Automating Exploratory Data Analysis for Efficient Data Mining

Jonathan D. Becher
Accrue Software, Inc.[1]
510-580-4940
becher@accrue.com

Pavel Berkhin
Accrue Software, Inc.
510-580-4945
pavelb@accrue.com

Edmund Freeman
Accrue Software, Inc.
206-527-8114
eef@accrue.com

ABSTRACT

Having access to large data sets for the purpose of predictive data mining does not guarantee good models, even when the size of the training data is virtually unlimited. Instead, careful data preprocessing is required, including data cleansing, handling missing values, attribute representation and encoding, and generating derived attributes. In particular, the selection of the most appropriate subset of attributes to include is a critical step in building an accurate and efficient model. We describe an automated approach to the exploration, preprocessing, and selection of the optimal attribute subset whose goal is to simplify the KDD process and dramatically shorten the time to build a model. Our implementation finds inappropriate and suspicious attributes, performs target dependency analysis, determining optimal attribute encoding, generates new derived attributes, and provides a flexible approach to attribute selection. We present results generated by an industrial KDD environment called the Accrue Decision Series on several real world Web data sets.

Keywords

Attribute selection, automation, transformation, encoding.

1. INTRODUCTION

Much of the research and discussion in the Knowledge Discovery in Data (KDD) community has centered around a variety of approaches — including cross validation and bootstrapping — for handling the situation in which there is not enough data to build a good model. Although these techniques are of interest to industrial practitioners, in practice we typically have access to a nearly limitless supply of data. This is particularly true when mining Web data sources, which seem to double in size every few months.

Given this wealth of data, industrial practitioners face the opposite problem: how to effectively simplify and narrow down the scope of data used for building a model. Smaller data sets provide a number of benefits, including:

- Reduced elapsed CPU time for building a model (training and verification phases)
- Reduced elapsed CPU time for using a model (forecasting or scoring phase)
- Potentially increased model accuracy
- Increased explanatory power of the model

Given our focus on Web data mining, we are faced with a prime consideration: getting a model into production as quickly as possible. This stems from the fact that Web time is famously seven times faster than "real" time; our client's patience is typically low as their business is constantly being reinvented overnight. In response, we have attempted to automate as many steps of the KDD process as possible, from data collection, to attribution encoding, transformation and selection, through model parameter searching, to best model selection.

In this paper, we describe an approach to automating the exploratory data analysis (EDA) step of the KDD process. In particular, we focus on the problem of attribute selection, narrowing the potentially hundreds of thousands of attributes (a.k.a. variables or features) down to a manageable subset without destroying the viability of the subsequent model. Our focus stems from the facts that automated attribute selection is less well understood than case (a.k.a. record or instance) selection through sampling techniques, and that attribute selection has historically been highly labor intensive and error prone.

The techniques described in this paper are available commercially in the Accrue Decision Series [ADS], a highly scalable knowledge discovery workbench with seven separate predictive and descriptive mining engines. The Decision Series also contains techniques outside the scope of this paper for automating other steps in the KDD process, including sampling, parameter searching and model selection.

2. EXPLORATORY DATA ANALYSIS

Although exploratory data analysis (EDA) can be used as a pre-processing step for both predictive and descriptive engines, much of our focus has been on the efficient building of models using standard predictive techniques: neural networks [12,11], classification and regression decision trees [2, 25], and Bayesian learning [7]. Due to the practical limitations of commercial mining, we have tried to achieve a balance between the time spent on data exploration and the gains we get in this process. In this paper, we assume that the data consists of homogeneous cases each with fixed number of attributes**.** Furthermore, we confine ourselves to numeric attributes with float or integer values and to nominal

KDD 2000, Boston, MA USA

categorical attributes. We call the first type of attributes continuous and the second type of attributes categorical.

Our EDA approach involves a four-step process:

- Identifying inappropriate and suspicious attributes
- Selecting the most appropriate attribute representation
- Creating derived attributes
- Choosing an optimal subset of attributes

Some attributes are obviously inappropriate to be used in a particular model and can be discarded automatically. Other attributes are borderline and marked by the system as suspicious. For these suspicious attributes, the modeler may choose to manually intervene by providing additional domain specific information that cause them to be removed. Otherwise, the suspicious attributes are retained for processing in subsequent steps.

All retained attributes are processed to determine the most appropriate representation. This step handles outliers, missing values, and encoding. Continuous attributes are encoded by thresholding (a.k.a. discretizing) [8] the original values into a small number of value ranges. For categorical attributes, encoding merges several values (categories) together. This grouping is similar to a subset option in C4.5 [25]. As both thresholding and grouping can cause the fatal loss of some of the detail contained in original attributes, we try to intelligently balance this loss against the advantages of efficient encoding by using three association measures: mutual information, chi-squared Cramer's V, and Goodman-Kruskal index. These associations are measured between the source attributes and a specific target (a.k.a. dependent variable) to determine which encoding is optimal for a given model. This optimization is performed by an EDA function we call target dependency analysis (TDA).

After determining the most appropriate attribute representation, EDA attempts to create new derived attributes that may be more beneficial than the existing ones. It does so by experimenting with a variety of univariate and multivariate transformations. When all original and new derived attributes are cleansed, confirmed to be appropriate, and discretized, EDA uses two independent approaches to attribute selection [18, 13], which are both based on filter model selection [14]. Using two algorithms provides additional flexibility and increases our confidence in the results.

3. INAPPROPRIATE AND SUSPICIOUS

As previously mentioned, inappropriate attributes are automatically removed from further processing. EDA identifies following types of inappropriate attributes:

- ***Constant*** - only contains a single value;
- ***Null*** - has all Null (missing) values;
- ***Near Null*** - has a fraction of Null (missing) values larger than a specified threshold;
- ***Many Values*** - has a fraction or number of values larger than a specified threshold; These descriptive attributes are typically identifiers such as phone or social security numbers.

In contrast to inappropriate attributes, suspicious attributes warrant some skepticism about their appropriateness but are kept unless specified by the user. EDA identifies following types of suspicious attributes:

- ***Artifact*** - association or correlation with the target is greater than a specified threshold. These attributes are often unintentionally included and, without proper identification, they lead to artificially good models that do not generalize well.
- ***Poor Predictor*** - association or correlation with the target is less than a specified threshold. These attributes do not contribute much by themselves in predicting the target but combinations of these attributes may have increased predictive power. Attribute selection is the ultimate way to decide whether they are appropriate for further modeling.
- ***Near Constant*** - one value of an attribute covers more than specified fraction of all cases.
- ***Few Values*** - has less than a specified number of distinct values.
- ***Few Cases*** - has less than a specified number of distinct non-Null cases.

4. TARGET DEPENDENCY ANALYSIS

The objective of TDA is to come up with the generic measures of association between the source and target attributes, and to research the strength and variation of these associations. These associations measure the strength of dependency between attributes and provide a nonlinear generalization to the classic linear concept of correlation. TDA supplies three choices of association measures: mutual information, Cramer's V, and Goodman-Kruskal index.

Given two categorical attributes, source X with values $j=1:J$ and target Y with values $q=1:Q$, with joint distribution P_{jq}, and marginal distributions $P_{j.}$, $P_{.q}$, *mutual information* $I(X,Y)$ is defined as

$$I(X,Y) = \sum_{jq} P_{jq} \log(P_{jq} / P_{j.}P_{.q}).$$

where we use base two logarithm if the information units are bits. This measure is widely used in information theory [3] and machine learning [25]. It is sometimes referred to as information gain, due to a property that it is equal to a decrease in entropy $H(Y)-H(Y|X)$ caused by knowing X, where $H(X) = -\sum_{q} P_{.q} \log(P_{.q})$. There is also a relation between mutual information and so-called Kullback Leibler distance (KL-instance) [19]. TDA uses a normalization of mutual information $I(X,Y)/H(X)$, which is scaled to [0,1].

The second association measure based on chi-squared statistic

$$\chi^2(X,Y) = N \sum_{jq} (P_{jq} - P_{j.}P_{.q})^2 / (P_{j.}P_{.q}).$$

where N is total number of cases. A well known normalization of chi-squared statistic scaled to [0,1], which represents the strength of association is *Cramer's V* [24]

$$V(X,Y)= \chi^2(X,Y)/(N \min(Q,J)-1).$$

The third measure of association used in EDA is a *Goodman-Kruskal association index*. Consider a trivial classifier, which chooses as a forecast the most frequent target value. When X is available, we can do better, by choosing the most frequent forecast among all cases with the same X-value j (maximum likelihood forecast). The Goodman-Kruskal index is a difference in error rate between trivial and X conditioned forecasters. While mutual information and Cramer's V demonstrate a high level of consistency, the Goodman-Kruskal index has a drawback in that it can be zero for non binary target, even when other two measures are positive.

These association measures are used as the objective function in TDA to generate optimal thresholding for continuous attributes and optimal grouping for categorical attributes. This optimization must take into account both the number of groups or thresholds and their location. For a continuous attribute and a fixed number of thresholding intervals, the corresponding cut points are optimized by means of an annealing algorithm [23]. In practice, we also impose a lower bound on the number of cases per thresholding interval to ensure that the ranges are relevant. For a categorical attribute and a fixed number of groups (less than J), we experimented with two approaches for grouping of categorical values. The first one was based on clustering of j-conditional distributions of target q-values and used the traditional K-means technique to cluster J points in Q-dimensional space. We preferred a direct information based clustering, similar to K-means, but using an information based objective function rather than L_2 norm as it explicitly relates to the I(X,Y) association measure defined above.

To determine both the number of thresholding intervals for continuous attributes and also the number of groups for categorical attributes, we used a simple heuristic based on the rate of flattening of the objective function. Higher dimensions produce better results but also introduce more complexity; as long as the objective function is increasing rapidly we continue to increase the number of dimensions. We also check if adding Nulls as a separate category would more than marginally affect the objective function. Choosing the best number of intervals and groups is an area of on-going research for us.

After running TDA, we have a new set of discretized categorical attributes that result from thresholding and grouping the original attributes. As all attributes are now categorical, it is easy to compare their overall benefit to the model and rank their predictive power with respect to the target.

5. DERIVED ATTRIBUTES

When a source attribute is continuous, certain univariate transformations may increase its correlation with the target. These transformations are typically only beneficial to linear regression models (such as the regression tree technique in the Decision Series) and, as such, can be disabled when using other mining engines to save computational time. Several transformations, including quadratic function, inverse function, exponent, logarithm, power function, and square root, are tried for each continuous attribute. Some of the functions have free parameters, which are optimized to find the best value. If a given transformation increases the correlation with the target beyond a specified threshold, that transformation is retained for further processing. EDA relies on the fact that the concept of correlation can be generalized to a continuous-categorical couple [26] so that these transforms can be used regardless of whether the target is continuous or categorical.

EDA also supports exploring functions of several continuous attributes, including linear combinations with undefined coefficients, ratios and products. If one of these functions has a significantly higher correlation than each of the original attributes does, it becomes a new derived attribute. A good example of this from the financial community is the classic price/earning ratio, which is typically more predictive than the attributes price or earnings alone. Since this feature is computationally expensive, it is typically restricted to certain subsets of the original attributes.

6. ATTRIBUTE SELECTION

The goal of attribute selection is to select a subset of attributes without significantly affecting the overall quality of the resultant model. Reducing the total number of attributes used reduces computational time and memory requirements, and in many cases leads to more accurate and/or more easily explainable models. We use two independent algorithms for attribute selection to ensure the widest possible range of applicability.

The first selection algorithm is based on the concept KL-distance mentioned above and is a modification of attribute selection methodology suggested in [18]. The essence of this algorithm is the minimization of the *expectation of KL-distance* between the target distribution $P(Y=q|X_1=j_1,..X_k=j_k)$, conditioned by joint distribution of all k source attributes, and the target distribution conditioned by s selected attributes $X_1,..X_s$, $P(Y=q|X_1=j_1,..X_s=j_s)$, $s<k$, (for simplicity we assume that exactly the first s attributes are selected)

$$\delta(X_{1:k},X_{1:s}) = \sum_{j1,..,jk} P_{j1,..,jk}\, KL(P_{q|j1,..,jk} \,\|\, P_{q|j1,..,js}),$$

$$KL(P_q\|R_q) = \sum_q P_q \log (P_q/R_q).$$

To make this idea computationally feasible, the algorithm resorts to low dimensional *Markov Blankets* (MB), rather than comparing large attribute sets. The concept behind MB reflects the simple idea of information coverage. An attribute X_0 is associated with a small subset of attributes or blanket $X_{1:b}$. If $\delta(X_{0:b}, X_{1:b})$ is small, an attribute X_0 is well covered by its blanket and is a good candidate for exclusion. In practice, the implementation must address such issues as the choice of the original blankets and the exclusion criterion. The attribute with the smallest δ-measure is not necessarily the best candidate for exclusion, since it potentially could be a member of another blanket used at some previous iteration to exclude another attribute, and its exclusion could cause a chain effect. This MB algorithm belongs to the category of backward selection algorithms. Care must be taken in setting the size of the blanket as small increases in the MB dimensions can results in large increases in computational resources; however, very modest dimensions generally result in a good selection.

The second attribute selection algorithm is based on the concept of *Inconsistency Rate* (IR), which is a generalization to many attributes of the Goodman-Kruskal index described above (see also [13]). IR is the error rate of a trivial (maximum likelihood) classifier which predicts the majority target outcome on each subset $X_1=j_1,..X_k=j_k$. If the omission of a certain attribute does not affect IR, the error rate of this simple classifier

remains intact without this attribute, and the attribute is a good candidate for exclusion. We use a step-wise heuristic with a major loop of backward selection, based on the iterative exclusion of the attribute that minimally affects IR. Forward steps are used to check if a previously excluded attribute can be re-included beneficially to overall monotone sequence of IR for subsequent k,k-1,...- dimensional attribute subsets.

For both the MB and IR algorithms, the backward process stops a when user-specified minimum number of retained attributes is reached. Because all excluded attributes are ordered by the iteration count at which they have been excluded, attributes excluded at the earliest stages are eliminated until the requested number (large enough to cover minimal retained set) is reached. For example, we can request to exclude attributes with IR rates below a certain threshold. The output at each step of the δ measures for MB algorithm and the inconsistency rates for IR algorithm provides a good heuristic of what number of attributes to request.

7. EXPERIMENTAL EVALUATION

This section provides example results from using EDA on real world Web data sets. Due to confidentiality requirements of paying clients, the sources of the data have been made anonymous.

In the first example, an on-line newspaper publishing company wanted to identify repeat visitors to its site in advance of their returning. Overall the return rate was 25%; the problem is predict which visitors are most likely to return in the next 30 days from the last 90 days of Web site traffic information. The training set contained ~10,000 cases; the verification set contained 2,500 cases. More than 300 attributes were available for modeling, including recency, frequency, duration (RFD) information, browser type, referring domain and URL, and a variety of demographic data.

Out of the original 300 attributes, EDA identified more than 50 as inappropriate and removed them from subsequent processing. As expected, many of these inappropriate attributes were Null or Near Null. A significant portion of them, however, were classified as Many Values. These pseudo-identifiers had nearly a different value for every case; with such a high cardinality they provided little value to the modeling process. One common example of this phenomenon is the attribute that captures the Browser/IP pair. In fact, Browser/IP pairs serve as a visitor identifier at those sites which have no registration or cookie mechanism.

As we can see from the following output (see Figure 1), EDA also identified that the TOPLEVELDOMAIN attribute was suspicious because the value of COM covers more than the default 90% of the cases. (For those readers who may not be familiar with Internet terminology, top level domain refers to the third part of an URL, as in the com part of www.accrue.com.) The distribution of values is indeed suspicious in light of the fact that most Web sites see about 75% of their traffic come from the COM top-level domain. The open question is what the modeler should do about it. As previously mentioned, by default the Accrue Decision Series will continue to use it and ultimately decide during attribute selection whether it should be eliminated. A clue to its value is contained in the output; a significantly higher percentage than normal (89%) of the visitors from the EDU domain do not return to the Web site.

```
TOPLEVELDOMAIN
Status is Suspicious: Near Constant
Type is Categorical
Number of cases with non NULL values
is 9107 (99.3%)
Number of distinct attribute values
is 5

Values Statistics:
 No    cases:     0    1 :  Value Name
  1     8244:  0.76 0.24 :  COM
  2        1:  1.00 0.00 :  GOV
  3      193:  0.89 0.11 :  EDU
  4      152:  0.71 0.29 :  ORG
  5      517:  0.72 0.28 :  NET

Value COM covers a larger percentage
(90.5%) of cases than specified
threshold (90.0%).
```

Figure 1. Example suspicious attribute

In the following related output (see figure 2) we see three optimized encodings determined by the EDA module. The age attribute was a continuous attribute that was thresholded into five ranges using four cut points while the original 14 categorical values of the education attribute were replaced with five groups. On the other hand, EDA was not able to optimize the default encoding for the sex attribute; the two original categories (M and N) were each encoded as separate values.

```
Age threshold 23.5 27.5 35.5 61.5 ; --
4 cut points determined.

Education category { 10th 11th 12th
1st-4th 5th-6th 7th-8th 9th Preschool
} { Bachelors } { Doctorate Masters
Prof-school } { HS-grad  Some-college
} ; -- 5 groups determined.

Sex category; --  No grouping (2
original values).
```

Figure 2. Optimized encodings determined by the EDA module

In the second example, a clicks and mortar retailer wanted to optimize their direct marketing campaigns by using historic data to target consumers for upcoming campaigns. Overall the response rate from previous campaigns was 4.4% with 200,000 cases split into a 9000 case 50/50 stratified training set and a 100,000 case unstratified verification set. There were 593 source attributes available, including purchase transactions, financial information, and demographics.

A number of experiments were run to test the efficacy of the EDA attribute encoding and selection. Three base models were built using all 593 attributes marked No Selection (see table 1), using 254 attributes selected by the Markov Blanket technique, and using 16

attributes selected by the Inconsistency Rate technique. These base models used default, unoptimized encodings in which continuous attributes were thresholded using an equal frequency algorithm and categorical attribute were unchanged. From these three base models, we also constructed three corresponding models with optimized encodings, using the thresholding and grouping suggestions from EDA.

The resultant six models were compared using two separate measures, lift in the top 5% quantile and ROC. We prefer these measures to the overall error rate on the verification set as they more closely reflect the actual business goals: optimizing behavior on a subset of the population as opposed to the entire audience. From the following table, we see that halving the number of attributes significantly improves the model; in fact a further 63% reduction in number of attributes to only 16 causes almost no degradation in performance with the associated saving in elapsed time. What's more, using the optimized encoding suggested by EDA improves the model such that there is virtually no difference between using 254 attributes and using 16.

Two other results are worth pointing out. First, to test whether the increase in these measures was significant, we ran multiple models with different training and verification sets. We calculated from formulas that the top 5% lift had a standard deviation of 0.11 and the ROC metrics [20, 6] had a standard deviation of 0.0037. Second, these results were obtained using a boosted naïve bayesian classifier; a classification tree induction technique produced analogous results.

Table 1. Unoptimized Encoding vs EDA Encoding

		Unoptimized Encoding	EDA Encoding
No Selection	Attributes Used	593	593
	Training Time	151	130
	ROC	0.712	0.724
	Top 5% Lift	2.76	2.71
MB	Attributes Used	254	254
	Training Time	74	79
	ROC	0.729	0.742
	Top 5% Lift	3.28	3.39
IR	Attributes Used	16	16
	Training Time	44	40
	ROC	0.712	0.735
	Top 5% Lift	3.03	3.35

8. RELATED WORK

Data preprocessing is a standard practice in statistics [28], pattern recognition and data mining [26]. Generic data cleansing techniques are well described in [10]. Grouping of categorical values as it relates to tree induction techniques is discussed in [25] while thresholding of continuous variables is discussed in [8]. In [9] information based thresholding of continuous attributes is augmented by the use of the minimal description length principal. A comprehensive introduction to information theory is contained in [3]. For a practical study related to deriving new attributes in context of data mining, see [1]. In linear statistical modeling, many similar approaches have been used; most notably, principal component analysis [15]. The idea that EDA is inherently an iterative, interactive endeavor is advocated in [27]. While we agree with this philosophy in some aspect, our primary focus is on automatic process. Visualization environment for EDA is discussed in [4].

For attribute selection in unsupervised learning see [5]. Two models, filter and wrapper, exist for attribute selection and both are described in [14]. For an earlier work on attribute selection see [16]. The Markov Blanket attribute selection algorithm is a modification of the algorithm introduced in [18]. Inconsistency Rate, utilized in IR attribute selection, serves as an objective function in [13], where a Las Vegas type algorithm is used for the major iterative loop. For wrapper type attribute selection see [17,22].

9. CONCLUSIONS

This paper presents an approach for automating the exploratory data analysis step in the knowledge discovery in data. This EDA process identifies inappropriate and suspicious attributes, selects the most appropriate attribute representation, create univariate and multivariate derived attributes, and chooses an optimal subset of attributes to retain for the model. Using the resultant simplified attribute subset reduces elapsed CPU time for building and using a model, increases model accuracy, and improves the explanatory power of the model. In practice, these benefits are magnified several-fold because the process of building an effective model typically involves building multiple models with different parameters, training sets, or business objectives.

10. ACKNOWLEDGMENTS

Most of the work described in this paper was performed while the authors were employed by NeoVista Software, Inc. which was later acquired by Accrue Software Inc. The impetus for many of the ideas was specific needs that came up during paid client modeling projects. As mentioned above, these ideas are realized in the exploratory data analysis (EDA) model of the Accrue Decision Series. The authors would like to thank the members of the knowledge discovery engineering (KDE) organization who tested early versions of the software and made several suggestions that improved the product.

11. REFERENCES

[1] Asker, L., and Maclin, R. Feature Engineering and Classifier Selection: A Case Study in Venusian Vulcano Detection. *Machine Learning: ICML'97*, 3-11, Morgan Kaufmann, San Francisco, CA (1997).

[2] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J. *Classification and Regression Trees*. Wadsworth, Belmont, CA (1984).

[3] Cover, T.M., and Thomas J.A. *Elements of Information Theory*. John Wiley & Sons, Ney York, NY (1990).

[4] Derthick,M., Kolojejchick, J., and Roth, S.F. An Interactive Visualization Environment for Data Exploration. *KDD-97*, 2-9, AAAI Press, Menlo Park, CA (1997).

[5] Devaney, M., and Ram, A. Efficient Feature Selection in Conceptual Clustering. *Machine Learning: ICML'97*, 92-97, Morgan Kaufmann, San Francisco, CA (1997).

[6] Egan J.P. Signal Detection Theory and ROC Analysis, Series in Cognitition and Perception, Academic Press, 1975, New York, NY.

[7] Elkan C. Boosting and Naïve Bayesian Learning. *Technical Report No.CS97-557, UCSD* (Sept. 1997).

[8] Fayyad U.M., and Irani K.B. Multi-interval discretization of continuous-valued attributes for classification learning. *IJCAI' 93*, 1022-1027 (1993).

[9] Friedman, N. Discretizing Continuous Attributes While Learning Bayesian Networks. *Machine Learning: ICML'96*, 157-165, Morgan Kaufmann, San Francisco, CA (1996).

[10] Guyon, I., Matic, N., and Vapnik, V. Discovering Information Patterns and Data Cleaning. *KDD-94*, 145-166, AAAI Press, Menlo Park, CA (1994).

[11] Haykin, S. *Neural Networks. A comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ (1999).

[12] Hertz, J., Krogh, A, and Palmer, R.G. *Introduction to the theory of neural computations.* Addison-Wesley, Reading, MA (1991).

[13] Huan Liu, and Setiono, R. A Probabilistic Approach to Feature Selection. A Filter Solution. *Machine Learning: ICML'96*, 319-327, Morgan Kaufmann, San Francisco, CA (1996).

[14] John, G.H., Kohavi, R., and Pfleger, K. Irrelevant Feature and Subset Selection. *Machine Learning: ICML'94*, 121-129, Morgan Kaufmann, San Francisco, CA (1994).

[15] Jolliffe, I.T. *Principle Component Analysis*, Springer-Verlag, New York, N.Y. (1986).

[16] Kira, K., and Rendell, L. The Feature Selection Problem: Traditional Methods and New Algorithm. *Tenth Natl. Conf. on Artificial Intelligence*, 129-134, MIT Press (1992).

[17] Kohavi, R., and John, G.H. Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97, 273-324 (1997).

[18] Koller, D., and Sahami, M. Toward Optimal Feature Selection. *Machine Learning: ICML'96, 284-292*, Morgan Kaufmann, San Francisco, CA (1996).

[19] Kullback, S., and Leibler, R.A. On information and sufficiency. *Annals of Mathematical Statistics*, 22, 76-86, (1951).

[20] Ling C.X., Li C. Data Mining for Direct Marketing: Problems and Solutions, KDD-98, AAAI Press, Menlo Park, CA (1998).

[21] Mitchel, T.M. *Machine Learning*. McGraw-Hill, Boston, MA (1997).

[22] Ng, A.Y. On Feature Selection: Lerning with Exponentially many Irrelevant Features as Training Examples. *Machine Learning: ICML'98*, 404:412, Morgan Kaufmann, San Francisco, CA (1998).

[23] Otten, R.H.J.M., and van Ginneken, L.P.P.P, *The Annealing Algorithm*. Kluwer, Boston, MA (1989).

[24] Press, W.H., Teukolsky, S.A., Vettering, W.T., and Glannery, B.P. *Numerical Recipes in C*. 2nd ed., Cambridge Univ. Press, New York, NY (1992).

[25] Quinlan J.R. C4.5: *Programs For Machine Learning*. Morgan Kaufmann, San Mateo, CA (1993).

[26] Schurmann, J. *Pattern Classification. A unified view of statistical and neural approaches*. John Wiley & Sons, New York, NY (1996).

[27] Smyth, P., and Wolpert, D. Anytime Exploratory Data Analysis for Massive Data Sets. *KDD-97*, 54-60, AAAI Press, Menlo Park, CA (1997).

[28] Tukey, J.W., Exploratory Data Analysis. Addison-Wesley, Reading, MA (1977).