzeniach konkretnych strategii marketingowych. W ramach operacji na danych zgromadzonych w HD mogą być wykonywane operacje związane z odkrywaniem wiedzy (data mining), o czym w dalszej części rozdziału.

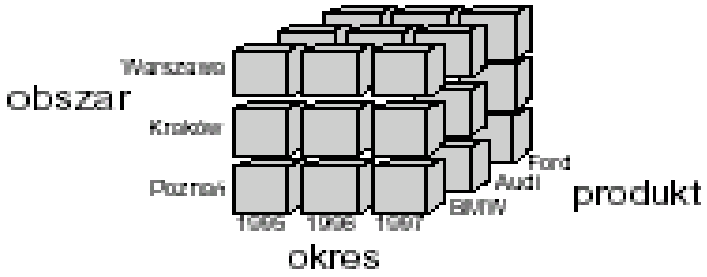
OLAP składa się z trzech podstawowych elementów, którymi są:

1) struktura danych opisująca logiczną organizację danych oraz sposób postrzegania danych przez użytkowników,
2) zbiór operatorów umożliwiających wyszukiwanie i modyfikowanie danych,
3) ograniczenia integralnościowe, specyfikujące poprawność danych.



Rys. 1. Rodzaje przetwarzania danych w hurtowni danych
Źródło: opracowanie własne

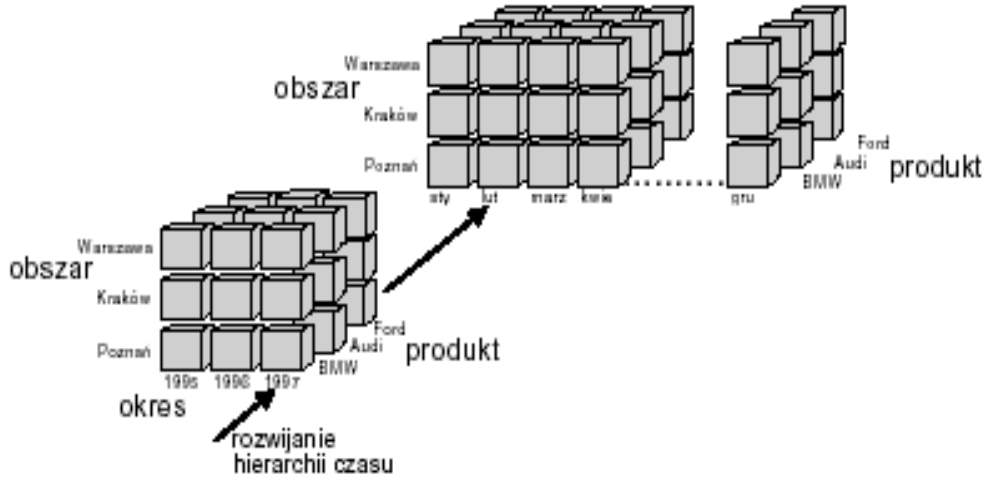
W OLAP dane postrzegane są przez użytkownika w postaci wielowymiarowej perspektywy, przedstawianej poglądowo jako kostki OLAP. Obiektem analizy jest zbiór miar numerycznych, które nazywane są faktami. Fakt opisuje pojedyncze zdarzenie i jest daną ilościową (numeryczną) reprezentującą aktywność biznesową, na przykład sprzedaż produktów, średni zysk, wartość produktu krajowego, etc. Wartość miary zależy od zbioru wymiarów, który z kolei określa kontekst miary. I tak na przykład sprzedaż produktów można rozpatrywać w kontekście miasta, dostawców, klientów, okresów sprzedaży czy konkretnego produktu. Miara jest przedstawiana jako punkt w wielowymiarowej przestrzeni wymiarów. Z każdym wymiarem związany jest zbiór atrybutów. Na przykład wymiar klient może mieć takie atrybuty, jak identyfikator klienta, jego nazwa, adres, NIP, telefon, fax, etc.; wymiar produkt może opisy przez takie atrybuty, jak obszar, okres, produkt. Dane wymiarów są zdenormalizowane, co pozwala użytkownikowi na wygodne eksplorowanie „w dół" (ang. *drill-down*), „w górę" - agregowanie (ang. *drill-up*) oraz „w poprzek" (ang. *drill across*). Dane referencyjne mogą zawierać tysiące wierszy, ale w porównaniu z nimi dane faktów - miliony. Dane wymiarów nie zmieniają się tak często, jak dane faktów czy dane zagregowane. Atrybuty wymiaru mogą tworzyć hierarchię atrybutu [10, 12]. Na rysunku 2. przedstawiono przykładową kostkę trójwymiarową z wymiarami obszar, produkt i okres dla faktu wielkość sprzedaży samochodów w Polsce.



Rys. 2. Kostka trójwymiarowa OLAP [7]

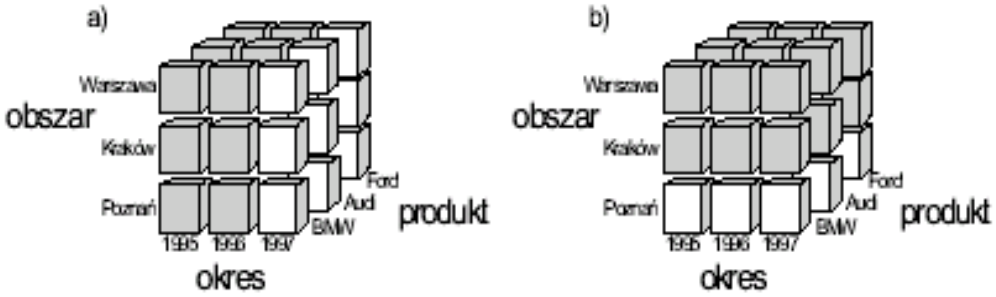
Operacje wykonywane na strukturach danych hurtowni określane są za pomocą operatorów. Zatem zbiór operacji (operatorów) określa operacje przetwarzania struktur danych. Do podstawowych zalicza się takie, jak:

- wyznaczanie punktu centralnego (ang. *pivoting*) mające za zadanie wskazanie miary, która jest interesująca dla użytkownika i wybranie dwóch wymiarów, w których ma być ona reprezentowana. Na przykład w wymiarze produktu reprezentującego samochodów Fabia i wymiarze dealerzy może być prezentowana liczba sprzedaży samochodów,
- rozwijanie wymiaru (ang. *drill – down*) oznaczające rozwijanie agregatu na części składowe, na przykład sprzedaż w poszczególnych branżach, sprzedaż w poszczególnych kategoriach samochodów, poszczególnych okresach. Jako przykład niech będą dane informacje o sprzedaży samochodów różnych marek, w latach 1995, 1996, 1997, w poszczególnych miastach. Aby sporządzenia analizy sprzedaży w poszczególnych miesiącach w roku 1997, rozwija się hierarchię okres reprezentującą rok 1997. Analiza sprzedaży w poszczególnych dniach danego miesiąca jest możliwa po rozwinięciu hierarchii reprezentującej dany miesiąc,



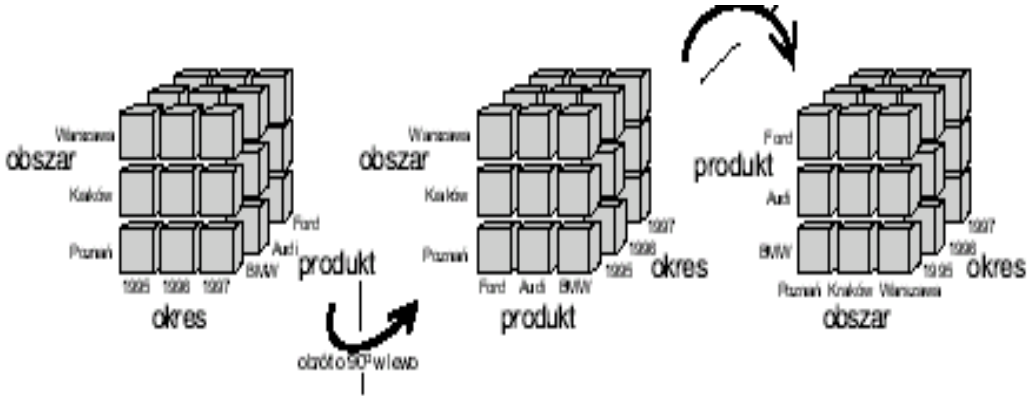
Rys. 3. Operacja rozwijania hierarchii wymiaru [7]

- zwijanie (ang. *roll – up*) - operacja oznaczająca dla danego wymiaru zwijanie w górę hierarchii wymiaru w celu prezentacji większych agregatów,
- wycinanie (ang. *slice and dice*) – odpowiada operacji redukcji liczby wymiarów, a więc zachodzi projekcja danych na wybranym podzbiorze wymiarów dla wybranych wartości innych wymiarów,



Rys. 4. Operacje wycinania danych w różnych wymiarach [7]

- obracanie – umożliwiające prezentowanie danych w różnych układach,



Rys. 5. Operacje obracania danych [7]

- rankowanie (ang. *ranking*) umożliwiające wybór pierwszych n elementów, na przykład trzy najlepiej sprzedające się produkty w miesiącu styczniu,
- inne operacje, jak selekcja, procedury składowane, etc.

Zbiór ograniczeń integralnościowych określa poprawność przechowywanych danych. Ograniczenia integralnościowe dla kostki wielowymiarowej można podzielić na dwie grupy, jakimi są:

- ograniczenia integralnościowe pojedynczej kostki danych (ang. *intra cube constrains*), związane z definicjami zależności między atrybutami wymiarów, wymiarami a miarami, oraz hierarchiami wymiarów,
- ograniczenia integralnościowe pomiędzy kostkami danych (ang, *inter cube constrains*) określające związki pomiędzy dwoma bądź więcej kostkami danych, jak zależności między miarami kostek, wymiarami kostek, miarą jednej kostki a wymiarami innych kostek.

Wynikiem wykonanych na danych operacji wielowymiarowych analiz są informacje, które przedstawione w postaci raportu – być może okażą się wystarczające dla menedżera w procesie decyzyjnym. Ale menedżer może zażyczyć sobie wykonania dalszych czynności, związanych z poszukiwaniem wśród tych informacji wiedzy. Wówczas uruchamiany jest data mining, czyli „uzupełnieniem" technologii OLAP są w takiej sytuacji techniki data mining, których zadaniem jest między innymi odkrywanie wzorców i trendów w danych, trudnych do odkrycia innymi metodami.

3. Techniki data mining

Data mining może zostać uruchomiony w dwóch sytuacjach: albo jako samodzielne zadanie zlecone przez użytkownika zupełnie abstrahując od hurtowni danych, realizowane na przygotowanym uprzednio zbiorze danych, albo jako kolejny krok po zakończeniu realizacji wielowymiarowych analiz w hurtowni danych, na zestawie wygenerowanych informacji. W obu przypadkach uzyskujemy „na wyjściu" użyteczną wiedzę. Z punktu widzenia data mining nie ma znaczenia, czy wystąpił pierwszy czy drugi przypadek, albowiem sposób realizacji technik data mining jest taki sam.

Istnieje wiele różnych technik data mining. Można je ująć na przykład następująco (por. [2], [3], [9]):

- odkrywanie zależności (ang. *mining association rules*),
- wielopoziomowe uogólnianie danych (ang. *multi-level data generalization*),
- klasyfikacja (ang. *data classification*),
- grupowanie (ang. *clustering analysis*),
- odkrywanie podobieństw w oparciu o wzorce (ang. *pattern similarity search*),
- odkrywanie schematów ścieżek (ang. *mining path traversal patterns*).

Odkrywanie zależności w przypadku bazy danych o transakcjach przeprowadzonych w sklepie będzie polegać na identyfikacji artykułów nabywanych razem (na przykład mleko i chleb). Jeśli $A = \{a_1, a_2, ..., a_n\}$ będzie zbiorem elementów reprezentujących artykuły w sklepie, a $T = \{T_1, T_2, ..., T_n\}$ będzie zbiorem transakcji reprezentującym fakt zakupienia dwóch artykułów, to oznacza, że $T_i \subset A$. Jeśli założymy, że $X \subset A$, wówczas o transakcji T_i możemy powiedzieć, że zawiera zbiór X wtedy i tylko wtedy, gdy $X \in T_i$. Zależność tę możemy przedstawić w formie implikacji $X \Rightarrow Y$, gdzie $X \subset A$, $Y \subset A$, a $X \cap Y = \varnothing$. O zależności $X \Rightarrow Y$ można powiedzieć, że posiada wiarygodność c (gdy c% transakcji ze zbioru T zawierających podzbiór X - zawiera również podzbiór Y) określającą siłę zależności. Jeśli s% transakcji ze zbioru T zawiera podzbiór X lub Y, wówczas o zależności $X \Rightarrow Y$ można powiedzieć, że ma wsparcie o wartości s, informujące nas o częstości pojawiania się zależności w bazie danych. Główne zadanie algorytmu data mining to znalezienie silnych zależności, które charakteryzuje duża wiarygodność i silne wsparcie, a więc zidentyfikowanie największych zbiorów elementów w bazie o wsparciu powyżej wyznaczonej granicy i wykorzystanie ich do wygenerowania poszukiwanych zależności. Ilustracją może być tutaj algorytm Apriori [1]. Algorytm ten konstruuje zestaw różnych, równolicznych podzbiorów, złożonych z elementów transakcji będących kandydatami na podzbiory o wystarczająco dużym wsparciu. W kolejnych iteracjach tworzone są podzbiory (jedno-, dwu-, trójelementowe itd.), aż do utworzenia odpowiednio licznych podzbiorów z wystarczająco dużym wsparciem lub do momentu, gdy nie można utworzyć podzbiorów w kolejnej iteracji. Dla każdego zbioru kandydującego oblicza się wsparcie, a następnie wybiera te, których wsparcie jest większe od założonego przez użytkownika.

Przyjmijmy, że D_k będzie zestawem podzbiorów kandydujących (po iteracji k), a L_k

zestawem podzbiorów z D_k posiadających wystarczające wsparcie. Zestaw D_{k+1} w kolejnej iteracji algorytmu Apiori jest następujący:

$$\{X \cup Y;\ X, Y \in L_{k\ ;}\ |X \cap Y| = k\text{-}1\}. \quad (1)$$

Każdy podzbiór z zestawu D_{k+1} musi spełniać warunek, że dowolny jego podzbiór (od 1 do k elementowy) posiada wystarczające wsparcie, czyli znajduje się w dowolnym zestawie od L_1 do L_k z poprzednich iteracji. Jako przykład możemy przyjąć bazę danych do analizy za pomocą algorytmu Apriori. Algorytm Apriori znajduje reguły asocjacyjne w relacyjnej bazie danych. Asocjacją pomiędzy danymi nazywamy zależność implikacyjną reprezentowaną za pomocą reguł logicznych połączonych zależnością – jeśli X to Y. Przykładem może być: jeśli garnitur to koszula, jeśli buty to torebka, jeśli płaszcz to apaszka. Przez wsparcie reguły w danym zbiorze rozumiemy stosunek liczby zbiorów zawierających daną regułę do liczby wszystkich zbiorów. Duża wartość wsparcia może zawierać informacje dotyczące strategii działania firmy. Można na tej podstawie ustalić rozmieszczenie towarów w sklepie. Jeśli towary po obniżonej cenie rozmieścimy pomiędzy produktami X i Y zachodzi duże prawdopodobieństwo, że zostaną zakupione pomimo podwyższenia ceny towaru Y.

Przykładem może być transakcja opisana za pomocą identyfikatora transakcji T_i (identyfikator klienta sklepu, data i godzina robienia zakupów) oraz produkty zakupione w ramach tej transakcji. W przypadku eksploracji danych w bazie danych supermarketu będzie to znalezienie prawidłowości w kolejności kupowania różnych produktów.

T_i – identyfikator transakcji	Produkty
1	P1 P3 P4
2	P2 P3 P5
3	P1 P2 P3 P5
4	P2 P5

D_1 jest zestawem następujących podzbiorów: {P1}, {P2}, {P3}, {P4}, {P5}. Dla każdego podzbioru obliczamy wsparcie i eliminujemy te, które mają najmniejsze wsparcie:

D_1 L_1

Podzbiory	Wsparcie
{P1}	2
{P2}	3
{P3}	3
{P4}	1
{P5}	3

Podzbiory	Wsparcie
{P1}	2
{P2}	3
{P3}	3
{P4}	3

W następnej iteracji bierzemy pod uwagę wszystkie dwuelementowe i postępujemy jak poprzednio. W L_2 otrzymujemy:

Podzbiory	Wsparcie
{P1 P3}	2
{P2 P3}	2
{P2 P5}	3
{P3 P5}	2

W trzeciej iteracji C_3 zawiera podzbiory {P2 P3 P5}, dla których wsparcie wynosi 2. W tej sytuacji algorytm Apriori kończy swoje działanie, gdyż nie jest już możliwe utworzenie dalszych iteracji.

Innym jest algorytm GSP (ang. *Generalized Sequential Patterns*) służący do odkrywania reguł sekwencyjnych w bazie danych. Podczas pracy algorytmu wykonuje się wiele odczytów baz danych, z których pierwszym zadaniem jest obliczenie wsparcia poszczególnych pozycji, będącego liczbą sekwencji zawierających te pozycje. Po pierwszym odczycie znane są jednoelementowe zbiory częste, a więc pozycje mające minimum wparcia. Następnie algorytm startuje ze zbiorami wzorców częstych znalezionych w poprzedniej fazie. Każdy następny wzorzec ma o jedną pozycję więcej niż ten, którym posłużyliśmy się do jego wygenerowania. Wsparcie dla tych wzorców oblicza się podczas jednego pełnego odczytu bazy danych, co pozwala stwierdzić, które wzorce można uznać potencjalnie za częste i wykorzystać do generacji w następnym kroku. Algorytm kończymy, gdy po kolejnym odczycie brak jest wzorców częstych lub nie udało się wygenerować nowych wzorców potencjalnie częstych. Ograniczenia czasowe związane z sekwencjami (dotyczące bazy danych z przykładu Apriori) to między innymi:

- minimalna odległość (min-gap) – minimalna różnica czasów wystąpień dwóch kolejnych sekwencji, pozwalająca na sprowadzenie pewnego produktu, którego wzrost jest bardzo prawdopodobny, gdyż z obserwacji wynika wzrost innego produktu,
- maksymalna odległość (max-gap) – maksymalna różnica czasów wystąpień dwóch kolejnych pozycji wzorca sekwencji, która pozwala na odrzucenie klientów, którzy rzadko robią zakupy w naszym sklepie.

Algorytmy służące do wielopoziomowego uogólniania danych oprócz cech odkrywania zależności mogą zawierać elementy ułatwiające przeprowadzanie analiz takie jak na przykład [HAN96]: generowanie zależności zbudowanych na różnych poziomach abstrakcji, definiowanie różnych minimalnych wartości wsparcia dla różnych poziomów hierarchii, warunkowe badanie zależności na niższym poziomie wówczas, gdy ta zależność posiada na wyższym poziomie odpowiednie wsparcie.

Celem technik klasyfikacji jest znalezienie wspólnych cech charakterystycznych wśród obiektów bazy danych i przyporządkowanie ich do odpowiednich klas (grup), które pozwolą odróżnić je od pozostałych klas obiektów. Do konstruowania klasyfikacji mogą być wykorzystane różne metody klasyfikacyjne, na przykład drzewa decyzyjne, których tworzenie odbywa się przez rekurencyjny podział zbioru na podzbiory aż do uzyskania ich jednorodności ze względu na przynależność obiektów do klas. Aby zbudowane drzewo było jak najmniejsze, dokonuje się jego porządkowania, usuwając te fragmenty, które mają niewielkie znaczenie dla jakości wyników klasyfikacji. Każdy algorytm tworzący drzewo decyzyjne musi rozwiązać trzy problemy [5]98, s. 170]:

- jak wybrać jedną lub kilka cech, w oparciu, o które nastąpi podział zbioru obiektów,
- kiedy zakończyć podział powstałego podzbioru obiektów, oraz
- w jaki sposób przydzielić obiekty znajdujące się w liściu drzewa do pewnej klasy.

Efektywność algorytmu zależy od sposobu podziału zbiorów obiektów w węzłach drzewa, a więc pojedynczych cech lub ich kombinacji liniowych. Wyboru dokonujemy w oparciu o pewną miarę jakości podziału (miary jednorodności lub zróżnicowania). W przypadku miary jednorodności wybieramy podział, który maksymalizuje wartość stosowanej miary. Wybierając miary zróżnicowania – podział, który minimalizuje jej wartość. Jeśli przyjmiemy, że $O = \{o_1, o_2, ..., o_n\}$- będzie zbiorem obiektów należących do

jednej z klas K_1, K_2, ..., K_k , przy czym liczność klasy K_i oznaczymy jako l_i - wówczas dla każdego zbioru obiektów możemy zbudować wektor prawdopodobieństwa przynależności do klas w postaci:

$$p = (p_1, p_2, ..., p_k) = (\frac{l_1}{n}, \frac{l_2}{n}, ..., \frac{l_k}{n}), \quad \text{gdzie } \sum_{i=1}^{k} p_i = 1. \qquad (1)$$

Możemy powiedzieć, że pewien zbiór obiektów jest jednorodny, gdy $\exists i=1,...k\ p_i=1$. Natomiast jego maksymalne zróżnicowanie występuje wówczas, gdy $\forall i=1,...,k\ p_i= 1/n$.

Grupowanie pozwala identyfikować grupy zdarzeń lub podobnych do siebie obiektów ze względu na kryteria. Wykorzystując określony sposób pomiaru odległości (podobieństwa) obiektów w wielowymiarowej przestrzeni cech, można zbiór podzielić na podzbiory tak, aby zawierały obiekty najbardziej do siebie podobne. Można tu wykorzystać jedną z poniższych technik:

- optymalno-iteracyjne (dokonuje się podziału zbioru na k rozłącznych podzbiorów, gdzie k jest podane przez badacza),
- hierarchiczne (w ramach których skupienia tworzą binarne drzewa, liście reprezentują obiekty, a węzły ich grupy; skupienia wyższych poziomów zawierają w sobie skupienia niższych poziomów),
- tworzące skupienia nierozłączne (niektóre obiekty ze zbioru mogą należeć do więcej niż jednej grupy) [5].

Technika odkrywania podobieństw na podstawie wzorców najczęściej wykorzystywana jest do analizy szeregów czasowych, a więc zbiorów danych, w których jednym z atrybutów jest czas bądź inny atrybut zależny od czasu. Możemy mieć tutaj do czynienia z dwoma przypadkami: zapytaniami związanymi z określonym wzorcowym obiektem, których celem jest znalezienie obiektów spełniających wcześniej zdefiniowane warunki dotyczące podobieństwa do określonego wzorcowego obiektu, lub zapytaniami porównującymi wszystkie pary elementów ze sobą, których celem jest znalezienie par obiektów spełniających określony przez użytkownika warunek podobieństwa [3, s. 866 - 883].

W rozproszonym środowisku pomiędzy dokumentami i obiektami utrzymywane są połączenia, które ułatwiają interaktywny dostęp do nich. Zrozumienie wzorców dostępu użytkowników w takim środowisku nie tylko ułatwia projektowania systemu, lecz również prowadzi do podejmowania lepszych decyzji marketingowych. Uchwycenie wzorców dostępu użytkownika w środowiskach rozproszonych określane jest mianem odkrywania wzorców ścieżek powiązań w sposób krzyżowy.

Zanim odkryta wiedza zostanie przeznaczona od użytku, podlega ocenie (interpretacji), jaką jest jej weryfikacja. Weryfikacja ma na celu identyfikację oraz skorygowanie takich niepożądanych właściwości wiedzy, jak niekompletność i niespójność. Obydwa rodzaje anomalii wiedzy występują w różnym stopniu, zależnie od sposobu reprezentowania wiedzy czy dziedziny, której dotyczy. Ocena odkrytej wiedzy może być realizowana na dwa sposoby. Oceny może dokonać ekspert dziedzinowy i/lub można sporządzać oceny w sposób zautomatyzowany [14, s. 111]. Proces oceniania odkrytej wiedzy sprowadza się do wyznaczenia jej zgodności z założeniami i celami data mining. Przyjmuje się zazwyczaj dwa podstawowe kryteria weryfikacji wiedzy, jakimi są kompletność oraz spójność [11].

Kompletność oznacza, że wiedza - będąca do dyspozycji określonego podmiotu - jest wystarczająca do generowania odpowiednich wniosków wynikających z wyznaczonego celu systemu. Ujmując ten problem w rozumieniu bardziej intuicyjnym - wiedza kompletna oznacza „pokrycie” wszystkich możliwych przypadków, w jakich będzie ona

wykorzystywana (por. np. [11]). W praktyce – w szczególności w odniesieniu do generowanych baz wiedzy – mamy do czynienia z pewnym podzbiorem przekształconej wiedzy dziedzinowej lub bazy wiedzy utworzonej na podstawie rozwiązywanych problemów, którą możemy uznać za kompletną wobec wykonywanych zadań.

Przyjmuje się z kolei, że baza wiedzy jest *spójna*, jeżeli w bazie faktów nie ma takich, które dla określonych więzów spójności (ang. *consistency constraints*) nie pozwoliłyby na realizację celów systemu. Więzy spójności dotyczą istotnych dla bazy właściwości strukturalnych jak przykładowo wykluczania wiedzy konfliktowej czy redundantnej. Są one dość często ignorowane dla generowanych baz wiedzy.

Istnieją różne techniki weryfikacji odkrytej wiedzy. Zalicza się do nich takiej, jak [11]:

- podejścia wykorzystujące *tablice decyzyjne* (ang. *decision-table*) zaimplementowane w procesorze Expert System Checker przez B. J. Craguna i H. J. Steudela [4]. Dla weryfikowanej bazy wiedzy jest opracowywana obszerna tabela decyzyjna na podstawie zdefiniowanych reguł, która następnie podlega podziałowi na podtabele dotyczące zbliżonych pod względem zawartości reguł,
- zastosowanie *metawiedzy* (ang. *metaknowledge*), zaproponowanej przez L. J. Morella 6. Wyodrębnia on podczas weryfikacji spójność statyczną obok dynamicznej, natomiast kompletność weryfikuje podczas aktywizacji procedur wnioskowania; całość weryfikacji konfrontuje z modelem bazy wiedzy interpretowanym jako metawiedza,
- wykorzystanie *grafów zorientowanych* (ang. *directed graphs*), wprowadzonych przez D. L. Nazaretha oraz M. H. Kennedy'ego [8], ułatwia weryfikacje bazy wiedzy poprzez klarowną prezentację zależności miedzy regułami bazy. Podstawą weryfikacji jest przyjęta klasyfikacja potencjalnych błędów pojawiających się w regułach i dotyczących strony leksykalnej, składniowej, strukturalnej oraz semantycznej,
- technika wykorzystująca tak zwane „K-drzewa" (ang. *K-tree*), opracowane przez Y. H. Suha i T. J. Murraya, odwołująca się do specyficznego przypadku *drzew decyzyjnych*, pozwala nie tylko na weryfikacje bazy wiedzy, ale umożliwia ponadto (w ograniczonym stopniu) jej udoskonalenie [SUMU94].

M. Owoc w ramach powyższych technik wyróżnia *przyrostową* (*stopniową) weryfikację* (ang. *incremental verification*). Podejście to służy w zasadzie do wartościowania tych elementów wiedzy, które zostały do niej wprowadzone w czasie późniejszym - po pierwotnej weryfikacji. Zapobiegać to ma powtarzaniu, zbędnych w takim kontekście, procedur weryfikacji bazy wiedzy wcześniej przetestowanej. Do grupy innych technik zalicza *maszynowe uczenie* się procedur weryfikacyjnych (ang. *machine learning approach*). Technika ta polega na sukcesywnym testowaniu wiedzy pozyskiwanej przez system poprzez wykrywanie nieprawidłowości dotyczących niekompletności oraz formalnej poprawności reguł bazy wiedzy. Idea uczenia polega na generowaniu tak zwanych węzłów poprawności bazy wiedzy [11].

Odkryta wiedza prezentowana jest menedżerowi w żądanej przez niego postaci. Stosowane są różne techniki wizualizacji, jak zestawienia tabelaryczne, wykresy, opisy, etc.

4. Podsumowanie

W artykule przedstawiono techniki przetwarzania struktur danych w hurtowni danych oraz zaprezentowano techniki data mining odkrywania wiedzy z hurtowni danych. Należy oczekiwać w nieodległej przyszłości pojawienia się standardów *de jure* oceny odkrytej

wiedzy, albowiem jak na razie takowych brak.

Literatura

1. Agrawal R., Srikant R.: Fast Algorithms for Mining Association Rules in Large Databases, Proceedings of the 20th International Conference on Very Large Data Bases, September 1994, s. 478-499.
2. Berry M.J.A., Linoff G.: Data Mining Techniques for Marketing, Sales and Customer Support, Wiley Computer Publishing, 1997.
3. Chen M.S., Han J., Yu P.S.: Data Mining: An Overview from a Database Perspective, IEEE Transactions on Knowledge and Data Engineering, 8(6): p.866-883, 1996.
4. Cragun B.J., Steudel H.J.: A Decision-Table-Based Processor for Checking of Completeness and Consistency in Rule-Based Expert Systems. IEEE 1989.
5. Gatner E.: Symboliczne metody klasyfikacji danych, PWN, Warszawa, 1998, s. 170.
6. Morell L.J.: Use of Metaknowledge in the Verification of Knowledge-Based Systems. IEEE, 1989.
7. Morzy T.: Przetwarzanie danych w magazynach danych, [w:] Projektowanie i implementowanie magazynów (hurtowni) danych. V seminarium PLOUG, Warszawa 29.05.2002.
8. Nazareth D. L., Kennedy M. H.: Verification of Rule-Based Knowledge using Directed Graphs. [w:] Knowledge Acquisition, Academic Press Ltd., 1991.
9. Nycz M. (red.): Pozyskiwanie wiedzy menedżerskiej. Podejście technologiczne, Wyd. AE, Wrocław, 2007.
10. Nycz M.: Problemy związane z pozyskiwaniem wiedzy z baz danych, [w:] Prace Naukowe nr 850, Wyd. AE, Wrocław, 2000.
11. Owoc M.: Wartościowanie wiedzy w inteligentnych systemach wspierających zarządzanie, Wyd. AE, Wrocław, 2004.
12. Smok B. (red.): Środowisko ORACLE w odkrywaniu wiedzy z baz danych, Wyd. UE, Wrocław, 2008.
13. Suh Y. H., Murray T. J.: A Tree-Based Approach for Verifying Completeness and Consistency in Rule-Based Systems, Expert Systems with Applications, Vol. 7, No.2, 1994.
14. Owoc M. (red.): Elementy systemów ekspertowych, Część 1. Sztuczna inteligencja i systemy ekspertowe, Wydawnictwo Akademii Ekonomicznej im. O.Langego we Wrocławiu, Wrocław, 2006.

Dr hab. inż. Małgorzata NYCZ
Katedra Systemów Sztucznej Inteligencji
Instytut Informatyki Ekonomicznej
Wydział Zarządzania, Informatyki i Finansów
Uniwersytet Ekonomiczny we Wrocławiu
54-345 Wrocław, ul. Komandorska 118/120
tel.: (0-71) 36-80-507
e-mail: malgorzata.nycz@ue.wroc.pl