Building Data Mining Models in the Oracle 9i Environment

Krzysztof Hauke, Mieczyslaw L. Owoc, and Maciej Pondel
Wroclaw University of Economics, Poland

krzysztof.hauke@ae.wroc.pl **mieczyslaw.owoc@ae.wroc.pl**
m-pondel@credit.ae.wroc.pl

Abstract

Data Mining (DM) is a very crucial issue in knowledge discovery processes. The basic facilities to create data mining models were implemented successfully on Oracle 9i as the extension of the database server. DM tools enable developers to create Business Intelligence (BI) applications. As a result Data Mining models can be used as support of knowledge-based management. The main goal of the paper is to present new features of the Oracle platform in building and testing DM models. Authors characterize methods of building and testing Data Mining models available on the Oracle 9i platform, stressing the critical steps of the whole process and presenting examples of practical usage of DM models. Verification techniques of the generated knowledge bases are discussed in the mentioned environment.

Keywords: data mining, cases modelling, Business Intelligence, data mining utilisation

Introduction

Oracle company introduces their product Oracle 9i as a complete e-Business Intelligence infrastructure. The infrastructure consists of two basic parts: Oracle 9i Database that fulfills function of Business Intelligence Data Server and Oracle 9i Application Server that run all Business Intelligence applications. Data server is equipped with Data Warehouse, OLAP Server and Data Mining Tool. Application Server gives functionality to run web portals but also offers Query and Reporting tool and BI Components that are Java Enterprise Beans designed for data analysing.

Data Mining is a quite complex issue. This is a new approach to data analysis that can enrich processes of company management. Data mining algorithms can be part of the whole Business Intelligence system.

Data Mining Process and Expectation

Application of Data Mining can bring significant benefits. The tool could be applied where there are huge sets of data, among which the valuable information is hidden (see: Chen, Han, Yu, 1997). Nearly every enterprise that functions in conditions of free and competitive market gathers information about its customers. In the bigger databases the more complex is process of data analysis. Traditional data analyzing tools gives opportunities to observe general trends and this way valuable data hidden in huge sets can be ignored. Data Mining package analyses every record and every attribute even if it consists of thousand of variables. Analysing data in such a way we can better understand our customer's behavior (see: Fayyad, 1997 or Chen, Han, Yu, 1997). That information can be essential to recommend to them products and services that match their needs best. Such an activity leads to increase turnovers with customers and level of their confidence to a company. A company can use discovered information also to acquire new customers and to reduce costs of such acquisition (see: Holsheimer, Siebes, 1994). Several new algorithms are introduced to Data Mining therefore we can find knowledge necessary to match our individual customer needs. We can predict customer future behavior as well as anticipate his needs.

Components of Oracle Data Mining

Oracle Data Mining consists of two basic elements:

- Data Mining Server – is database server that stores input data, results of process and provides metadata repository,
- Data Mining API – component that allows developers to create applications that are able to data mine. This is a set of Java classes with implemented methods of building data mining models, testing and applying them.

The second release of Oracle 9i Server is equipped with ODM tool that is much more developed than previous version. It has several new capabilities. The number of data mining algorithms has risen.

Now Oracle Data Mining has implemented such functions as:

- Classification - this problem is very complicated and data mining is one of possible methods of solving it. It appears when we have many records described by several attributes and we want to classify records to a few cases. All attributes of describing record are predictors while one target attribute plays a classifying role. Classification is useful in customer's segmentation, business modeling, credit analysis and many other applications. The classification task begins with analyzing data for which class assignments are known. This means that we have to possess database with record already classified. On this basis we can build classification model – the set of rules that describes when a given record belongs to given class and with what confidence.
- Clustering – this function is a little bit similar problem to classification. It also consists in dividing records into several groups. The difference is that we have no natural grouping in clustering problem. We use this method to find if natural grouping may exists. A cluster is a collection of objects that are similar to each other. Clustering models are different from predictive model. In this case we don't need to have classified database. Clustering models determine clusters itself, and do not need a target attribute.
- Association Rules - this solution is often associated with market basket analysis. It is used to find correlation and relationships among attributes describing records in database. It is useful in discovering business trends by analyzing customer transactions. However, they can also be used effectively to predict Web page accesses for personalization. In this case data mining model is a set of rules describing which attribute implies another on what confidence and support.
    - **Support:** Support of a rule is a measure of how frequently the items involved in it occur together. Using probability notation, support (A = > B) = P(A, B).
    - **Confidence:** Confidence of a rule is the conditional probability of B given A; confidence (A => B) = P (B | A), which is equal to P(A, B) or P(A).
- Attribute Importance - this task is also known as a feature selection. It helps in building classification models because it examine predictive attributes and eliminate redundant, irrelevant, or uninformative ones. According to its assumptions we can identify attributes that have greatest influence on a target attribute. The attribute importance model helps in building classification model. It decreases time of

building such model by indicating a subset of attributes most important in determining target attribute.

The presented functions refer to different DM tasks and require specialised algorithms. Some of these tasks have to be performed using more than one algorithm, for example:

- ***Classification*** through: Adaptive Bayes Network (see: Yarmus, 2003), Naive Bayes or Model Seeker,
- ***Clustering*** through: *k*-Means or O-Cluster (see: Milenova, Campos, 2003),
- ***Attribute importance*** through: Predictive variance
- ***Association rules*** through: Apriori (see: Park, Chen, Yu, 1995)

One may expect the list of algorithms will be extended soon (see: Han, 1996). The detailed description of the algorithms can be found: Oracle9i, 2001.

Data Mining Infrastructure

General Assumptions of Mining Data

Every process of data mining should begin with preparation of data. There are two ways of formatting table:

- Transactional - in this data format each table should be consisted of attributes describing: Sequence identifier, Attribute name and Attribute value. In transactional data format each row describe only one attribute of given case. This case is identified by integer value representing sequence identifier.
- Non-Transactional - in the transactional format whole case is described only by one row. Table consists of key column identifying whole case and set of columns representing values of attributes.

The particular attribute values incorporating into tables can appear in discretized form (binned - as a numerical value) or undiscretized as a string for both data formats.

The proper Data Mining process consists of the following tasks:

- DM Building Model
- Verification of the built DM model – internally consisting of: testing and computing lift steps,
- Scoring a DM model.

Not every data mining function needs all of these tasks. All of them should be done during classification. Clustering requires the first and the third one (Building Model and Scoring). Association Rules and Attribute Importance consist of DM Building Model. In this paper we focus on the first of the mentioned tasks. Additionally we try to apply external verification techniques used for knowledgebases generated by classification algorithms apart of suggested in the Oracle environment (see respectively: Owoc and Galant, 1998 or Oracle 9ia, 2002).

Steps of Building Model

Building Data Mining model in Oracle 9i we have to specify which function we want to use or which of available algorithms can be applied. When function settings do not contain ODM algorithm specification, the package chooses itself appropriate algorithms and provides defaults for the relevant parameters. A model is built and stored on a Data Mining Server. The main steps of the task are as follows:

1. Describing input data by building a "**PhysicalDataSpecification**" object that stores information of location of input data and its type.
2. Creating object "**MiningFunctionSettings"** that specifies values of parameters of chosen algorithm.
3. Create a "**LogicalDataSpecification**" and associate this specification with the mining function Settings.
4. Create a build task and invoke the execute method.

This scenario can be applied for different cases leading to elaboration of DM models useful for the defined functions. The indirect components created during building model (e.g. PhysicalDataSpecification, MiningFunctionSettings and LogicalDataSpecification are temporally stored in DM resources.

Examples of Data Mining Models

There are some examples of data and programs included in ODM that performs specified steps and build data mining models. There are also sets of example data that can be used to test ODM tool.

- The first example refers to building Data Mining model for classification using data stored as **CENSUS_2D_BUILD_BINNED** table in ODT_MTR schema (delivered by Oracle Corporation – see: Oracle9ia, 2001). This table stores records characterizing customers of a bank. Data describe the following attribute set: AGE, WORKCLASS, WEIGHT, EDUCATION_NUM, MARTIAL STATUS, OCCUPATION, RELATIONSHIP, RACE, SEX, CAPITAL GAIN, CAPITAL LOST, HOURS PER WEEK and NATIVE COUNTRY. All these attribute are predictive. In addition one can define the target attribute – in our model named "**Class**" and it takes value "1" for each customer that was interested in a new offer of bank – credit, and "0" for customers not interested in it.

Using Adaptive Bayes Network algorithm we can receive such a Data Mining Model. Before building a model we should set values of parameters. We can define amount of classes and a Cost Matrix. It represents costs of a mistake done during scoring data. The mistake is defined as function between actual class and predicted one. Table 1 shows the default cost matrix.

		predicted	
		0	1
actual	0	0	2
	1	1	0

It means that when we predict our customer to be interested in new offer, and he is not our cost is 2. But reverse situation, when we predict a customer to be not interested in our offer and in fact he is – it means that our cost is 1 - we don't take a chance of gaining a customer. In reality, cost matrix should be parameterized individually for each single case. Essentially, proportion between costs should be established on grounds of real costs. Sometimes gaining wrong customer can be much more expensive than losing the right one. On the other hand cost of attempt of getting customer that is not interested can be incomparably smaller than the cost of losing the right one. Model built with default cost matrix is presented in Table 2.

If we find that default cost matrix do not fit to our real cost we can modify it. Model received with that cost matrix should be more sensitive on classifying cases to 0, because wrong classification of this type is very expensive. This case is presented in Table 3.

Before scoring data with received model we should test it on data taken from reality. The result of such testing can be defined as a factor called accuracy that shows similarity of predicted values with actual ones. Accuracy can help us with selection of best data mining model to score data.

If RELATIONSHIP in (1, 5) then CLASS equal (0)
If RELATIONSHIP in (2, 3, 4, 6) then CLASS equal (1)
If MARITAL_STATUS in (1) then CLASS equal (0)
If MARITAL_STATUS in (2, 3, 4, 5, 6) then CLASS equal (1)
If AGE in (10, 4, 5, 6) then CLASS equal (0)
If AGE in (1, 2, 3, 7, 8, 9) then CLASS equal (1)
If HOURS_PER_WEEK in (4, 5) then CLASS equal (0)
If HOURS_PER_WEEK in (1, 2, 3) then CLASS equal (1)
If OCCUPATION in (1, 3) then CLASS equal (0)
If OCCUPATION in (2, 4, 5, 6) then CLASS equal (1)
If EDUCATION in (3, 4) then CLASS equal (0)
If EDUCATION in (1, 2, 5, 6) then CLASS equal (1)
If CAPITAL_GAIN in (10, 2, 3) then CLASS equal (0)
If CAPITAL_GAIN in (1) then CLASS equal (1)
If SEX in (1, 2) then CLASS equal (1)
If RACE in (1, 2, 3, 4, 5) then CLASS equal (1)
If CAPITAL_LOSS in (4, 5, 6, 7, 8) then CLASS equal (0)
If CAPITAL_LOSS in (1, 10, 2, 3) then CLASS equal (1)

Table 2: Set of rules describing dependencies between attributes' values and class

If RELATIONSHIP in (1, 2, 3, 4, 5, 6) then CLASS equal (1)
If MARITAL_STATUS in (1, 2, 3, 4, 5, 6) then CLASS equal (1)
If AGE in (1, 10, 2, 3, 4, 5, 6, 7, 8, 9) then CLASS equal (1)
If HOURS_PER_WEEK in (1, 2, 3, 4, 5) then CLASS equal (1)
If OCCUPATION in (1, 2, 3, 4, 5, 6) then CLASS equal (1)
If EDUCATION in (1, 2, 3, 4, 5, 6) then CLASS equal (1)
If CAPITAL_GAIN in (10, 2, 3) then CLASS equal (0)
If CAPITAL_GAIN in (1) then CLASS equal (1)
If SEX in (1, 2) then CLASS equal (1)
If RACE in (1, 2, 3, 4, 5) then CLASS equal (1)
If CAPITAL_LOSS in (7, 8) then CLASS equal (0)
If CAPITAL_LOSS in (1, 10, 2, 3, 4, 5, 6) then CLASS equal (1)

Table 3: Set of rules describing dependencies between attributes' values and class after modification

The last step of the procedure is applying data mining model to data that we want to be classified. As a result of this step we obtain assigning each of the existing cases to the proper class.

- The second example comes also from Oracle Corporation. This time we build Data Mining model using data stored in the **MARKET_BASKET_2D_BINNED** table in ODM_MTR schema. The goal is searching for association rules on the basis of delivering data.

This table stores information of content of a market basket. Goods in a supermarket are divided into the following categories: CANNED_GOODS, TV_DINNER, BEER, CANDY, BUTCHER_SHOP, DAIRY_GOODS, SEAFOOD, WINE and SODA.

A Data Mining model built by Associate Rules algorithm is a set of rules describing dependencies of specified attributes' values. We present small part of such a model in Table 4.

The model we received as an effect of associate rules algorithm is final result of the data mining process. It can be used by analytics to create marketing strategy or just to make a better goods arrangement in a supermarket.

```
Getting top 5 rules for model: Sample_AR_Model sorted by support.
    Rule 50: If TV_DINNER=1 then CANNED_GOODS=1 [support: 0.173, confidence: 0.57284766]
    Rule 108: If CANNED_GOODS=1 then TV_DINNER=1 [support: 0.173, confidence: 0.5709571]
    Rule 54: If BEER=1 then TV_DINNER=1 [support: 0.17, confidence: 0.5802048]
    Rule 4: If TV_DINNER=1 then BEER=1 [support: 0.17, confidence: 0.5629139]
    Rule 90: If BEER=1 then CANNED_GOODS=1 [support: 0.167, confidence: 0.5699659]
    Rule 89: If CANNED_GOODS=1 then BEER=1 [support: 0.167, confidence: 0.5511551]
```

Table 4: Set of rules describing dependencies between goods put into a market basket sorted by support.

Verification of Knowledgebases Generated by Data Mining Algorithms

General Notes on Knowledgebase Verification

In general, a set of rules as elaborated a knowledgebase should be checked applying some specific criteria: completeness and consistency for example. Usually this procedure is termed as knowledgebase verification and this is a part of more global process – knowledge validation (see: Nguyen, 1987). Primarily, all verification techniques have been developed for knowledge bases, created in the more natural manner (mostly with employing a knowledge acquisition phase). However these techniques seems to be useful also for a set of rules generated by data mining algorithms because of the following reasons (see: Owoc and Galant, 1998):

1) generated knowledgebases are used for classification tasks in the same purpose, as in the case of other expert systems, which use domain knowledge (concordance of the goals),
2) reasoning techniques, employed during the classification process, are very similar to heuristic rules obtained from experts (act procedures concordance),
3) knowledge bases to be generated, are created as a consequence of machine learning procedures and can be later expressed as one of the common accepted knowledge formalisms (knowledge representations concordance).

It does not mean fully substitution of both sorts of knowledgebases; on the contrary - some specific features of generated rules can be detected. Let us have a look on the procedures and approaches applied during the knowledge verification process.

There are several methods developed for verification of knowledge bases represented as rule sets: *tree-based approach, decision-tables approach and others* (for instance, see: Suh and Murray, 1994 and Cragun and Steudel, 1987). Applying one of them, we first need to transform rule sets into a form accepted by the method. Thus, we check initially rule set completeness (e.g. unreferenced and illegal attribute values, unreachable conclusions and so-called “dead-end” conditions or goals) and then we verify its consistency, searching: redundant, subsumed, conflicting and circular rules or unnecessary IF conditions.

Verification of Rules Completeness and Consistency

According to the method pointed out earlier, we have transformed rule sets into expanded decision tables (EDT). Every table row represents exactly one attribute in such a way, that condition states and actions can been derived from the successive rules.

It is evident from Table 5 that it is impossible to check knowledge represented in the rule set using classical techniques. In fact, values of attributes are symbolic (transformed from the real data sets) therefore

Condition Subjects	Condition States		
Relationship in(1,2,3,4,5,6)	*Yes*	*No*	*?*
Marital status in (1,2,3,4,5,6)	*Yes*	*No*	*?*
Age in(1,10,2,3,4,5,6,7,8,9)	*Yes*	*No*	*?*
Hours per week in(1,2,3,4,5)	*Yes*	*No*	*?*
Occupation in(1,2,3,4,5,6)	*Yes*	*No*	*?*
Education in(1,2,3,4,5,6)	*Yes*	*No*	*?*
Capital gain in(1,2,3,4,5,6)	*Yes*	*No*	*?*
Sex in(1,2)	*Yes*	*No*	*?*
Race in(1,2,3,4,5)	*Yes*	*No*	*?*
Capital loss in(7,8)	*Yes*	*No*	*?*
Action Subjects	Action Values		
Client interested in an offer	*X*		
Client not interested in an offer		*X*	
Unknown client reaction			*?*

Table 5: EDT for the rules describing "dependencies between attribute's vales and client behaviour".

"deep" knowledge must be discovered in order to verify its completeness (values 1,2, ...). The same we have to verify rules consistency. Every attribute value can be enough to classify the particular case as a result of no "tacit" interrelations among the chosen values.

Concerning the second introduced example we are not able to verify rules completeness and consistency. Table 6 illustrates every case stored on Data Mining Server and each of them is presented as one row of the table. Beside of ambiguous of attribute's location (each of them can be condition as well as action we are confused about checking rule set consistency. The bottom lines of Table 6 include specific "measures" of rule support and confidence.

Both of described cases don't confirm usability of common applied knowledge verification techniques. Therefore we are enforced to apply verification procedures included to ODM.

Internal Verification of Rules Accuracy

Verification of model's accuracy and calculating lifts are the last steps before its scoring. Using another

Condition Subjects	Condition States					
TV dinner	*1*			*1*		
Canned goods=1		*1*				*1*
Beer=1			*1*		*1*	
Action Subjects	Action Values					
TV dinner		*X*	*X*			
Canned goods	*X*				*X*	
Beer				*X*		*X*
Support	0,173	0,173	0,17	0,17	0, 167	0,167
Confidence	0,573	0,573	0,580	0,56	0,57	0,57
Rule No.	50	108	54	4	90	89

Table 6: EDT for the rules describing dependencies between goods put into market basket

part of real, classified data, process scores the model and compares results generated by an algorithm with the cases obtained in reality. The effect of this step is establishing an of accuracy value as a number between 0 and 1 and a confusion matrix. Accuracy is a quotient of number of well-classified records and number of all records. Confusion matrix provides an understanding of model accuracy. It shows types of errors model discovered during scoring records. Usually it has a form of a table showing dependencies between actual and predicted class as a number of records (see Fig. 1).

As we see the second example has a lower accuracy what means that this model generates more errors. It is worth to notify that number of errors with higher cost has decreased in comparison with the first example. Decision which model makes better scoring depends of analysis considering also business logic of whole data mining process.

Example 1
Accuracy : 0.7565947
Confusion matrix

		predicted	
		0	1
actual	0	464	165
	1	38	167

Example 2
Accuracy : 0.73021585
Confusion matrix

		predicted	
		0	1
actual	0	430	199
	1	26	179

Figure 1: Results of two examples of models testing.

Conclusion

In this paper, we have demonstrated results from the experiments focused on data mining performed in the Oracle 9i environment. We have tried to evaluate chosen properties of ODM. The basic contributions of the research are:

1. Rule sets generated by the package are in low numbers, usually less than 10. All rules represent rather "simple" knowledge discovered from the input data sets. It is assumed, that every rule concludes with prescribing new examples to one from the defined classes. These properties of the rule sets are essential during the verification procedures.
2. This is very difficult to verify knowledge consistency and completeness in such environment. Specific ways of building DM model (during this phase one can decide about logic and entries of a model) cause knowledge applied in a system can be regarded as "deep" knowledge.
3. Verification techniques available in Oracle 9i environments offer very limited area of model evalution. Mostly, value of rule set accuracy is this sort can be interesting knowledge base evaluator. Classification accuracy is relatively high under conditioning the well-applied algorithms.

As future directions of the current study, we will be interested in: using another method of DM Modelling for real cases. It should be valuable extension of actual research initiated recently by authors (see: Hauke, Owoc, Pondel, 2002).

References

Chen M., Han J., Yu P.S. (1997). *Data Mining: An Overview from Database Perspective*, Elect.Eng.Department, National Taiwan University Taipei, Taiwan, ROC.

Cragun B.J, Steudel H.J. (1987). A Decision-Table-Based Processor for Checking of Completeness and Consistency in Rule-Based Expert Systems. *IEEE*.

Fayyad M. U., (1997). Editorial in: *Data Mining and Knowledge Discovery* (An International Journal), vol.1, issue 1, 1997

Hauke K., Owoc M.L., Pondel M. (2002). *Drazenie danych w srodowisku Oracle 9i*. [w:] Nowoczesne technologie informacyjne w zarzadzaniu, Niedzielska E., Dudycz H., Dyczkowski M.[red.]. Prace Naukowe AE nr 955, Wydawnictwo AE Wroclaw [Hauke K., Owoc M.L., Pondel M.: *Datamining in Oracle 9i Environment* [In:] Modern Information Technologies in Management. Niedzielska E., Dudycz H., Dyczkowski M. (eds.) Research Papers No 955 Wroclaw University of Economics, Wroclaw] (in Polish)

Han J. (1996). *Data mining techniques*, ACM-SIGMOD'96 Conference Tutorial, Database Systems Research Laboratory, School of Computing Science, Simon Frazer University, Canada, June.

Holsheimer M., A.P.J.M. Siebes (1994). *Data Mining: the search for knowledge in databases*, Computer Science/Department of Algorithmics and Architecture, CR-R9406

Milenova B.L., Campos M.M. (2003). O-Cluster: Scalable Clustering of Large High Dimensional Data Sets. Retrieved March 10, 2003 from: http://otn.oracle.com/products/bi/pdf/o_cluster_algorithm.pdf

Nguyen T.A. (1987). Verifying Consistency of Production Systems. *Proceedings of the Third Conference on Artificial Intelligence Applications*, Washington D.C., IEEE Computer Society Press.

Oracle9*i* (2001). *Data Mining Concepts*, Release 9.0.1, Part No. A90389-01. Oracle Corporation.

Oracle9ia (2001). *Data Mining*, Data Sheet February 2002, Oracle Corporation.

Owoc M.L., Galant V. (1998). *Validation of Rule-Based Systems Generated by Classification Algorithms*. [in:] Evolution and Challenges in System Development J. Zupancic, W. Wojtkowski, W.G. Wojtkowski, S. Wrycza (red). Proceed. of the Information Systems Development Conference, Bled'98 (Slovenia). Kluwer Academic/Plenum Publishers, New York

Park J.S., Chen M.S.,Yu P.S. (1995). *Efficient Parallel Data Mining for Association Rules*, Proceedings of the 4th International Conference on Information and Knowledge Management, 29.11-3.12.

Suh Y.H., Murray T.J. (1994). *A Tree-Based Approach for Verifying Completeness and Consistency in Rule-Based Systems*. Expert Systems with Applications, Vol. 7, No.2. Amsterdam, The Netherlands

Yarmus J.S (2003). *ABN: A Fast, Greedy Bayesian Network Classifier*. Retrieved March 10, 2003 from: *(http://otn.oracle.com/products/bi/pdf/adaptive_bayes_net.pdf*

Biographies

Hauke Krzysztof, Ph.D. is a lecturer in the Faculty of Management and Computer Science Wroclaw University of Economics, Poland. Dr Krzysztof Hauke has authored almost 40 publications mostly oriented on databases and intelligent systems topics. His scientific interests concentrate on expert systems, knowledge discovery problems, computer supported learning, databases, temporal databases, object databases. In recent years he has been engaged in the international projects: *"Knowledge Acquisition and Intelligent Distributed Learning in Resolving Managerial Issues"* with Belgium

Owoc Mieczyslaw L., Ph.D. Eng. is a lecturer in the Faculty of Management and Computer Science Wroclaw University of Economics, Poland. Dr Owoc has authored almost 100 publications mostly oriented on databases and intelligent systems topics. In recent years he has been engaged in the international projects: *"Knowledge Acquisition and Intelligent Distributed Learning in Resolving Managerial Issues"* with Belgium and *"Machine Learning 2000"* with Sweden and Latvia. His current research is in modern information technologies including distance learning and knowledge management with focus on knowledge acquisition and validation.

Pondel Maciej, MSc. Eng. is a Ph.D. student in the Faculty of Management and Computer Science Wroclaw University of Economics, Poland. M. Pondel has authored almost 10 publications oriented on databases, data warehouses, analytical systems and intelligent mechanisms supporting management of a company. His current research is in information and knowledge discovery for usage of intelligent enterprise.