Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2005 Proceedings

Americas Conference on Information Systems (AMCIS)

2005

A Conceptual Model of Recommender System for Algorithm Selection

Mujtaba Ahsan
University of Wisconsin - Milwaukee, mahsan@uwm.edu

Lin Ngo-Ye
University of Wisconsin - Milwaukee, linye@uwm.edu

A Conceptual Model of Recommender System for Algorithm Selection

Mujtaba Ahsan
University of Wisconsin-Milwaukee
mahsan@uwm.edu

Lin Ngo-Ye
University of Wisconsin - Milwaukee
linye@uwm.edu

ABSTRACT

Classifier selection process implies mastering a lot of background information on the dataset, the model and the algorithms in question. We suggest that a recommender system can reduce this effort by registering background information and the knowledge of the expert. In this study we propose such a system and take a first look on how it can be done. We compare various classifiers against different datasets and then come up with the most appropriate classifier for a particular dataset based on its unique characteristic.

Keywords:

Data mining, algorithm selection, UCI, Weka

INTRODUCTION

Data mining is one of the important techniques of information technology that can assist management decisions via the discovery of patterns in large databases (Bigus, 1996; Chen, Han and Yu, 1996; Fayyad, Piatetsky-Shapiro and Smyth, 1996; Fayyad and Stolorz, 1997). It is an important tool that marketers can rely on to reveal patterns in databases while emphasizing the marketing one-to-one strategy (Pitta 1998). In the past few years various areas of business have also witnessed the increased use of data mining. Some examples of these are the personal bankruptcy prediction (Donato, Schryver, Kinkel, Schmoyer, Leuze and Grandy, 1999), the hotel data mart (Sung and Sang, 1998), and the customer service support (Hui and Jha, 2000). Adriaans and Zantinge (1996) and Bigus (1996) also provide a fundamental concept for the utilization of data mining in business problems covering marketing segmentation, customer ranking, real estate pricing, sales forecasting, customer profiling, and prediction of bid behavior of pilots. With the growing popularity of data mining as indicated by literature above, it is important that critical aspects of the data mining process are highlighted so that organizations become aware of them.

One such critical aspect of the data mining process is the model and algorithm selection. In order to undertake a data mining process a data analyst has to first select an appropriate model and algorithm. This selection is probably one of the most difficult problems in data mining since there is no model or algorithm that is better than all others independently of the particular problem characteristic (Aha, 1992; Salzberg, 1991; Shavlik, Mooney and Towell, 1991; Weis and Kapouleas, 1989.)

Each algorithm has a certain distinct advantage (Brodley 1995), i.e. the algorithm in question, under certain conditions or for specific types of problems is better than the rest. This happens because every algorithm has an "inductive bias" (Mitchel 1997) caused by the assumptions it makes in order to generalize from the training data to the unknown examples. Hence, the analyst must posses a lot of experience to be able to identify the most appropriate algorithm for the morphology of the problem at hand.

Another important function of data mining is the production of a model. A model can be descriptive or predictive. A descriptive model helps in understanding underlying processes or behavior. For example, an association model describes consumer behavior. A predictive model is an equation or set of rules that makes it possible to predict an unseen or unmeasured value (the dependent variable) from other, known values (independent variables). The form of the equation or rules is suggested by mining data collected from the process under study. Some training or estimation technique is used to estimate the parameters of the equation or rules. The process of selecting the appropriate models and algorithms is described in detail by Brodley and Smyth (1997).

The model of an algorithm actually defines the "search space" or "hypothesis space", such as k-DNF or k-CNF forms, linear discriminant functions, rules etc. The algorithm searches this space for the hypothesis that better fits to the data. The algorithm determines the order of visiting the states in this space. For example, two algorithms that both start their search in

DNF space, one might start the search from DNF forms that contain the complete set of features, while the other might start from sets consisting of only one feature (Gordon and desJardin, 1995).

The wrong choice of algorithm may result in a slow convergence towards the right hypothesis, or may even end at a suboptimal solution due to a local minimum. Also wrong choice of model can have a more very negative impact like a hypothesis appropriate for the problem at hand being ignored because it is not contained in the model's search space (Gordon and desJardin, 1995). In this paper, we look at the classifier selection process based on the morphology and special characteristic of the problem at hand.

LITERATURE REVIEW

The knowledge discovery process is an iterative one, as depicted in Fig. 1 (details explained on p.5). The analyst must first select the right model for the classification task to be performed and, within it, the right algorithm, whereby the special morphological characteristics of the problem must always be taken into account. The algorithm is then invoked and its output is evaluated. If the evaluation results are poor, the process is repeated from a previous state with new selections. The trial-and-error procedure is apparent in this model and errors could occur under this process.

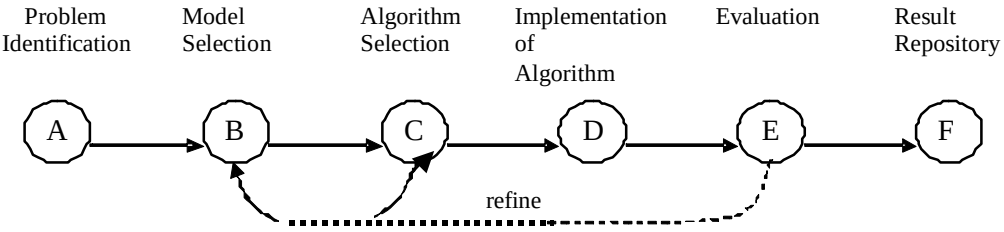

Figure 1: Knowledge Discovery Process

Hence, researchers can come up with a number of studies to facilitate the selection of the appropriate model and algorithm. Rendell, Seshu and Tcheng (1987) proposed an architecture that tries to predict which of the available algorithms will perform better for a given classification problem on hand. It uses problem characteristics such as the number of examples and the number of features. The system produces new knowledge for each new problem it faces, by associating problem characteristics with algorithm performance. The main disadvantage of this system is that it is trained as new classification tasks are presented to it; this makes it quite slow. Moreover, only a single performance criterion i.e. execution time is being considered. Schaffer (1993) proposed a brute force method for selecting the appropriate algorithm. In this all available algorithms are executed for the problem at hand and their accuracy is estimated using cross validation. The one achieving the highest score is selected. The disadvantage of this method is the high demand on computation resources, which can be a problem to SME's.

A later study done Provost and Buchanan (1995) stated the problem of model and algorithm selection as a search in the meta-space of possible models of representation and in the meta-spaces of the possible ways to traverse each one of the possible models. Movements in those meta-spaces are performed by the dedicated operators implemented in the system. This system does not produce new knowledge and can only utilize existing knowledge that can be given in the form of preconditions in the operators. The analyst must encode this knowledge explicitly. Further Brodley (1995) proposed that the selection of models and algorithms from a pool of available ones be performed on the basis of existing knowledge from the expert, encoded in the form of rules. However, this method is inflexible as the encoded rules are incorporated into the system and cannot be extended.

An innovative approach was proposed by Gama and Brazdil (1995) in their article, which was aimed at the automatic derivation of rules to guide algorithm selection. The approach is based on the characteristics of the data. They define a set of characteristics that are expected to affect the performance of the algorithms. Then, they invoke machine learning techniques to create models that associate the characteristics with the accuracy of the algorithms. The main advantage of this approach is the automated procedure of producing new knowledge on the expected performance of each new algorithm. The limitations of this are that the method has only been used in a limited number of problems and the accuracy is the sole performance measure used.

When multiple performance criteria are desirable, the problem of mapping the performance results into comparable scalar values becomes apparent. The "Data Envelopment Analysis" (DEA) methodology (Nakhaeizadeh and Schnabl, 1997) has

been proposed as a solution to this problem. In this study, positive performance metrics like accuracy, and negative ones, like training time, are combined into a single performance ration, called "efficiency". A common weighting scheme is not required; it is computed on the basis of any among the algorithms under inspection. Despite the intuition that such a weighting scheme would be biased, it is shown that an objective comparison of the algorithms is possible (Nakhaeizadeh and Schnabl, 1997). The limitation of this study is that it does not propose a methodology of using the comparison results to select the most appropriate algorithm for a new problem.

From the reviewed literature we see that no one approach is the "best" and each has its pros and cons. Based on the literature we have developed a conceptual model by which the data mining process can be improved and this is explained in detail in the following section.

METHODOLOGY

The system that we are proposing is based on the Fig. 1 and Fig 2. The model is selected (B) based on the morphology of problem at hand (A) and the characteristics of the dataset (e.g. # of instances). The data analyst then chooses an algorithm (C) that they feel best suits the dataset and proceed to invoke (D) the algorithm. The results are then evaluated (E) and the result is then stored in the repository (F). If the model and algorithm selected are not accurate the process is repeated with another model and algorithm, this iterative cycle is carried on until a model and algorithm is found that gives accurate result and has low error rate. Once the analyst is satisfied with the selection the results are stored in the repository (F). The entire process is highly iterative to ensure that only reliable results are stored in the repository. The depth of information and the breadth of datasets included in the repository are continuously increasing each time the data mining process is run on a new dataset.

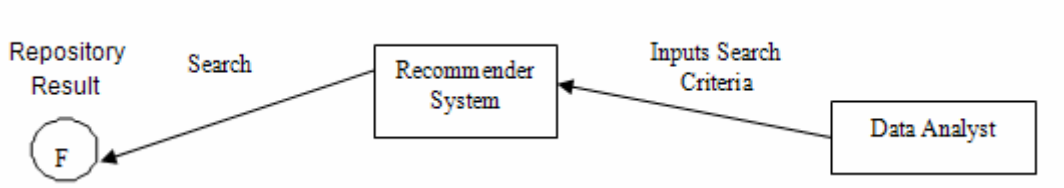

Figure. 2: Recommender System

When a new dataset is to be analyzed the data analyst can input the new dataset characteristics along with other relevant information such as type of model desired etc. into the recommender system. The recommender system then scans through the results stored in the repository (F) and comes up with a model and algorithm that best suits the new dataset and the morphology of problem at hand. The use of recommender system in model and algorithm selection could lead to both time and cost saving. Using recommender system could also potentially help in reducing the errors of selecting an incorrect model or/and algorithm. The limitation of this method is that it might not be useful unless a large number of results already exist in the repository (F). Also its applicability might be more relevant to organizations that utilize data mining process frequently. We conducted some experiments to test for the accuracy of various classifiers on different sets and have described the experiments in detail in the next section.

EXPERIMENTS

For the purpose of initial study we selected six different UCI datasets whose characteristics are shown in the Table 1 and focus our study on algorithm selection. In order to compare the performances of the various algorithms, we performed experiments on a collection of six data sets from the UCI Repository. We selected the six data sets based on the following criteria

1. Number of classes (i.e. two-class or multiple-class)

2. Types of features (i.e. Nominal, Numeric, or both)

No.	Name	#Classes	#Instances	#Features		
				Nominal	Numeric	Total
1	Audiology	24	226	69	0	69
2	Automobile	7	205	10	16	26
3	Glass Identification	7	214	0	9	9
4	Horse Colic	2	368	15	7	22
5	Ionosphere	2	351	0	34	34
6	Breast Cancer	2	286	9	0	9

Table 1: Characteristics of 6 UCI Data Sets

The experiments were run utilizing the Weka data mining tool (Witten and Frank, 2000). The next step was the algorithm selection; the selection process was limited to the ones that were available in the Weka suite.

The following algorithms were in the experiments: the tree learning algorithm J48, which is a re-implementation of C4.5, the k-nearest neighbor (IBk) algorithm, the naive Bayes (NB) algorithm, the neural network algorithm and the ZeroR algorithm. Also three algorithms for combining classifiers, namely bagging, boosting and stacking were included. The performance of each of these algorithms is assessed in terms of its accuracy and error rate. In all experiments, classification errors are estimated using 10-fold stratified cross validation. Cross validation is repeated ten times using different random generator seeds resulting in ten different sets of folds (Kohavi, 1995).

A brief description of the different classifiers and the cross-validation procedure used in the experiments is as follows:

ZeroR

ZeroR is the most primitive learner. It simply predicts the majority class in the training data if the class is categorical and the average class value if it numeric. Although it makes little sense to use this scheme for prediction, in our case it is used to serve as baseline performance benchmark.

IBk(K-Nearest Neighbor)

K-NN model falls within the general category of "instance-based" (versus "memory-based") technique's where all the data needs to be explicitly remembered. The non-trivial computation is performed in the prediction time, this behavior differs from conventional learner algorithm, in which "training" occurs between the reception of data and prediction. So the complexity grows significantly (non-linear) when the data set is growing larger for IBk.

J48

J48 is the decision tree classification algorithm. It builds a decision tree model by analyzing training data, and uses this model to classify user data. This is the Weka application of C4.5 decision tree learner introduced by Quinlan for inducing classification models or decision trees for the data. C4.5 is an extension of ID3 where it accounts for unavailable values, continuous attribute value range, pruning of decision trees, rule deviation and so on.

Naïve Bayes (NB)

NaiveBayes implements the probabilistic Naïve Bayesian classifier. This method is based on bayes's rule of conditional probability, which is a simple relationship between probability of a hypothesis and an evidence which bear on that hypothesis. It naively assumes all data to be independent. It is valid to multiply probabilities only when the events are independent.

Neural Networks (NN)

A Classifier that uses backpropagation to classify instances. This network can be built by hand, created by an algorithm or both. The network can also be monitored and modified during training time. The nodes in this network are all sigmoid (except for when the class is numeric in which case the output nodes become unthresholded linear units). In the experiments We used the default parameter settings.

Stacking

This is way of combining different classification models. A meta-learning model is fitted at a level1, which evaluates the level-0 fitted models using a cross-validation approach on each level-0 model.

The stacking format that we used in the experiments is

'-B \"weka.classifiers.trees.j48.J48 -C 0.25 -M 2\" -B \"weka.classifiers.lazy.IBk -K 1 -W 0\" -B \"weka.classifiers.bayes.NaiveBayes \" -B weka.classifiers.rules.ZeroR -X 10 -S 1 -M \"weka.classifiers.functions.Logistic -P 1.0E-13 -R 1.0E-8 -M 200\"'

Bagging

This uses random sampling with replacement in order to obtain different versions of a given data set. The size of each sampled data set equals the size of the original data set. On each of these versions of the data set the same learning algorithm, J4.8 in our case, is applied.

Boosting

This first builds a classifier with some learning algorithm (again J4.8 is our case) from the original data set. The weights of the misclassified examples are then increased and another classifier is built using the same learning algorithm. The procedure is repeated several times. The AdaBoost.M1 variant of boosting was used in our experiments.

Cross Validation

Selecting the best algorithm for a data set in the absence of prior knowledge is a search problem. One approach is to use cross-validation (Linhard and Zucchini, 1986; Kohavi, 1995). This is a method of estimating the accuracy of a classification or regression model. The data set is divided into several parts, with each part in turn used to test a model fitted to the remaining parts. Given a set of data, a *n*-fold cross-validation splits the data into *n* equal parts. Each candidate algorithm is run *n* times; for each run, $n - 1$ parts of the data are used to form a classifier, which is then evaluated using the remaining part. The results of the *n* runs are averaged and the algorithm that produced classifiers with the highest average classification accuracy is selected. Shaffer (1993) applied this idea to selecting a classification algorithm. The results of an empirical comparison of a cross-validation method (CV) to each algorithm considered by CV, illustrated that on average, across the test-suite of domains, CV performed best.

RESULT ANALYSIS

Datasets	NN	Bagging	AdaBoost	Stacking	J48	IBk	NaiveBayes	ZeroR
Audiology	83.2964	81.2866	84.747	48.5277	77.2648	78.4308	72.6383	25.2115
Glass	66.7814	73.9048	75.1515	15.0152	67.6255	69.9502	49.4459	35.513
Automobile	76.5643	81.4762	85.4571	29.3357	81.7667	74.5524	57.4143	32.7024
Breast Cancer	68.2734	73.1022	66.8879	71.8313	74.2808	72.8461	72.697	70.2956
Ionosphere	91.3127	91.6262	93.0476	92.1651	89.7444	87.0984	82.1675	64.1032
Horse Colic	80.6809	84.994	81.6299	84.6359	85.1554	79.1066	78.6997	63.0481

Table 2: Classifiers Accuracy

The results of the various experiments were analyzed on two aspects

a. Accuracy

Accuracy is an important factor in assessing the success of data mining. When applied to data, accuracy refers to the rate of correct values in the data. When applied to models, accuracy refers to the degree of fit between the model and the data. This measures how error-free the model's predictions are. Since accuracy does not include cost information, it is possible for a less accurate model to be more cost-effective. These results can be seen in the above Table 2.

b. Confusion Matrix

A confusion matrix shows the counts of the actual versus predicted class values. It shows not only how well the model predicts, but also presents the details needed to see exactly where things may have gone wrong. Based on confusion matrix, we can calculate true positive, false positive, true negative and false negative (Sinha and May, 2005). These results can be seen in the Table 3 below.

An analysis of the accuracy results of the various datasets gives us the following information.

Multi-Class Datasets

For the Audiology data set all the 7 algorithms selected performed better than the base algorithm (ZeroR). The algorithm AdaBoost was found to be the most accurate performer followed closely by NN. For the Glass Identification dataset all the algorithms except for Stacking performed better than the base algorithm (ZeroR). The algorithm AdaBoost was again found to be the most accurate performer followed by Bagging. For the Automobile data set again all the algorithms except for Stacking performed better than ZeroR our base algorithm. Once again AdaBoost was found to be the most accurate performer followed by J48 and Bagging respectively.

Two-Class Datasets

For the Breast cancer dataset, the NN and AdaBoost algorithms performed worse than our base algorithm ZeroR. The most accurate performer was found to be J48 followed by Bagging and IBk respectively. For the Ionosphere dataset all the algorithms performed better than our base algorithm ZeroR. AdaBoost was found to be the most accurate algorithm, followed closely by Stacking and Bagging respectively. For the Horse Colic dataset again all the algorithms performed better than our base algorithm ZeroR. For this dataset J48 was found to be the most accurate algorithm followed closely by Bagging and Stacking respectively.

The confusion matrixes were built using the True Positive Rate, False Positive Rate, True Negative Rate and False Negative Rate. This shows us how well the model predicted and from the results we can see exactly where things may have gone wrong. Apart from getting to know the counts of the actual versus predicted class rates, the confusion-matrixes are also useful in understanding the rate of Type I and Type II errors for the various algorithms and datasets.

In summary we can tentatively conclude from the tabulated accuracy and confusion-matrixes results that for multi-class nominal dataset AdaBoost should be the most preferred algorithm and Stacking should be the least preferred algorithm. From our analysis we can state that for the multi-class numeric dataset AdaBoost should be the most preferred algorithm and Staking should be the least preferred algorithm. Similarly, for the multi-class mixed dataset AdaBoost should be the most preferred algorithm and Staking should be the least preferred algorithm. For all multi-class dataset the AdaBoost algorithm is consistently the best performing algorithm and staking is consistently the worst performing algorithm.

From our analysis of the two-class nominal dataset, we can see that J48 should be the most preferred algorithm and AdaBoost should be the least preferred algorithm. For the two-class numeric dataset AdaBoost should be the preferred algorithm and Naïve Bayes should be the least preferred algorithm. The analysis for the two-class mixed dataset shows that J48 is the best performing algorithm and Naïve Bayes should be the least preferred algorithm. In the case of two-class dataset no one algorithm is consistently the best or worst performer, but from the results we see that that J48 performs well for the majority of the tested two-class dataset and Naïve Bayes algorithm performs poorly for most two-class datasets.

Dataset	Classifier	TP	FP	TN	FN
Audiology	Neural Network	0	0.002233	0.997767	0.1
	Bagging	0	0	1	0.1
	AdaBoost	0	0.001324	0.998676	0.1
	Stacking	0	0.005316	0.994684	0.1
	J48	0	0	1	0.1
	IBK	0	0.000434	0.999565	0.1
	NaiveBayes	0	0	1	0.1
	ZeroR	0	0	1	0.1
Glass	Neural Network	0.777143	0.215333	0.784667	0.222857
	Bagging	0.774286	0.13381	0.86619	0.225714
	AdaBoost	0.79	0.13419	0.86581	0.21
	Stacking	0.117143	0.079524	0.920476	0.882857
	J48	0.714286	0.159048	0.840952	0.285714
	IBK	0.752857	0.156048	0.843952	0.247143
	NaiveBayes	0.744286	0.405238	0.594762	0.255714
	ZeroR	0	0	1	1
Automobile	Neural Network	0	0	1	0
	Bagging	0	0	1	0
	AdaBoost	0	0	1	0
	Stacking	0	0.2	0.98	0
	J48	0	0	1	0
	IBK	0	0	1	0
	NaiveBayes	0	0	1	0
	ZeroR	0	0	1	0
Breast Cancer	Neural Network	0.7926	0.5758	0.4242	0.2074
	Bagging	0.929	0.7368	0.2632	0.071
	AdaBoost	0.789	0.615	0.385	0.211
	Stacking	0.8938	0.6961	0.3039	0.1062
	J48	0.9473	0.74	0.26	0.0527
	IBK	0.8935	0.6613	0.3388	0.1065
	NaiveBayes	0.851	0.5663	0.4338	0.149
	ZeroR	1	1	0	0
Ionosphere	Neural Network	0.7958	0.0213	0.9787	0.2042
	Bagging	0.8103	0.0244	0.9756	0.1897
	AdaBoost	0.8544	0.027	0.973	0.1456
	Stacking	0.8406	0.0328	0.9672	0.1594
	J48	0.8207	0.0596	0.9404	0.1793
	IBK	0.6888	0.0272	0.9728	0.3112
	NaiveBayes	0.8646	0.2025	0.7975	0.1354
	ZeroR	0	0	1	1
Horse Colic	Neural Network	0.8517	0.2701	0.7299	0.1483
	Bagging	0.9259	0.2797	0.7203	0.0741
	AdaBoost	0.8649	0.2667	0.7333	0.1351
	Stacking	0.9189	0.2777	0.7223	0.0811
	J48	0.9307	0.2835	0.7165	0.0693
	IBK	0.8321	0.2791	0.7209	0.1679
	NaiveBayes	0.8013	0.2375	0.7625	0.1987
	ZeroR	1	1	0	0

Table3: True Positive, False Positive, True Negative, and False Negative

CONCLUSION

From a theoretical point of view, we tackled the problem of identifying the best algorithm given the dataset characteristics and utilizing a recommender system to perform the selection process. Results indicated that AdaBoost is a relative stable performer compared to other algorithms. The results of this study also indicate that characteristic of datasets has an influence on algorithm performance.

From a managerial point of view, this study shows that the performance of algorithm varies with the dataset and understanding this is critical for selecting the correct model and algorithm. It also highlights the importance of the model and algorithm selection process and illustrates how a recommender system can lead to both cost and time saving by potentially reducing errors in model and algorithm selection.

Our conceptual presentation of how a recommender system works shows well how managers/companies can benefit by employing this technique. For example, let us consider that we have a new dataset that is two-class nominal type dataset. When the data analyst inputs these attributes into the search criteria, the recommender system would search through our result repository and find that J48 was the best performing algorithm for this type of dataset. The recommender system would then recommend that we use a decision tree model and algorithm J48 in particular (Please note that we tested only the J48 algorithm from the many decision tree model algorithms available. There exist a possibility that another algorithm in the decision tree model could have performed better than J48).

An important limitation of our study is related to the dataset selection, i.e. the types of datasets that we selected were limited. This study has to be replicated with a larger sample of datasets to come up with a more generalizable result. Another limitation of this study is that the choice of models and algorithms was limited by Weka. Further work needs to be done with other models and algorithms in order to generalize the findings of this study. Also the results were compared only on the basis of accuracy and error rate, inclusion of other comparative measures, such as ROC, training time, testing time etc. can lead to the reliability of the findings. Finally, the use of recommender system is presented only at a conceptual level; future work should try to develop a prototype to test the use of recommender system for data mining purposes.

ACKNOWLEDGMENTS

We like to sincerely thank Atish Sinha and Humin Zhao for their helpful suggestions.

REFERENCES

1. Adriaans, P. and Zantinge, D. (1996) Data mining, Reading Mass.: Addison-Wesley, Chapter 6
2. Aha, D.W. (1992) Generalizing from case studies: A case study. In *9th Int. Machine Learning Conf.*
3. Bigus, J.P. (1996) Data mining with neural networks, New York: McGraw-Hill, pp. 131-177
4. Brodley, C.E. (1995) Recursive automatic bias selection for classifier construction. *Machine Learning*, 20:63-94
5. Brodley, C.E. and Smyth, P. (1997) Applying classification algorithms in practice. *Statistics and Computing*
6. Chen, M.S., Han, J. and Yu, P.S. (1996) "Data mining: an overview from a database perspective", IEEE Transactions on Knowledge and Data Engineering, Vol. 8, pp.866-883
7. Donato, J.M., Schryver, J.C., Kinkel, G.C., Schmoyer Jr., R. L., Leuze, M.R. and Grandy, N. W. (1999) "Mining multi-dimensional data for decision support", Future Generation Computer Systems, Vol. 15, No. 3, pp. 433-441
8. Fayyad, U. M, Piatetsky-Shapiro, G. and Smyth, P. (1996) From data mining to knowledge discovery in databases, AI Magazine, Vol. 17, pp.37-54
9. Fayyad, U. and Stolorz, P.(1997) Data mining and KDD: promise and challenge, Future Generation Computer Systems, Vol. 13, No. 2-3, pp.99-115
10. Gama, J. and Brazdil, P. (1995) Characterization of classification algorithms. In Pinto Fereira, C. and Mamede, N. editors, *Progress in AO, 7th Portuguese Conf. in AI (EPIA'(%)*, pages 83-102. Springer Verlag
11. Gordon, F. and DesJardin, M. (1995) Evaluation and selection of biases. Machine Learning, 20:5-22 (http://www.ai.univie.ac.at/oefai/ml/metal/metal-theoretical.html)
12. Hui, S.C. and Jha, G. (2000) "Data mining for customer service support", Information and Management, Vol.38, No. 1, pp. 1-13

13. Kohavi, R. (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In C.S. Mellish (ed.), *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann, pp. 1137-1143.
14. Kohavi, R. (1996) Data mining using MLC++, a machine learning library in C++. In *Tools with AI*
15. Linhard, H. and Zucchini, W. (1986) *Model Selection.* NY: Wiley
16. Mitchel, T. (1997) *Machine Learning*. MacGraw Hill
17. Nakhaeizadeh, G. and Schnable, A. (1997) Development of multi-criteria metrics for the evaluation of data mining algorithms. In *KDD'97*, pages 37-42. AAAI Press
18. Pitta, D. (1998) "Marketing on-to-one and its dependence on knowledge discovery in databases", Journal of consumer marketing, Vol. 15, No.5, pp. 468-480
19. Provost, F.J. and Buchanan, B.G. (1995) Inductive policy: The pragmatics of bias selection. *Machine Learning*, 20:35-61, 1995.
20. Rendell, L.A., Seshu, R.M., and Tcheng, D.K. (1987) Layered concept learning and dynamically variable bias management. In *10th Int. Joint Conf. on Artificial Intelligence*, pages 308-314
21. Shavlik, J.W., Mooney, R. and Towell, G. (1991) Symbolic and neural computation: An experimental approach. *Machine learning*, 6:111-114
22. Salzberg, S. A. (1991) Nearest hyper-rectangle learning method. *Machine Learning*, 6:251-276
23. Schaffer, C. (1993) Selecting a classification method by cross-validation. *Machine Learning*, 13:135-143
24. Schaffer, C. (1993) Selecting a classification method by cross-validation. *Preliminary Papers of the Fourth International Workshop on Artificial Intelligence and Statistics* (pp. 15-25)
25. Sinha, A.P., and May, J.H. (2005) Evaluating and tuning predictive data mining models using receiver operating characteristic curves. *Journal of Management Information Systems*, 21, 3, 249-280.
26. Sung, H.H. and Sang, C.P.(1998) "Application of data mining tools to hotel data mart on the Intranet for database marketing", Expert Systems With Applications, Vol. 15, No. 1, pp. 1-31
27. Weiss, S.M. and Kapouleas, I. (1989). An empirical comparison of pattern recognition, neural nets and machine learning classification methods. In *11th Int. Joint Conf. in Artificial Intelligence*
28. Witten, I.H., and Frank, E. (2000) Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation. San Francisco, CA: Morgan Kaufmann.