Co dalej z tą eksploracją? Rola i miejsce eksploracji danych w architekturze współczesnych systemów baz danych.

Mikołaj Morzy
Instytut Informatyki Politechniki Poznańskiej
e–mail: Mikołaj.Morzy@put.poznan.pl

Abstrakt. Eksploracja danych (ang. *data mining*) to nowa, prężnie rozwijająca się dziedzina, zajmująca się metodami i algorytmami automatycznego pozyskiwania wiedzy z dużych wolumenów danych. Swą popularność zawdzięcza szerokiej stosowalności metod eksploracji w różnych domenach aplikacyjnych oraz komercyjnemu znaczeniu odkrywanej wiedzy. Odkryta wiedza znajduje zastosowanie w optymalizacji procesów biznesowych, wykrywaniu nieprawidłowości i osobliwości, oraz predykcji przyszłych zdarzeń i zachowań. Historycznie rzecz ujmując, pierwsze systemy eksploracji danych były tworzone w architekturze klient-serwer i stanowiły, z jednej strony, implementację wybranych algorytmów, a z drugiej strony, środowisko wizualizacji analizowanych danych i odkrytej wiedzy. Ostatnie lata przyniosły istotną zmianę. Narzędzia eksploracji danych podążają ścieżką wytyczoną przez narzędzia analizy wielowymiarowej OLAP i są często włączane do jądra systemu zarządzania bazą danych jako naturalne rozszerzenie funkcjonalności oferowanej przez system baz danych.

W niniejszym artykule zaprezentowano krótkie wprowadzenie do eksploracji danych oraz naszkicowano historię rozwoju dziedziny na przestrzeni ostatnich lat. Główna część artykułu jest poświęcona przedstawieniu dzisiejszej roli i pozycji systemów eksploracji danych we współczesnych systemach baz danych. Jako przykład opisano architekturę systemu Oracle i szczegółowo zaprezentowano moduł Oracle Data Mining, skupiając się na stopniu integracji systemu eksploracji danych z systemem zarządzania bazą danych. Przedstawiono także architekturę interfejsów programistycznych dla języków PL/SQL i Java, rozszerzenia języka SQL dla eksploracji danych oraz narzędzia zewnętrzne do wizualizacji danych i odkrywanej wiedzy.

1. Wprowadzenie

Od piętnastu lat jesteśmy świadkami gwałtownie przyspieszającego procesu gromadzenia ogromnych ilości danych. Zjawisko to wciąż przybiera na sile, a tempo zwiększania objętości repozytoriów danych nie tylko nie słabnie, wręcz przeciwnie, rozmiary współczesnych baz danych na przestrzeni ostatnich lat rosną lawinowo. Według raportu Winter Corporation [Wint05] rozmiar największej operacyjnej bazy danych w roku 2005 osiągnął 23 TB (Land Registry for England and Wales), podczas gdy rozmiar największej hurtowni danych przekroczył 100 TB (Yahoo!). Na przebicie tych wyników nie przyjdzie nam czekać zbyt długo. W maju 2008 zostanie oddany do użytku akcelerator wiązek protonowych LHC, dla którego zaprojektowano bazę danych umożliwiającą składowanie eksabajta danych (1 EB = 1024 PB = 10^{18} B) [LHC05]. Szacuje się, że akcelerator będzie generował 15 petabajtów danych rocznie ze średnią prędkością 1,5 GB/sek. Już dziś eksperymenty naukowe wykorzystujące akcelerator LHC są zaplanowane na najbliższych piętnaście lat, co w efekcie daje niewyobrażalny wolumen 225 petabajtów danych do przeanalizowania. Rzecz jasna, przypadek akceleratora LHC jest ekstremalny, ale symptomatyczny. W każdej dziedzinie ludzkiej działalności, do której wkroczyły komputery, ilość gromadzonych danych gwałtownie rośnie. Ma to związek, z jednej strony, ze spadkiem ceny mediów do składowania informacji cyfrowych, a z drugiej strony, z upowszechnieniem się technologii służących do automatycznego pobierania danych. Spadek cen mediów do składowania danych jest o wiele większy, niż spadek cen pamięci RAM czy spadek cen procesorów w przeliczeniu na flopsy. Oznacza to, że w praktyce koszt urządzeń do składowania nawet ogromnych ilości danych jest całkowicie pomijalny (przykładowo, setki tysięcy użytkowników poczty elektronicznej GMail ma dziś do dyspozycji za darmo skrzynki pocztowe o pojemności 3GB i pojemność ta nieustannie rośnie). W efekcie coraz częściej spotykamy się z sytuacją, w której bazy danych mają charakter tylko i wyłącznie przyrostowy, nowe dane są dodawane do bazy danych a stare dane nie są nigdy usuwane, co najwyżej podlegają mechanizmowi postarzenia i przeniesienia na media drugorzędne. Z drugiej strony, szerokie upowszechnienie takich technologii, jak czytniki kodów kreskowych, bezprzewodowe czujniki wyposażone w adaptery radiowe, czy digitalizacja dokumentów w urzędach publicznych, firmach ubezpieczeniowych i bankach powodują, że pozyskiwanie i gromadzenie danych staje się zadaniem coraz łatwiejszym, a służące do tego technologie stają się tańsze i powszechniejsze.

Zmiany społeczne wymuszone przez rewolucję informacyjną stanowią sprzężenie zwrotne, istotnie wpływające na ewolucję systemów informatycznych. Trudno dziś wskazać jakąkolwiek dziedzinę ludzkiego życia, w której komputery nie stanowiłyby ważnego komponentu infrastruktury. W szczególności, nie sposób sobie dziś wyobrazić żadnego systemu informatycznego, którego częścią nie byłby system baz danych. Systemy baz danych są z powodzeniem stosowane w tak różnych dziedzinach jak: bankowość, finanse, ubezpieczenia, handel detaliczny, administracja publiczna, usługi medyczne, handel elektroniczny, i wiele wiele innych. Ewolucja systemów baz danych analizowana na przestrzeni ostatnich trzech dziesięcioleci jest fascynującą historią ciągłych zmian, adaptacji i nieustannego rozszerzania oferowanej funkcjonalności. Począwszy od prostych systemów plikowych, poprzez sieciowe i hierarchiczne systemy baz danych, a kończąc na systemach relacyjnych, post-relacyjnych, obiektowych i semistrukturalnych, rozwój systemów baz danych wyznaczał kamienie milowe rozwoju informatyki. Z jednej strony rozwój baz danych był podyktowany rosnącymi wymaganiami użytkowników i koniecznością implementowania coraz to bardziej złożonych modeli danych, oferujących potężne operatory, z drugiej strony, od kolejnych systemów baz danych wymagano efektywnego przetwarzania coraz większych kolekcji danych. W ten sposób w trakcie swego rozwoju bazy danych musiały ewoluować w sposób stanowiący kompromis między złożonością i bogactwem oferowanego modelu a efektywnością przetwarzania. Jak się okazało, najskuteczniejszą strategią rozwoju okazała się niczym nie pohamowana ekspansja.

Analizę rozwoju systemów baz danych warto rozpocząć od początku lat 90-tych XX wieku. W punkcie wyjścia powszechnie przyjętym standardem był system zarządzania relacyjną bazą danych implementujący język dostępu SQL i umożliwiający dostęp do danych z poziomu aplikacji klienckiej za pomocą interfejsu zastrzeżonego dla danego systemu. Od początku lat 90-tych było coraz bardziej oczywiste, że taki model danych spełniał swoje zadania w przypadku prostych aplikacji finansowo-księgowych, ale nie umożliwiał przeprowadzania złożonych analiz. Z drugiej strony, wolumeny danych gromadzonych przez przedsiębiorstwa i organizacje gwałtownie rosły. Potrzeby informacyjne przedsiębiorstw wymusiły opracowania nowego modelu przetwarzania, nazwanego przetwarzaniem analitycznym OLAP (ang. *online analytical processing*) oraz wielowymiarowego modelu danych, który oferował operatory do agregacji i analizy statystycznej gromadzonych danych. Prostą konsekwencją opracowania koncepcji modelu OLAP było powstanie hurtowni danych (ang. *data warehouse*), specjalizowanej architektury systemu baz danych przeznaczonej do weryfikacji hipotez, zaawansowanego raportowania i wspomagania decyzji w oparciu o scenariusze i dogłębną wielowymiarową analizę danych. Przez pewien czas model wielowymiarowy był implementowany w postaci osobnych modułów, zwanych serwerami danych wielowymiarowych. Serwery danych wielowymiarowych implementowały natywnie operatory analizy wielowymiarowej, takie jak obrót (ang. *pivot*), przycięcie (ang. *slice-and-dice*), czy drążenie (ang. *drill-down*) oraz zapewniały mechanizmy składowania i indeksowania danych w specjalizowanych strukturach danych. Ponieważ jedną z podstawowych operacji w kontekście hurtowni danych jest ładowanie hurtowni danymi odczytywanymi z operacyjnej bazy danych, stosunkowo szybko pojawiła się koncepcja zaimplementowania serwera danych wielowymiarowych jako integralnej części serwera relacyjnej bazy danych. Faktycznie, rozwiązanie takie, znane pod nazwą ROLAP (ang. *relational OLAP*) po niedługim czasie doczekało się pierwszych komercyjnych implementacji. Początkowo stopień integracji serwera danych wielowymiarowych z serwerem bazy danych był niewielki – dane wielowymiarowe były przechowywane fizycznie w osobnych plikach o specyficznej strukturze a interfejs programistyczny opierał się na niestandardowych językach zapytań. Jednak w kolejnych wydaniach popularnych serwerów baz danych integracja tradycyjnej bazy danych z hurtownią danych następowała coraz ściślej. Współczesna baza danych pozwala, z poziomu języka SQL, definiować struktury danych specyficzne dla hurtowni danych (tabele faktów i wymiarów, indeksy bitmapowe, perspektywy materializowane), wykonywać operatory analizy danych wielowymiarowych (polecenia `CUBE`, `ROLLUP`, `GROUPING SETS`), czy wreszcie uwzględniać charakterystykę zapytań kierowanych do hurtowni danych podczas optymalizacji zapytań (transformacja gwiaździsta, zapytania gwiaździste). Można śmiało powiedzieć, że współczesne systemy baz danych „połknęły” serwery danych wielowymiarowych i całkowicie przejęły ich funkcjonalność, oferując ją jako naturalne rozszerzenie funkcjonalności systemów baz danych. Jak się okazuje, serwery danych wielowymiarowych stanowiły tylko początek długiej serii takich „wrogich przejęć”.

Począwszy od połowy lat 90-tych XX wieku coraz większe znaczenie zaczęły sobie zdobywać dane multimedialne. Związane to było, z jednej strony, z gwałtownym rozwojem Internetu, a z drugiej strony, z upowszechnieniem się stosunkowo tanich urządzeń osobistych do rejestracji i przetwarzania zawartości multimedialnej (kamery cyfrowe, odtwarzacze muzyki, dyktafony cyfrowe, itp.) Mogłoby się wydawać, że trudno znaleźć dane mniej odpowiednie do przechowywania w modelu relacyjnym niż dane multimedialne. I tu jednak potrzeba okazała się matką wynalazku. W bardzo krótkim czasie relacyjne bazy danych zostały wzbogacone o zdolności obiektowe poprzez wcielenie modelu obiektowego do jądra systemu bazy danych. Operatory charakterystyczne dla systemów przetwarzania obiektowego, takie jak wielopostaciowość, dziedziczenie, enkapsulacja, czy możliwość definiowania abstrakcyjnych typów danych, zostały zaimplementowane jako operatory na poziomie języka SQL i dały początek obiektowo-relacyjnym systemom baz danych. To z kolei umożliwiło zaprojektowanie specjalnych typów danych zoptymalizowanych pod kątem przechowywania i przetwarzania zawartości multimedialnej. Początkowo pojawiła się możliwość przechowywania w relacyjnej bazie danych dużych obiektów binarnych (przy użyciu typu `BLOB`, ang. *binary large object*), oraz dużych kolekcji tekstowych (przy użyciu typu `CLOB`, ang. *charakter large object*). Wkrótce pojawiły się także specjalne typy danych do przechowywania obrazów, dźwięków, czy sekwencji wideo. Na poziomie implementacyjnym zazwyczaj możliwość przechowywania złożonych typów danych była oferowana pod postacią rozszerzenia (ang. *extender*) systemu baz danych, ewentualnie pod postacią osobno płatnej i licencjonowanej opcji. Ukoronowaniem procesu wchłaniania danych multimedialnych do relacyjnej bazy danych było ukonstytuowanie się standardu języka zapytań do danych multimedialnych, znanego jako SQL/MM. Dziś możliwość przechowywania zawartości multimedialnej w bazie danych jest dla większości osób oczywistym i naturalnym rozszerzeniem funkcjonalności relacyjnych baz danych, warto jednak pamiętać, że taka możliwość jest w rzeczywistości oferowana od stosunkowo niedawna.

Przełom wieków to okres gwałtownego rozwoju systemów informacji geograficznej. Główną siłą sprawczą, napędzającą rozwój rynku systemów GIS (ang. *geographical information system*) była bez wątpienia rewolucja w bezprzewodowym dostępie do sieci Internet i upowszechnienie się standardu lokalizacji obiektów GPS (ang. *global positioning system*). Znaczący spadek cen urządzeń bezprzewodowych, które były wyposażone w nadajniki GPS, umożliwił szybki rozwój usług zależnych od lokalizacji klienta, czyli usług LBS (ang. *location-based service*). W efekcie bardzo szybko wzrosła ilość danych lokalizacyjnych gromadzonych przez systemy informatyczne. Należy zauważyć, że charakter zapytań do bazy danych geograficznych jest zupełnie inny od charakteru tradycyjnych zapytań. Usługi LBS bazują na zapytaniach topologicznych, których przeformułowanie na język SQL i relacyjny model danych jest zadaniem często karkołomnym, a czasem niemożliwym. Zapytania typowe dla bazy danych geograficznych, to np. „czy Poznań i Warszawa leżą nad tą samą rzeką?" „czy województwa wielkopolskie i mazowieckie mają wspólną granicę?" „czy obszary występowania wilka i żbika posiadają część wspólną?" „jakie miasto wojewódzkie leży najbliżej Siemiatycz?" itp. W szczególności, zapytania topologiczne o zawieranie się, przecinanie się, czy sąsiedztwo, należą do najtrudniejszych i najbardziej kosztownych. Efektywne przetwarzanie takich zapytań wymaga specjalizowanych ścieżek dostępu (indeksy typu R-drzewo, KD-drzewo) oraz specjalnych metod optymalizacji zapytań. Mimo tak dużych różnic, i w tym przypadku relacyjne systemy baz danych zdołały zdominować rynek i zaoferować rozwiązania, które okazały się zadowalające dla zastosowań komercyjnych. Dziś każdy duży system zarządzania relacyjną bazą danych zawiera komponent umożliwiający składowanie, przetwarzanie i wyszukiwanie informacji geograficznych za pomocą specjalnych języków umożliwiających formułowanie dowolnie złożonych zapytań topologicznych.

Pierwsze publikacje naukowe dotyczące eksploracji danych (ang. *data mining*) pojawiły się w 1994 roku. Trzeba było jednak kilku lat, aby dziedzina eksploracji danych wzbudziła większe zainteresowanie, przede wszystkim wśród potencjalnych konsumentów tej technologii, czyli wśród osób zajmujących się analizą danych i podejmowaniem decyzji w organizacjach i przedsiębiorstwach. Począwszy od końca lat 90-tych XX wieku obserwujemy jednak gwałtowny wzrost zainteresowania technikami eksploracji danych. Pierwsze informacje o praktycznych zastosowaniach wiedzy odkrytej w ogromnych repozytoriach danych spowodowały, że eksploracja danych stała się z dnia na dzień modnym tematem. Niestety, o ile rynek wskazywał wyraźnie zapotrzebowanie na profesjonalne

narzędzia do eksploracji danych, o tyle pojawienie się pierwszych takich narzędzi nie nastąpiło od razu. Ze względu na duży stopień skomplikowania dziedziny pierwsze systemy komercyjne stanowiły zaledwie rozwinięcie prototypów tworzonych w świecie akademickim i wzbogacały owe prototypy o czytelny interfejs użytkownika oraz mechanizmy wizualizacji danych i wzorców.

2. Metody eksploracji danych

Eksploracja danych to „...proces odkrywania nowych, wcześniej nieznanych, potencjalnie użytecznych, zrozumiałych i poprawnych wzorców w bardzo dużych wolumenach danych" [FPSU96]. Jest to dziedzina wybitnie interdyscyplinarna, znajdująca się na przecięciu takich dziedzin, jak: . systemy baz danych, statystyka, systemy wspomagania decyzji, sztuczna inteligencja, uczenie maszynowe, czy wizualizacja danych. Wiedza odkryta w danych może być postrzegana jako wartość dodana, podnosząca jakość danych i znacząco polepszająca jakość decyzji podejmowanych na podstawie danych. Do prezentowania wiedzy odkrytej w bazie danych wykorzystywany jest model wiedzy. Każdy model wiedzy odpowiada innemu rodzajowi wzorca znalezionego w danych, wyposażony jest we własne miary statystycznej oceny i może posiadać specjalizowane metody wizualizacji i agregacji. Najpopularniejsze modele wiedzy to reguły asocjacyjne [AIS93], reguły dyskryminacyjne i charakterystyczne [Cen87], klasyfikatory bayesowskie [LIT92], drzewa decyzyjne [Qui86], wzorce sekwencji [AS95], skupienia obiektów [ELL01], przebiegi czasowe i osobliwości.

Techniki eksploracji danych można podzielić na dwie rozłączne grupy zgodnie z dwoma klasyfikacjami. Pierwsza klasyfikacja dzieli techniki eksploracji danych na techniki predykcyjne i techniki deskrypcyjne. Techniki predykcyjne starają się, na podstawie wzorców odkrytych w dużych wolumenach danych, dokonać przewidywania cech, wartości i zachowań obiektów. Techniki deskrypcyjne służą do automatycznego formułowania uogólnień dotyczących danych, w celu uchwycenia ogólnych cech opisywanych obiektów. Druga klasyfikacja dokonuje podziału technik eksploracji danych ze względu na charakter wykorzystywanych danych źródłowych. Techniki uczenia nadzorowanego (ang. *supervised learning*) wykorzystują zbiory danych, w których każdy obiekt posiada etykietę przypisującą obiekt do jednej z predefiniowanych klas. Proces uczenia polega na zbudowaniu modelu wiedzy zdolnego do „odróżniania" obiektów należących do różnych klas. Model taki może posłużyć w przyszłości do przypisania nowego obiektu do klasy, w przypadku gdy takie przypisanie nie jest znane. Łatwo się domyślić, że techniki uczenia nadzorowanego są silnie zależne od jakości zbioru uczącego, w szczególności, od kompletności tego zbioru i poprawności etykiet. Najczęściej spotykanymi technikami uczenia nadzorowanego są techniki klasyfikacji (drzewa decyzyjne [Qui86], algorytmy bazujące na *n* najbliższych sąsiadach [Aha92], sieci neuronowe [MI94], statystyka bayesowska [Bol04]), oraz techniki regresji. Techniki uczenia bez nadzoru (ang. *unsupervised learning*) są wykorzystywane wtedy, gdy żadne etykiety obiektów nie są znane. Techniki te starają się sformułować model wiedzy maksymalnie zgodny z obserwowanymi danymi. Przykłady technik uczenia bez nadzoru obejmują techniki analizy skupień [ELL01] (ang. *clustering*), samoorganizujące się mapy [Koh00], oraz algorytmy maksymalizacji wartości oczekiwanej [DLR77] (ang. *expectation-maximization*).

W literaturze często pojawiają się terminy eksploracja danych (ang. *data mining*) i odkrywanie wiedzy w bazach danych (ang. *knowledge discovery in databases*, *KDD*). Wbrew powszechnemu przekonaniu, nie są to synonimy. Odkrywanie wiedzy obejmuje proces akwizycji wiedzy, począwszy od wyboru danych źródłowych, poprzez czyszczenie, transformację, kompresję danych, odkrywanie wzorców, a skończywszy na walidacji i ocenie odkrytych wzorców. W tym kontekście eksploracja danych stanowi zaledwie pojedynczy krok procesu okrywania wiedzy w bazach danych i oznacza zastosowanie wybranego algorytmu. Poniżej przedstawiono, z konieczności bardzo skrótowe, opisy najpopularniejszych technik eksploracji danych.

- **reguły asocjacyjne**: reprezentują współwystępowanie podzbiorów elementów w dużej kolekcji zbiorów, najczęściej stosowane do analizy koszyka zakupów, gdzie badaną kolekcją jest zbiór transakcji dokonanych przez klientów, a znalezione podzbiory odpowiadają produktom, których sprzedaż jest ze sobą powiązana, przykładowa reguła asocjacyjna mogłaby wyglądać tak: `{makaron,anchois}⇒{kapary} (0.5%,65%)` , co należy zinterpretować w następujący sposób:

0.5% klientów kupiło w trakcie jednej wizyty w sklepie makaron, anchois i kapary, przy czym 65% klientów, którzy kupili makaron i anchois, kupili również kapary. Znalezione reguły asocjacyjne można wykorzystać do konstrukcji ofert sprzedaży wiązanej, budowania promocji, ustawiania produktów na półkach.

- **wzorce sekwencji**: stanowią rozwinięcie modelu reguł asocjacyjnych o element następstwa zdarzeń, reprezentują podsekwencje zdarzeń elementarnych występujące często w bazie danych sekwencji. Przykładowy wzorzec sekwencji to: `{Ojciec chrzestny}⇒{Kasyno}⇒{Człowiek z blizną} (1.5%)`, a jego interpretacja jest następująca: 1.5% klientów wypożyczalni wypożyczyło wymienione filmy w takiej właśnie kolejności. Dodatkowo, kolejne wystąpienia elementów wzorca mogą być ograniczone przez szerokość okna czasowego, wewnątrz którego muszą się znaleźć, aby utworzyć wzorzec sekwencji. Wzorce sekwencji można stosować do analizy koszyka zakupów, ale także do znajdowania częstych sekwencji w logach serwerów WWW, do odkrywania częstych sekwencji w historii połączeń telekomunikacyjnych, czy do znajdowania sekwencji sygnałów świadczących o grożącej awarii sieci komputerowej.
- **klasyfikacja**: polega na zbudowaniu modelu przypisującego nowy, wcześniej nie widziany obiekt, do jednej ze zbioru predefiniowanych klas, przy czym przypisanie to następuje na podstawie doświadczenia nabytego przez model w fazie uczenia na zbiorze uczącym. Klasyfikacja to technika rozwijana równolegle w domenach sztucznej inteligencji, uczenia maszynowego, wspomagania decyzji i eksploracji danych, a wynikiem prowadzonych prac było opracowanie dziesiątków, jeśli nie setek, algorytmów klasyfikacji. Najpopularniejsze algorytmy to klasyfikacja bayesowska, drzewa decyzyjne, sieci neuronowe, sieci bayesowskie, techniki bazujące na *k* najbliższych sąsiadach oraz algorytmy SVM.
- **regresja**: jest to technika bardzo podobna do klasyfikacji, a podstawowa różnica polega na tym, że w przypadku klasyfikacji zadanie polega na przewidzeniu wartości atrybutu dyskretnego, podczas gdy techniki regresji próbują, na podstawie doświadczenia zdobytego na zbiorze uczącym, przewidzieć nieznaną wartość atrybutu numerycznego. Regresja znajduje zastosowanie w analizie danych finansowych i systemach logistycznych (np. podczas przewidywania przyszłego poboru energii elektrycznej).
- **analiza skupień**: polega na podziale zbioru obiektów na partycje w taki sposób, aby jednocześnie maksymalizować podobieństwo między obiektami przypisanymi do tej samej partycji i minimalizować podobieństwo między obiektami przypisanymi do różnych partycji zgodnie z zadaną miarą podobieństwa między obiektami. Podobnie jak klasyfikacja, analiza skupień doczekała się dziesiątków algorytmów, wśród najpopularniejszych wymienić należy algorytm k-średnich, algorytm k-medoids, samoorganizujące się mapy Kohonena, algorytmy CURE, PAM, CLARA, CLARANS, Chameleon, i wiele innych.

3. Ewolucja systemów eksploracji danych

Jak już wcześniej wspomniano, pierwsze komercyjne systemy eksploracji danych pojawiły się stosunkowo późno i stanowiły najczęściej rozwinięcie akademickich prototypów. Najlepszym przykładem takiego systemu był program IBM Intelligent Miner [IBMIMin], którego wersja komercyjna była nieznacznie „podrasowaną" wersją prototypowego systemu Quest, rozwijanego w laboratorium badawczym IBM Almaden jako system eksperymentalny. Akademickie korzenie posiada również bardzo popularny pakiet Weka3 [Weka], który przez wiele lat był rozwijany jako projekt badawczy na Uniwersytecie Waikato. Symptomatyczne jest to, że praktycznie wszystkie narzędzia do eksploracji danych, jakie pojawiły się na rynku w pierwszym etapie rozwoju tej dziedziny, były tworzone w architekturze klient-serwer. System baz danych był w tej architekturze tylko biernym repozytorium dostarczającym surowe dane. Cały proces wyszukiwania wiedzy w bazie danych, począwszy od czyszczenia i wstępnego przetwarzania danych (identyfikacja osobliwości, dyskretyzacja atrybutów kategorycznych i numerycznych, normalizacja atrybutów) był wykonywany po stronie narzędzia do eksploracji. Takie rozwiązanie było jednak obarczone wieloma wadami. Po pierwsze, taka architektura w żaden sposób nie wykorzystywała w pełni potencjału systemu zarządzania bazą danych. Po drugie, ogromne wolumeny danych musiały być kopiowane ze źródłowej

bazy danych do systemu eksploracji danych, co pociągało za sobą duże koszty komunikacyjne. Po trzecie, uruchamianie algorytmów eksploracji danych na stacji klienckiej nakładało na nią bardzo wysokie wymagania sprzętowe. Wreszcie, wykorzystywanie dedykowanych systemów eksploracji danych powodowało, że wzorce odkryte w procesie eksploracji nie były przenaszalne między systemami i wykorzystanie odkrytych wzorców w procesie podejmowania decyzji wiązało się z koniecznością pisania specjalizowanych aplikacji analitycznych.

Pierwszą próbą wprowadzenia ładu do dziedziny eksploracji danych było wprowadzenie standardowego języka opisu modeli wiedzy PMML (ang. *predictive modeling markup language*) [PMML]. Jest to oparty na XML język opisu modeli i wzorców odkrytych w bazie danych. PMML umożliwia przenoszenie modeli i wzorców między różnymi narzędziami eksploracji danych oraz pozwala na standardowe składowanie modeli i wzorców w relacyjnej bazie danych. Dzięki wprowadzeniu tego standardu użytkownicy przestali być zależni od produktów określonego dostawcy, ponieważ wiedza odkryta za pomocą jednego narzędzia mogła być zapisana do bazy danych, przekazana do wizualizacji w narzędziu drugiego dostawcy, a stamtąd zaimportowana do systemu wspomagania decyzji oferowanego przez trzeciego dostawcę.

Kolejnym krokiem w historii rozwoju systemów eksploracji danych było opracowanie standardowego interfejsu eksploracji danych dla języka programowania Java. Jest to dziś najpopularniejszy język tworzenia aplikacji i standaryzacja procesu eksploracji przy użyciu języka Java bardzo szybko doprowadziła do powstania wielu interesujących narzędzi do eksploracji i wizualizacji danych. Aktualnie standard JDM 2.0 (ang. *Java Data Mining*) [JDM] jest rozwijany pod auspicjami konsorcjum JCP (ang. *Java Community Process*) jako standard JSR-247 i aktywnie wspierany przez największych graczy na rynku. Możliwość standardowego dostępu do metod i wyników eksploracji danych z poziomu języka Java powoduje, że eksploracja danych może być łatwo włączana do współczesnych aplikacji wielowarstwowych, a co za tym idzie, aplikacje te można wyposażyć w „inteligencję" nigdzie wcześniej nie spotykaną.

Wraz z pojawieniem się wersji systemu zarządzania bazą danych Oracle 9*i* pojawiła się na rynku nowa jakość: pierwsza baza danych włączająca funkcjonalność eksploracji danych do jądra systemu bazy danych. Pierwsza edycja produktu Oracle Data Mining (ODM) nie była zbyt udana, ale wraz z opracowaniem kolejnej edycji systemu zarządzania bazą danych, oznaczonej jako Oracle 10*g*, moduł ODM uległ całkowitemu przepisaniu i rozwinięciu. Wkrótce za firmą Oracle podążyli kolejni dostawcy systemów baz danych. Dziś eksploracja danych jest wspierana przez SQL Server 2005 i IBM DB2 Data Warehouse Edition. Te wiodące na rynku baz danych produkty oferują silniki do eksploracji danych jako części składowe jądra systemu bazy danych oraz dostarczają interfejsów programistycznych dla języków Java, PL/SQL, MDX i DMX. W większości przypadków umożliwiają także uruchamianie algorytmów eksploracji danych i przeglądanie znalezionych wzorców z poziomu języka SQL, zazwyczaj poprzez niestandardowe rozszerzenia i funkcje języka SQL (choć i tu w przyszłości można się spodziewać pierwszych prób standaryzacji). W kolejnym rozdziale przyjrzymy się bliżej konkretnemu systemowi eksploracji danych, produktowi Oracle Data Mining.

4. Case study – Oracle Data Mining

Pierwsza, nieudana wersja produktu Oracle Data Mining pojawiła się wraz z opublikowaniem wersji systemu zarządzania bazą danych Oracle 9*i* R2. Można powiedzieć, że pierwsza wersja ODM miała charakter prototypowy. Był to zbiór klas języka Java zainstalowany w wydzielonym schemacie bazy danych. Wszystkie klasy i biblioteki potrzebne do uruchomienia procesu eksploracji były umieszczone w schemacie `ODM`, a dane podlegające eksploracji musiały być umieszczone w schemacie `ODM_MTR`. Co gorsza, opracowano dwa niespójne interfejsy programistyczne: dla języka Java i dla języka PL/SQL. W momencie publikowania pierwszej wersji ODM standard JDM nie był jeszcze ukończony, stąd cały interfejs dla języka Java dla ODM był dedykowany i nieprzenaszalny. Co ciekawe, metody udostępniane przez Java Data Mining API i PL/SQL Data Mining API nie były nawzajem kompatybilne, część algorytmów była dostępna tylko i wyłącznie przez Java Data Mining API, a modele i wzorce opracowane za pomocą jednego interfejsu były całkowicie niedostępne

poprzez drugi interfejs. Dodatkowo, ODM nie zawierał żadnego narzędzia do wizualizacji danych, wzorców, ani do uproszczonego uruchamiania algorytmów.

Konieczność dostarczenia wygodniejszego interfejsu do ODM była tak paląca, że jeszcze przed premierą wersji Oracle 10*g* firma zdecydowała się na przygotowanie tymczasowej łatki w postaci rozszerzenia środowiska programistycznego JDeveloper. Rozszerzenie to, zwane DM4J (ang. *Oracle Data Mining for Java*), stanowiło zbiór kreatorów, za pomocą których można było w trybie graficznym uruchamiać algorytmy ODM. Dodatkowo, DM4J zawierał pierwsze proste narzędzia do wizualizacji uzyskanych modeli, w tym narzędzia do wyświetlania krzywych lift i ROC (ang. *receiver-operator characteristics*). Takie rozwiązanie pozostawiało cały proces eksploracji wewnątrz serwera bazy danych i wymuszało składowanie odkrytych wzorców w bazie danych, a jednocześnie ułatwiało pracę z systemem ODM i poszerzało potencjalne grono konsumentów tej technologii.

Na szczęście sytuacja diametralnie się zmieniła wraz z opublikowaniem systemu zarządzania bazą danych Oracle 10*g*. Najważniejsza zmiana dotyczyła interfejsu Java Data Mining API, cały interfejs został przepisany od podstaw i całkowicie uzgodniony ze standardem JDM. Firma Oracle zdecydowała się na drastyczny krok, polegający na całkowitym zerwaniu kompatybilności w dół i wymuszenie na klientach, którzy skorzystali z ODM w poprzedniej wersji, przepisania wszystkich aplikacji od nowa. Bez wątpienia przed podjęciem tego kroku przeprowadzono analizy biznesowe i symulacje kosztów. Można tylko przypuszczać, że w opinii firmy Oracle wcześniejsza edycja ODM była tak mało popularna, że firma mogła sobie pozwolić na zerwanie kompatybilności. Tradycyjnie już, nowa wersja ODM oferowała większą gamę algorytmów (dodano, między innymi, algorytm indukcji drzew decyzyjnych, algorytm odkrywania cech NNMF, oraz rodzinę algorytmów SVM do klasyfikacji, regresji i oznaczania osobliwości). Interfejsy Java Data Mining API i PL/SQL Data Mining API zostały uspójnione, zlikwidowano różnice implementacyjne między algorytmami uruchamianymi poprzez każdy z interfejsów. Dalej, modele i wzorce zbudowane za pomocą jednego interfejsu stały się dostępne dla wywołań metod drugiego interfejsu – to bardzo istotna zmiana zdecydowanie zwiększająca funkcjonalność aplikacji.

Oprócz usprawnień w jądrze systemu eksploracji danych firma Oracle przygotowała Oracle Data Miner [ODMin], aplikację w architekturze klient-serwer, która służy jako podstawowe narzędzie do programowania, uruchamiania, weryfikacji algorytmów i wizualizacji wyników. Oracle Data Miner jest darmowym narzędziem, które ściśle współpracuje z silnikiem bazy danych, a jednocześnie ogromnie upraszcza proces uruchamiania algorytmów i weryfikacji uzyskanych wyników. Należy tutaj wyraźnie zaznaczyć, że Oracle Data Miner jest zupełnie innym narzędziem niż pierwsze systemy eksploracji danych. Oracle Data Miner nie implementuje żadnych algorytmów eksploracji danych, a jedynie ułatwia uruchamianie algorytmów oferowanych przez ODM, stanowi zatem tylko graficzny *front-end* do systemu ODM. Jest to nieoceniona pomoc w ćwiczeniu i uczeniu się poszczególnych technik eksploracji danych oraz tworzeniu i prototypowaniu aplikacji eksploracyjnych.

Kolejną nowością w wersji Oracle 10*g* narzędzia Oracle Data Mining było wprowadzenie rozszerzeń języka SQL służących do odczytywania modeli eksploracji oraz stosowania wybranych modeli do danych w trybie *online*. Rozszerzenia języka SQL, dostarczane w postaci specjalizowanych funkcji, obejmują techniki klasyfikacji danych, analizy skupień oraz znajdowania cech. Mimo, że funkcje te nie są standardowe, bardzo ułatwiają upowszechnienie się technik eksploracji danych w aplikacjach bazodanowych, ponieważ umożliwiają zastosowanie modelu składowanego w bazie danych do nowych danych bez konieczności pisania jakiegokolwiek kodu. Cała funkcjonalność może zostać wprowadzona do aplikacji użytkownika przez tradycyjny mechanizm perspektyw. Poza funkcjami języka SQL, zbiór pakietów PL/SQL został wyposażony w dodatkowe pakiety pomocnicze, rozszerzające i tak już bogatą funkcjonalność narzędzia ODM. Ciekawym przykładem rozszerzenia jest pakiet `DBMS_PREDICTIVE_ANALYTICS`, zawierający procedury do całkowicie automatycznego wyznaczania względnej ważności atrybutów oraz klasyfikacji. Pakiet ten wpisuje się w popularny ostatnimi czasy nurt, który można określić jako „*one click data mining*”, czyli postulat tworzenia narzędzi eksploracji danych wymagających zerowej lub prawie zerowej interwencji użytkownika. W ten sam nurt wpisuje się także *Oracle Spreadsheet Add-In for Predictive Analytics*, wtyczka dla użytkowników arkusza kalkulacyjnego Microsoft Excel umożliwiająca trywialnie proste uruchomienie algorytmu określenia względnej ważności atrybutów oraz algorytmu budowania modelu klasyfikacji

bezpośrednio z arkusza kalkulacyjnego. Należy się spodziewać, że kolejne edycje narzędzia ODM będą zawierać więcej tego typu rozwiązań, które są zdolne do automatycznej adaptacji i doboru optymalnych wartości parametrów sterujących przebiegiem algorytmu, bez konieczności angażowania użytkownika końcowego.

5. Podsumowanie

W niniejszym artykule przedstawiliśmy historię rozwoju systemów eksploracji danych na przestrzeni ostatnich lat. Przeszły one zdumiewającą metamorfozę, od prostych aplikacji klient-serwer aż do integralnych części składowych systemu zarządzania bazą danych. Ewolucja systemów eksploracji danych wynika z ogólnej filozofii rządzącej rozwojem baz danych – coraz nowe obszary są wchłaniane przez systemy baz danych jako naturalne rozszerzenia podstawowej funkcjonalności systemu zarządzania bazą danych. W przyszłości należy spodziewać się zdecydowanej kontynuacji tego trendu. Rynek dedykowanych systemów eksploracji danych jest poważnie zagrożony przez następujący rozwój silników eksploracji danych dostarczanych wraz z bazami danych. Oczywiście, funkcjonalność, łatwość obsługi, czy wreszcie zdolność do wizualizacji danych i wzorców nadal pozostawiają wiele do życzenia i zdecydowanie na tym polu ustępują systemom dedykowanym. Niemniej jednak, oczywiste zyski wynikające ze ścisłej integracji systemu eksploracji danych z systemem zarządzania bazą danych powodują, że trend rozwojowy systemów eksploracji danych jest coraz silniej związany z systemami zarządzania bazami danych i na tym polu należy oczekiwać w niedalekiej przyszłości bardzo ciekawych propozycji.

Bibliografia

[Aha92] Aha D.: Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms. International Journal of Man-Machine Studies 36(2), pp.267-287

[AIS93] Agrawal R., Imielinski T., Swami A.: Mining association rules between sets of items in large databases, Proc. of 1993 ACM SIGMOD International Conference on Management of Data, Washington D.C., May 26-28 1993, pp. 207-216, ACM Press, 1993

[AS95] Agrawal R., Srikant R.: Mining sequential patterns, In Proc. of the 11th International Conference on Data Engineering, Taipei, Taiwan, 1995

[Bol04] Bolstad W.M.: Introduction to Bayesian statistics. Wiley-Interscience, 2004

[Cen87] Cendrowska J.: PRISM: An algorithm for inducing modular rules. International Journal of Man-Machine Studies 27(4), pp.25-32, 1987

[DLR77] Dempster A., Laird N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B, 39(1):pp.1-38, 1977

[ELL01] Everitt B.S., Landau S., Leese M.: Cluster analysis, Arnold Publishers, 2001

[FPSU96] Fayyad U.M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R.: Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996

[IBMIMin] IBM Intelligent Miner, http://www-306.ibm.com/software/data/iminer/

[JDM] Java Data Mining 2.0 Specification, http://www.jcp.org/en/jsr/detail?id=247

[Koh00] Kohonen T.: Self-organizing maps, Springer Verlag, 2000

[LHC05] LHC Computing Grid, http://lcg.web.cern.ch/LCG/index.html

[LIT92] Langey P., Iba W., Thompson K.: An analysis of Bayesian classifiers. In Proc. of 10th National Conference on Artificial Intelligence, San Jose, CA, AAAI Press, pp.223-228, 1992

[MI94] McCord Nelson M., Illingworth W.T.: Practical guide to neural nets, Addison-Wesley, 1994

[ODMin] Oracle Data Miner, http://www.oracle.com/technology/products/bi/odm/odminer.html

[PMML] Predictive Modeling Markup Language v.3.0, http://www.dmg.org/pmml-v3-0.html

[Qui86] Quinlan J.R.: Induction of decision trees. Machine Learning 1(1),pp.81-106

[Weka] Weka3 Data Mining Software, http://www.cs.waikato.ac.nz/ml/weka/

[Wint05] Winter Corporation TopTen Program, http://www.wintercorp.com