Development of Multi-Criteria Metrics for Evaluation of Data Mining Algorithms

Gholamreza Nakhaeizadeh
Daimler-Benz AG, Research and Technology F3S/E,
P. O. Box 23 60, D-89013 Ulm, Germany
nakhaeizadeh@dbag.ulm.DaimlerBenz.COM

Alexander Schnabl
Technical University Vienna, Department of
Econometrics, Operations Research and Systems Theory,
Argentinierstr. 8/1, A-1040 Vienna, Austria
e9025473@fbma.tuwien.ac.at

Abstract

The main aim of this paper is to suggest multi-criteria based metrics that can be used as comparators for an objective evaluation of data mining algorithms (DM-algorithms). Each DM-algorithm is characterized, generally, by some positive and negative properties, when it is applied to certain domains. Examples of properties are the accuracy rate, understandability, interpretability of the generated results and stability. Space and time complexity and maintenance costs can be considered as negative properties. By now there is no methodology to consider all of these properties, simultaneously, and use them for a comprehensive evaluation of DM-algorithms. Most of available studies in literature use only the accuracy rate as a unique criterion to compare the performance of DM-algorithms and ignore the other properties. Our suggested approach, however, can take into account all available positive and negative characteristics of DM-algorithms and can combine them to construct a unique evaluation metric. This new approach is based on DEA (Data Envelopment Analysis). We have applied this approach to evaluate 23 DM-algorithms in 22 domains. The results are analyzed and compared with the results of alternative approaches.

Introduction

Knowledge Discovery in Databases (KDD) is a process that aims at finding valid, useful, novel and understandable patterns in data (Fayyad et al. 1996). The core of this process consists of the application of various *Data Mining algorithms (DM-algorithms)* based on statistical approaches, Machine Learning, Neural Networks etc. One of the essential issues in both the development and application phases of DM-algorithms is, however, the lack of objective metrics making a fair evaluation of the algorithms possible. In our opinion:

1. Such metrics, on one hand, should take into account not only positive properties (advantages) but also negative characteristics (disadvantages) of DM-algorithms. Only in this case is a fair evaluation possible. In buying a car e.g. people consider not only positive points like safety, comfort, quality and after sales service, but also the negative points like high price, high fuel consumption, high repair cost, environmental issues etc. We appeal for consistency in applying the same philosophy in the evaluation of DM-algorithms.
2. On the other hand, a fast and comprehensive evaluation of various algorithms is then possible, when a unique metric is available that can reflect objectivity and not in ad-hoc manner all known positive and negative properties of algorithms. In the above mentioned example, it would be a significant help for car buyers, if they can use a unique metric to conclude that car A is superior to car B, considering all known positive and negative properties of the cars.

The main aim of this paper is that to suggest metrics for the evaluation of Data Mining algorithms that cover both points 1 and 2 above. The rest of the paper is organized as follows: In section 2, we critically review the available evaluation criteria in the literature and discuss the need for developing multi-criteria based evaluation metrics reflecting all the available positive and negative properties of DM-algorithms. In section 3, we suggest a new evaluation approach based on the concept *of Data Envelopment Analysis (DEA)* that, in our opinion, covers both requirements 1 and 2. In section 4, we use this approach and evaluate the algorithms that have been involved in the project StatLog (Michie et al. 1994). The last section is devoted to discussions, conclusions and suggestions for further research.

Available Evaluation Criteria

The definition of the KDD-process given by Fayyad et al. (1996) considers a lot of positive properties for the patterns one obtains at the end of a KDD-Process. The patterns should be new, valid, understandable and usable. This leads to definition of *interestingness* (see also Hausdorf and Müller 1996, Silberschatz and Tuzhilin 1995). Such characteristics can be used for the evaluation of DM-algorithms which are used to obtain the patterns. For example, algorithm A would be superior to algorithm B if it leads to more understandable or more valid patterns. If such characteristics can be described in a measurable metric then they can be used for an objective evaluation of DM-algorithms.

One important problem in dealing with such properties is that by now most of them are not measurable. For example, novelty or usefulness are only subjective and can not be measured. In dealing with *complexity* one can argue that the number and the length of the extracted rules might be a measure for complexity (see also Klinkenberg and Clair 1996). This argument is, however, only valid, if one compares two or more rule based DM-algorithms. This approach can not be used to compare a rule-based algorithm e.g. CN2 (Clark and Niblett 1989) with an algorithm based on linear discrimination or neural networks. Measuring of *understandability* or *interpretability* is more difficult because in this case the domain-specific background knowledge should be available. Only in the light of this background expertise can the results be interpreted (Bratko 1995).

Specifically dealing with *validity* in one dimension there are, however, reasonable criteria like predictive accuracy rate, the cost of misclassification (Knoll et al. 1993), robustness (Hsu et al. 1995), generalization and domain coverage (Klinkenberg and Clair 1996) and stability (Turney 1995) that can be used as objective measurable metrics to evaluate DM-algorithms[1]. Such criteria are applicable, however, only to the DM-algorithms based on supervised learning. In the case of unsupervised learning, it is not easy to measure the validity (see for example the discussion on validity of cluster analysis algorithms in Gordon 1996).

As already mentioned, to perform a fair comparison of the alternative DM-algorithms, one should take into account not only positive but also negative properties. To negative properties belong e.g. high complexity, high cost of misclassification, high training and testing time, high inference time and high maintenance costs. It is true that in some available contributions in the literature the authors measure such negative properties as well and discuss them (see Michie et al. 1994, Klinkenberg and Clair 1996) but by now there is no comprehensive metric available for evaluation of DM-algorithms reflecting all known positive and negative properties of such algorithms, when they are applied to different domains.

In the next section we introduce a multi-criteria based metric that overcomes this shortcoming and can be used for an objective ranking of alternative DM-algorithms. Our approach uses the DEA concept developed originally by *Operations Research* Community to measure technical efficiency.

DEA-Based Evaluation Metrics

Main Idea

The main idea of DEA is due to Charnes et al. (1978). More recently, however, it has been further developed in different directions and applied to different domains (see Emrouznejad et al. 1996, for a comprehensive bibliography on DEA). It is not the aim of this paper to discuss the different versions of DEA. Our aim is to discuss the main idea and to explain how this idea can be used to develop evaluation metrics for ranking alternative DM-algorithms.

Originally, DEA tries to develop a ranking system for *Decision Making Units (DMUs)*. In our case, each DMU is a DM-algorithm. In DEA terminology, positive properties are called *output components* and negative properties *input components*. In our terminology, a typical example for an output component is the accuracy rate produced by a supervised DM-algorithm. A typical input component is the computation time that the DM-algorithm needs for training. Generally, output components are all components where higher values are better and input components are those where lower values are better. Using these components, we can now define the *efficiency* of a DM-algorithm as follows:

$$efficiency = \frac{\sum weighted\ output\ components}{\sum weighted\ input\ components}$$

As the above relation shows, the definition of efficiency covers all positive and negative characteristics of a DM-algorithm and efficiency in this form can be regarded as a multi-criteria based metric that can be used for the evaluation of DM-algorithms. In our opinion, efficiency as defined above is more general as *interestingness* defined by Fayyad et al. (1996) that covers only the positive properties of DM-algorithms. Due to the fact that the values of input and output components are given, only the values of the weights are necessary to determine the value of efficiency. Precisely in this point, namely in determining the weights, DEA differs from the alternative approaches, for example, from the *point awarding* method. By using point awarding, one can award to the accuracy rate (which is a positive property for DM-algorithms) a certain number of points, say 10, to each percentage accuracy rate which exceeds a threshold value (e.g. the accuracy rate of the naive predictor). This means that if the accuracy rate is three percent better than the naive predictor then the algorithm will be awarded 30 points. Awarding negative properties (e.g. time consume or space complexity) is done in the same way. The total achieved *score* of each DM-algorithm is determined by summing the points across different attributes. Such scores can be used now for ranking the DM-algorithms, the higher the score the higher the rank. This approach suffers, however, from some drawbacks. Specifically, it suffers from the

[1] It should be mentioned that in some cases is necessary to define standards. For example we need to standardize what is meant by max. memory. A linear discrimination algorithm (LDA) as implemented in SAS may need different memory as LDA implemented by SPSS, though the accuracy rate will be identical. This idea was suggested during a useful discussion with Charles Taylor.

subjective opinion of decision makers who determine the required awarded points. It might be e.g. that an attribute with the same importance is awarded different points.

In many cases, it is very difficult or impossible to award or calculate objective weights. If the weights are the corresponding prices of a unit of each input and output component (this would be a natural way to determine the values of input and output and calculate the efficiency), then who can say e.g. how much cost a unit of the accuracy rate or a unit of the rule complexity? Normally, such prices are unknown.

DEA evades the ad-hoc judgments described above. In DEA the awarded points (weights) are determined for each DM-algorithm individually during the computation by maximizing the efficiency in the following way. Suppose that we are evaluating n algorithms with p input and q output components and for the algorithm k let:

I_{kx} = amount of input x;
O_{kx} = amount of output y;
u_{kx} = weight for input x;
v_{ky} = weight for output y.

Denoting the efficiency of the DM-algorithm k by R_k now leads to:

$$R_k = \frac{\sum_{y=1}^{q} v_{ky}\, O_{ky}}{\sum_{x=1}^{p} u_{kx}\, I_{kx}} \qquad (1)$$

DEA chooses the weights so that the efficiency of the algorithm k is as close to 100% as possible (see Doyle and Green 1991 for more detail). At the same time, the other algorithms should not have an efficiency more than 100% using *the same particular weights*. Obviously this is an optimization problem that can be transformed to the following *linear program (LP)*:

Select values of $u_{k1}, u_{k2},\ldots, u_{kp}$ and $v_{k1}, v_{k2},\ldots, v_{kq}$ by maximizing R_k in the relation (1) subject to:

$$R_i = \frac{\sum_{y=1}^{q} v_{ky}\, O_{iy}}{\sum_{x=1}^{p} u_{kx}\, I_{ix}} \le 1 \qquad (2)$$

for $i = 1, 2,\ldots, k,\ldots, n$, $u_{kx} \ge 0$ and $v_{ky} \ge 0$ for all x and y.

If algorithm k does not achieve the given threshold (100%), then it is not efficient and there is at least one algorithm among the others dominating it. There are different ways for setting the weights in (2). The most used ones are the *input-oriented* and the *output-oriented* optimization (Ali and Seiford 1993). The goal in input-oriented optimization approach is to reduce *radially* the input-levels as much as possible. Radially means that all component-values are changed by the same proportion. Conversely in output-oriented optimization approach the main goal is to enlarge radially the output-levels as much as possible. Keeping the input as constant, the above LP is transformed to maximizing of:

$$R_k = \sum_{y=1}^{q} v_{ky}\, O_{ky}$$

subject to: $\sum_{x=1}^{p} u_{kx}\, I_{kx} = 1$ and (3)

$$\sum_{y=1}^{q} v_{ky}\, O_{iy} - \sum_{x=1}^{p} u_{kx}\, I_{ix} \le 0$$

for $i = 1, 2,\ldots, k,\ldots, n$, $u_{kx} \ge 0$ and $v_{ky} \ge 0$ for all k, x and y which can be solved for each algorithm using the *Simplex Method.*

After solving this LP and determining the weights, the algorithms with $R_k = 1$ (100%) are *efficient* algorithms and form the *efficiency frontier* or *envelope*. The other algorithms do not belong to the efficiency frontier and remain outside of it. As already mentioned, the definition of *efficiency* is more general than *interestingness* as suggested by Fayyad et. al. (1996). One can connect also both concepts in this form that more efficient algorithms are more interesting. For ranking the algorithms, one can use the approach suggested by Andersen and Petersen (1993) (AP-model). They use a criterion that we call it the AP-value. In input-oriented models the AP-value measures how much an efficient algorithm can radially enlarge its input-levels while remaining still efficient (output-oriented is analogous). For example, for an input-oriented method an AP-value equal to 1.5 means that the algorithm remains still efficient when its input values are all enlarged by 50%. If the algorithm is inefficient then the AP-value is equal to the efficiency value.

To explain the above approach, we present the following simple example.

Example

Suppose that we have four DM-algorithms *A, B, C* and *D* with one positive property (output) and two negative properties (input) given in Table 1. Figure 1 shows these data graphically in the input-space.

Comparision unit	A	B	C	D
Input 1	2.00	5.00	10.00	8.00
Input 2	8.00	5.00	4.00	6.00
Output	1.00	1.00	1.00	1.00

Table 1: The data of the example. Selection output values equal to 1 is just for an easier interpretation.

DEA creates an efficiency frontier (bold lines), which is convex in the input-space. The efficiency frontier is formed in Figure 1 by the algorithms *A*, *B* and *C*. Their efficiency values are equal to 1. In the input-space, *D* lies outside the efficiency frontier. If *D* could use, however, the input values of its corresponding projection *D**, then it would lie on the efficiency frontier. Graphical representations like Figure 1 are only possible for two or three input components. Using LP (3) leads to the solution given in Table 2.

We can see in Table 2 that DEA classifies all efficient algorithms *A*, *B* and *C* with the efficiency value of 1. The inefficient algorithm *D* gets the efficiency value 0.79 which is just the ratio of *OD** to *OD*. It has to reduce the input values by 21% to become efficient. In this case the values for input 1 and input 2 would be 6.32 and 4.47, respectively, which correspond to the coordinates of point *D**.

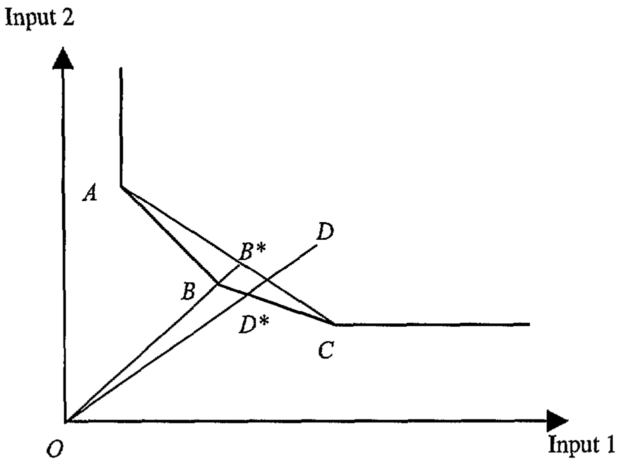

Figure 1: Efficiency frontier for DM-algorithms *A*, *B* and *C* (bold lines). Algorithm *D* is inefficient. To become efficient, this algorithm has to reduce both input values until it reaches the input values of the point *D**. Algorithm *B* which is efficient could use the input-levels of point *B** and would still be efficient.

Algorithm	Efficiency value	AP-value	Rank
A	1.00	2.50	1
B	1.00	1.20	3
C	1.00	1.25	2
D	0.79	0.79	4

Table 2: The solution of the DEA-algorithm for example

As mentioned before, to rank efficient algorithms, we use the AP-model. Algorithm *B* e.g. has an AP-value equal to 1.20, i.e. it can enlarge the input-levels by 20% remaining still efficient. Graphically we get this value from *OB**/OB.

Empirical Analysis

Multi-Criteria Ranking Results

The most comprehensive evaluation of DM-algorithms known to us is the study of Michie, Spiegelhalter and Taylor (MST 1994). They compare the performance of 23 classification algorithms on 22 different domains. To rank different algorithms, when they are applied to a certain domain, MST use only one property namely the accuracy rate for the test data set although they have data about maximum computer storage, training and testing time, training and testing error rates and training and testing misclassification costs (where the costs are available). As an example, the results reported by MST for the *Credit Management Data set* is presented in Table 3. Their ranking for the algorithms is given in the last column. The notation „*“ is used for missing (or not applicable) information, and „FD“ is used to indicate that an algorithm failed on this data set.

To obtain a DEA-based ranking for the algorithms applied by MST we have used the input and output-oriented versions of DEA described above. In the following, these versions are denoted by I and O, respectively. To rank the algorithms, we have used the AP-model. We have used three input components (max. storage, training time and testing time) and one output component (accuracy rate defined as 1 - error rate for the test data set). As an alternative, we have also used the version with an additional output component (accuracy rate for training data set). Input oriented versions are denoted by 4I (one output and three input components) and by 5I (two output and three input components). The same is valid for output-oriented versions denoted by 4O and 5O.

Algorithm	Max. Storage	Training Time (sec.)	Testing Time (sec.)	Training Error Rates	Testing Error Rates	Rank
Discrim	68	32.2	3.8	0.031	0.033	13
Quadisc	71	67.2	12.5	0.051	0.050	21
Logdisc	889	165.6	14.2	0.031	0.030	8
SMART	412	27930.0	5.4	0.021	0.020	1
ALLOC80	220	22069.7	*	0.033	0.031	10
k-NN	108	124187.0	968.0	0.028	0.088	22
CASTLE	48	370.1	81.4	0.051	0.047	19
CART	FD	FD	FD	FD	FD	-
IndCART	1656	423.1	415.7	0.010	0.025	6
NewID	104	3035.0	2.0	0.000	0.033	13
AC^2	7250	5418.0	3607.0	0.000	0.030	8
Baytree	1368	53.1	3.3	0.002	0.028	7
NaiveBay	956	24.3	2.8	0.041	0.043	16
CN2	2100	2638.0	9.5	0.000	0.032	12
C4.5	620	171.0	158.0	0.014	0.022	3
ITrule	377	4470.0	1.9	0.041	0.046	18
Cal5	167	553.0	7.2	0.018	0.023	4
Kohonen	715	*	*	0.037	0.043	16
DIPOL92	218	2340.0	57.8	0.020	0.020	1
Backprop	148	5950.0	3.0	0.020	0.023	4
RBF	253	435.0	26.0	0.033	0.031	10
LVQ	476	2127.0	52.9	0.024	0.040	15

Table 3: Results for Credit Management data set (2 classes, 7 attributes, 20 000 observations) p. 133, using 22 algorithms

The DEA-ranking results for the Credit Management data set are given in Table 4. We have omitted algorithms *CART* and *Kohonen* from further analysis because as Table 3 shows there is not always enough information about input and output components for these algorithms.

We can see from Table 4 that the MST-ranking results based only on one comparison criterion differs, generally, from our results which are based on multi-criteria metrics. For example, algorithm *NaiveBayes* is ranked by MST as 16th. Our ranking using different versions of DEA varies, however, between 7 and 10. The reason is that the relatively low accuracy rate of this algorithm (1 - 0.043 = 0.957) is compensated by low training and testing time. This is the same for *NewID* and *ITrule*. For these algorithms low accuracy rate is adjusted by low max. storage and low testing time. On the other hand, MST rank *DIPOL92* as the first. In 3 of the versions of DEA, it is ranked, however, between 6 and 9. In this case, the high accuracy rate is compensated by high training time. Considering *Quadisc*, we can see that it is ranked by MST as 21st. Our results show that this algorithm is not efficient at all. It has got a rank between 12 and 18 using different versions of DEA. It seems that in this case the low accuracy rate can not be adjusted by the other components. There are some cases for which the MST-ranking does not differ radically from our ranking. For example, *SMART* has got the first rank by MST and a ranking between 2 and 5 in our analysis. If we examine the input and output components of this algorithm, we can see that with the exception of the training time the other values are relatively good and apparently the high training time can not obscure, significantly, the positive effect of the other components.

Algorithm	MST	5I	4I	5O	4O
Discrim	13	*3*	*3*	*9*	*7*
Quadisc	21	14	12	18	17
Logdisc	8	16	13	15	12
SMART	1	*2*	*2*	*5*	*5*
k-NN	22	15	14	19	19
CASTLE	19	*6*	*9*	*9*	*7*
IndCART	6	17	17	14	13
NewID	13	*1*	*8*	*9*	*7*
AC^2	8	*9*	19	*7*	16
Baytree	7	*9*	*5*	*1*	*1*
NaiveBay	16	*7*	*10*	*9*	*7*
CN2	12	*9*	16	*8*	15
C4.5	3	*9*	*4*	*4*	*4*
ITrule	18	*8*	*11*	*9*	*7*
Cal5	4	*4*	*6*	*2*	*2*
DIPOL92	1	*9*	*1*	*6*	*6*
Backprop	4	*5*	*7*	*3*	*3*
RBF	10	18	15	16	14
LVQ	15	19	18	17	18

Table 4: Ranking algorithms for the Credit Management data set using MST and different DEA-models (italic figures mean efficient)

Concerning the different versions of DEA, we can see from Table 4 that for Credit Management data set each algorithm which is efficient in 4I is also efficient in 5I, but not all efficient algorithms in 5I are efficient in 4I. The same is valid for 4O and 5O. The more components are included into the DEA-model the more algorithms are classified as efficient. Our further examinations have shown that this is not valid, generally. Using different DEA-versions (as it was expected) does not lead to exactly the same but to similar rankings. Exceptions are *DIPOL92* in 4I, *NewID*, *Baytree* and *C4.5* in 5I and *AC^2*, *CN2* which in some cases are not efficient at all.

We have done this sort of analysis for all 22 data sets, but we can not report the whole results here. More details can be found in Jammernegg et al. (1997). To make an additional comparison, we report, however, in Table 5 the top 5 algorithms selected by DEA-model 4I for each data set and compare our results with those of MST reported in p. 185 of their book.

Data set	*First*	*Second*	*Third*	*Fourth*	*Fifth*
Credman	DIPOL92	SMART	Discrim	C4.5	Baytree
Craust	NewID	NaiveBay	ITrule	C4.5	DIPOL92
Dig44	Cascade	Quadisc	DIPOL92	Discrim	NaiveBay
KL	Backprop	LVQ	Cascade	Discrim	DIPOL92
Vehicle	DIPOL92	ALLOC80	SMART	NaiveBay	CART
Letter	LVQ	NewID	Baytree	NaiveBay	Discrim
Chrom	k-NN	DIPOL92	CASTLE	NaiveBay	NewID
SatIm	LVQ	C4.5	Discrim	NaiveBay	Baytree
Segm	Baytree	k-NN	C4.5	SMART	CART
Cut20	NewID	k-NN	Discrim	Backprop	LVQ
Cut50	Baytree	NewID	Discrim	Backprop	DIPOL92
Head	Discrim	CASTLE	Cascade	CART	Baytree
Heart	SMART	IndCART	NewID	Baytree	DIPOL92
CrGer	k-NN	Baytree	Cal5	DIPOL92	CART
Shuttle	CART	CASTLE	Cal5	Backprop	Baytree
Diab	DIPOL92	RBF	k-NN	Baytree	Cal5
DNA	C4.5	CASTLE	NewID	Discrim	Backprop
Tech	Baytree	k-NN	Discrim	NaiveBay	Cal5
Belg	*Logdisc*	k-NN	DIPOL92	NewID	Backprop
BelgII	Baytree	NewID	CASTLE	Backprop	Cal5
Faults	DIPOL92	NaiveBay	ITrule	CART	Discrim
Tsetse	Baytree	NewID	Discrim	NaiveBay	Cal5

Table 5: The Top 5 algorithms of the DEA-model 4I for each data set

As it was expected, our results based on the DEA-version 4I differ, generally, from the results of MST based on only one comparison criterion namely the accuracy rate of the test data set. For example, in our results *DIPOL92* has the first rank for four data sets. In MST results for none. On the other hand, *KNN* is selected as the best algorithm by MST for four data sets. In our results only for two.

Discussion and Conclusions

As we mentioned before, the main idea of DEA is extended in different directions. These extensions can handle some of the limitations of the basic versions of DEA. To such extensions belong dealing with non-discretionary inputs and outputs, handling of categorical inputs and outputs and dealing with flexible weights (see Charnes et al. 1996, Chapter 3).

DEA-models were originally developed in the Operations Research Community and are used to rank DMUs in different domains (see e.g. Paradi et al. 1995 and Emrouznejad and Thanassoulis 1996). In this paper we have shown that such models can be used effectively to rank DM-algorithms and provide a fair evaluation. This enables the KDD-community to have a better understanding of the real performance of the developed DM-algorithms. As discussed in section 2, the number of measurable input and output components characterizing the positive and negative properties of the DM-algorithms is at present too low. We have shown in this paper that even in this situation using the DEA-based multi-criteria metrics is more suggestive than using a single criterion i.e. the accuracy rate.

Further research can be done in different directions. First of all, the practicability of different extensions of DEA described above should be examined, when they are applied to evaluation of DM-algorithms. Secondly, the adaptive DEA-models are needed to handle the dynamic aspects (changing of inputs, outputs, preferences etc.) automatically. In the DEA-Community some efforts have be done in this direction (Färe and Grosskopf 1996, Schnabl 1996). Further basic research is still necessary in this field.

Acknowledgment

The Authors would like to thank Harald Kauderer, Werner Jammernegg, Mikuláš Luptácik, Thomas Reinartz, Charles Taylor and two anonymous reviewers for fruitful discussions and suggestions. The research of Alexander Schnabl was supported financially by Daimler-Benz AG, Research and Technology, Ulm, Germany.

References

Ali, A. I. and Seiford L. M. 1993. The Mathematical Programming Approach to Efficiency Analysis. The Measurement of Productive Efficiency. In Fried, H. O., Lovell, C. A. K. and Schmidt, S. S. eds. Techniques and Applications, 120-159, Oxford University Press.

Andersen, P. and Petersen, N. C. 1993. A Procedure for Ranking Efficient Units in Data Envelopment Analysis. *Management Science*, Vol. 39, No. 10: 1261-1264.

Bratko, I. 1995. Machine Learning: Between Accuracy and Interpretability. In Aha, D. and Riddle P. eds. Proceedings of the Workshop on Applying Machine Learning in Practice at the Twelfth International Machine Learning Conference.

Charnes, A., Cooper, W. W. and Rhodes, E. 1978. Measuring the Efficiency of Decision Making Units. *European Journal of Operational Research* 2(6): 429-444.

Charnes A., Cooper W. W., Lewin A. Y. and Seiford, L. M. 1996. *Data Envelopment Analysis: Theory, Methodology and Applications*. Kluwer Academic Publishers.

Clark, P. and Niblett, T. 1989. The CN2 induction algorithms. *Machine Learning*, 3: 261-285.

Doyle, J. R. and Green R. H. 1991. Comparing Products Using Data Envelopment Analysis. *OMEGA. International Journal of Management Science*, Vol. 19, No. 6: 631-638.

Emrouznejad, A. and Thanassoulis, E. 1996. An Extensive Bibliography of Data Envelopment Analysis (DEA). Volume I: Working Papers, Volume II: Journal Papers. Business School, University of Warwick, England.

Färe, R. and Grosskopf, S. 1996. *Intertemporal Production Frontiers with Dynamic DEA*. Kluwer Academic Publishers.

Fayyad, U. M., Piatetsky-Shapiro, G. and Smyth, P. 1996. From data mining to knowledge discovery: An overview. In Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R.. Advances in Knowledge Discovery and Data Mining. 1-30, AAAI/MIT Press.

Gordon, A. D. 1996. Cluster Validation. Paper presented at IFCS-96 Conference, Kobe, March 1996.

Hausdorf, C. and Müller, M. 1996. A Theory of Interestingness for Knowledge Discovery in Databases Exemplified in Medicine. In Lavrac, N., Keravnou, E. and Zupan, B. eds.. Proceedings of the First International Workshop on Intelligent Data Analysis in Medicine and Pharmacology. 31-36, Budapest.

Hsu, C. N. and Knoblock, C. A. 1995. Estimating the Robustness of Discovered Knowledge. In Fayyad, U. M. and Uthurusamy, R.. Proceedings of the First International Conference on Knowledge Discovery & Data Mining. 156-161, AAAI Press.

Jammernegg W., Luptácik M., Nakhaeizadeh G. and Schnabl A. 1997. Ist ein fairer Vergleich von Data Mining Algorithmen möglich? Forthcoming.

Klinkenberg, R. and Clair, D. S. 1996. Rule Set Quality Measures for Inductive Learning Algorithms. In Dagli, C. H., Akay, M., Chen, C. L., Fernandez, B. R. and Ghosh, J.. Proceedings of the Artificial Neural Networks in Engineering (ANNIE 96) Conference. 161-168, ASME Press, New York.

Knoll, U., Nakhaeizadeh, G. and Tausend, B. 1993. Cost sensitive pruning of decision trees. In Proceedings of the Eight European Conference on Machine Learning ECML94. 383-386, Berlin.

Michie, D., Spiegelhalter, D. J. and Taylor, C. C. eds. 1994. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, Chichester.

Paradi, J. C., Reese, D. N. and Rosen, D. 1995. Applications of DEA to Measure the Efficiency of Software Production at two Large Canadian Banks. *The Annals of Operations Research.*

Schnabl, A. 1996. Nichtparametrische Effizienzanalyse und technischer Fortschritt: Dynamische Data Envelopment Analysis. Master Thesis, Technical University Vienna.

Silberschatz, A. and Tuzhilin, A. 1995. On Subjective Measures of Interestingness in Knowledge Discovery. In Fayyad, U. M. and Uthurusamy, R.. Proceedings of the First International Conference on Knowledge Discovery & Data Mining. 275-281, AAAI Press.

Turney, P. 1995. Technical Note: Bias and the Quantification of Stability. *Machine Learning*, 23-33.