Multi-objective Classification with Info-Fuzzy Networks

Mark Last

Department of Information Systems Engineering
Ben-Gurion University of the Negev
Beer-Sheva 84105, Israel
Telephone: +972-8-6461397, Fax: +972-8-6477527
mlast@bgumail.bgu.ac.il

Abstract. The supervised learning algorithms assume that the training data has a fixed set of predicting attributes and a single-dimensional class which contains the class label of each training example. However, many real-world domains may contain several objectives each characterized by its own set of labels. Though one may induce a separate model for each objective, there are several reasons to prefer a shared multi-objective model over a collection of single-objective models. We present a novel, greedy algorithm, which builds a shared classification model in the form of an ordered (oblivious) decision tree called Multi-Objective Info-Fuzzy Network (M-IFN). We compare the M-IFN structure to Shared Binary Decision Diagrams and bloomy decision trees and study the information-theoretic properties of the proposed algorithm. These properties are further supported by the results of empirical experiments, where we evaluate M-IFN performance in terms of accuracy and readability on real-world multi-objective tasks from several domains.

Keywords: Multi-objective classification, info-fuzzy networks, information theory, decision graphs, multiple output function.

1 Introduction

Mitchell [25] defines the *classification* task as "to classify examples into one of a discrete set of possible categories" (p. 54). This definition is very similar to the one provided by Fayyad et al. [12]. Such formulation of the classification problem subsumes that the class labels (categories) in question are mutually exclusive, i.e. an object cannot belong to more than one class at the same time. In the training set, the class of each instance is given by one of its attributes, called the *class label attribute* [15]. Over the years, a wide range of *supervised learning* algorithms have been developed for inducing classification models from "labeled" training examples, i.e. data items with non-empty values of the class label attribute. Examples include the Back-propagation algorithm [25], Naïve Bayes Classifier [25], C4.5 [27], IFN [23], and many others.

As indicated in [7] and [29], the assumption that a learning task has only one objective is very restrictive. Data objects in many real-world databases may be simultaneously assigned multiple class labels related to multiple tasks. These objectives (dimensions) may be strongly related to each other, completely unrelated, or just weakly related. Examples include student's grades in several courses, symptoms and diagnoses of a given patient, phonemes and stresses associated with a given word [10], etc.

J.-F. Boulicaut et al. (Eds.): ECML 2004, LNAI 3201, pp. 239–249, 2004.

Time series prediction (see [19]) is another learning task, where each sequential observation (e.g., daily stock price) is an objective in its own right. More examples of concurrent learning tasks are described in [6].

The most straightforward approach to the problem of multi-objective classification is to induce a separate model for each objective using any single-objective classification algorithm. Though the resulting models may be the best (e.g., the most accurate) ones for every individual objective, the user may find a single multi-objective model much more comprehensible than a collection of single-objective models. In non-stationary processes (see [20]), storage and maintenance of multiple models may become a tedious task. Moreover, as demonstrated by Caruana [6], the combination of several classification tasks in a single model may even increase the overall predictive accuracy.

To provide a unified framework for single-objective and multi-objective classification, we study here an *extended classification task* which includes the following components (based on [23] and [29]):

- $R = (A_1,..., A_n)$ - a set of *n candidate input features* ($n \geq 1$), where A_i is an attribute *i*. The values of these attributes (features) can be used to predict the values of *class dimensions* (see next).
- $O= (C_1,..., C_m)$ - a non-empty subset of *m class dimensions* ($m \geq 1$). This is a subset of tasks (objectives) to predict. The *extended classification task* is to build an accurate model (or models) for predicting the values of *all* class dimensions, based on the corresponding *dependency subset* (or subsets) $I \subseteq R$ of *input features*.

Section 2 of this paper discusses the related work. The methodology for inducing Multi-Objective Info-Fuzzy Networks is presented in Section 3. We are also trying to answer the following critical questions: *why* multi-objective models should work better than single-objective models and *when* the proposed algorithm is expected to maximize the predictive accuracy of the induced model. To show the practical significance of our theoretical findings, Section 4 compares the performance of single-objective and multi-objective models in terms of predictive accuracy and model simplicity. The empirical comparison is based on three multi-objective classification tasks from the areas of web mining, meteorology, and microbiology. Finally, in Section 5, we sum-up the presented methodology and discuss open problems in multi-objective classification.

2 Related Work

Based on the framework in Section 1 above, the Single-Objective Classification task can be extended to the Multi-Objective Classification task of simultaneously predicting the values of several class dimensions for a given object. The Multi-Objective Classification task is different from *Multitask Learning* described by Caruana in [6]. The explicit goal of Multitask Learning is to improve the accuracy of predicting the values of a *single-dimensional class* (defined as the *main* learning task) by training the classification model, such as a neural network or a decision tree, on several *related* tasks (additional class dimensions). This is called *inductive transfer* between learning tasks. As emphasized by [6], the only concern of Multitask Learning is the generalization accuracy of the model, not its intelligibility. "The reason for training

multiple tasks on one learner is so one task can benefit from the information contained in the training signals of other tasks, *not to reduce the number of models that must be learned*" ([6], p. 68). In contrast to [6], this paper focuses on multi-objective classification rather than on multi-task learning, since it proposes a model for simultaneous prediction of *equally important* class dimensions.

A multi-objective classifier called a *bloomy decision tree* is presented in [29]. Like ID3 and C4.5, it employs a "divide and conquer" strategy by recursively partitioning the training set. However its leaf nodes (called *flower nodes*) may predict only a subset of class dimensions. Recursive partitioning along a given path continues as long as there are unpredicted class dimensions left. Consequently, the same path may include a "sandwich" of several flower and split nodes, which need to be traversed in order to predict the values of all class dimensions. This approach significantly increases the total number of internal nodes in a tree (each path may have a flower node for every dimension), while reducing the number of dimensions predicted by smaller partitions of the training set (known as the fragmentation problem).

Representation of multiple-output functions, where all outputs are equally important, is a well-known problem in VLSI design, system testing, and other areas of computer science. *Binary Decision Diagrams* [5][24] are commonly used for representing single-output and multiple-output Boolean functions due to their time and space efficiency. A Binary Decision Diagram is a rooted acyclic graph containing two types of vertices: *non-terminal* vertices related to input variables and *terminal* vertices representing the possible output values of a Boolean function. A *Function Graph* [5] is an *ordered* Binary Decision Diagram, where the input variables appear in the same order on every path of the graph. As shown by Bryant in [5], each Boolean function has a unique (up to isomorphism) reduced function graph representation, while any other function graph denoting the same function contains more vertices.

Function graphs can be easily enhanced for representation of multi-input multi-output functions (see [24]). The idea is to construct a *Shared Binary Decision Diagram* with multiple roots (one for each output variable) [1]. The number of terminal nodes in a typical *Shared Binary Decision Diagram* is two as long as all functions are assumed to have binary outputs only. Such a diagram can be easily converted into a decision tree, where the top level(s) are used for output selection [2].

Kohavi ([17] and [18]) has extended the internal structure of non-shared (i.e. single-objective) Multi-Terminal Binary Decision Diagrams by allowing any number of outgoing edges at non-terminal nodes. Kohavi has called his model an *Oblivious Read-Once Decision Graph (OODG)*. As explained in [18], "read-once" means that each nominal feature is tested at most once along any path, which is a common property of most decision-tree algorithms such as C4.5 [27]. The name "oblivious" indicates the fact that all nodes at a given level are labeled by the same feature. As indicated above, the same ordering restriction is imposed by Bryant [5] on *Function Graphs*. An entropy-based algorithm for inducing oblivious read-once decision trees and decision graphs from data is described and evaluated in [18]. The extensive experiments performed on benchmark datasets have revealed no consistent difference between the accuracy of Kohavi's algorithm and C4.5: on average, both methods perform the same. However, in terms of *representation*, the experiments have clearly shown the capability of the OODG algorithm to produce smaller models than C4.5 for most datasets. This empirical result supports the theorem proven by Bryant in [5] that each Boolean function has a unique function graph representation having a minimal number of vertices.

A single-objective *Info-Fuzzy Network* (see [21] and [23]) has nearly the same structure as an oblivious read-once decision graph with two important differences: it extends the "read-once" restriction of [18] to continuous features by allowing *multi-way splits* of a continuous domain at the same level and it associates *probability estimates* rather than categorical predictions with each leaf node. The *predicted value* of a categorical class dimension at a terminal node is found by the popular *maximum a posteriori* rule: the predicted class is the one with the highest probability [21]. In case of a continuous class dimension, its predicted value is calculated as the mean value of all training cases associated with the particular terminal node. As demonstrated in [21], the single-objective Info-Fuzzy Network induction algorithm produces much more compact models than C4.5, while preserving nearly the same level of classification accuracy. The rest of this paper is dedicated to adaptation of info-fuzzy networks to the task of multi-objective classification. The "fuzzy" aspect of info-fuzzy networks is related to evaluation of data reliability and it is beyond the scope of this paper. An interested reader is referred to [23] for details.

3 Multi-objective Info-Fuzzy Networks

3.1 Definition of Network Structure

We assume that a multi-objective info-fuzzy network (M-IFN) has a single *root node* and its internal "read-once" structure is identical for all class dimensions. This means that every internal node is shared among *all* objectives, which makes M-IFN an extreme case of a Shared Binary Decision Diagram, where only some nodes are shared among several output functions (see [5] and [24]). This also means that each terminal (leaf) node is connected to at least one *target node* associated with a value of every class dimension. "Flower nodes" connected to only a subset of class dimensions are not allowed in M-IFNs.

M-IFNs are different from multitask decision trees [6] and bloomy decision trees [29] in two additional aspects: they are *function graphs*, since they have an oblivious read-once structure and they are also *probability estimation trees* [26], since the same terminal node may be related to several values of the same class dimension. Based on the properties of shared binary decision diagrams and function graphs (see [5]), we believe that the proposed structure should produce compact multi-objective models. Our hypothesis is tested empirically in Section 4 of this paper.

The algorithms for inducing single-objective networks from training data have been thoroughly described in previous works (see [21] and [23]). A novel algorithm for constructing a multi-objective network is presented in the next sub-section.

3.2 The M-IFN Construction Algorithm

The M-IFN induction procedure starts with defining the target layer, which has a node for each category, or value, of every class dimension and the "root" node representing an empty set of input attributes. The direct connections between the root node and the target nodes represent unconditional *(*prior) probabilities of the target values. Unlike CART [4], C4.5 [27], and EODG [18], the M-IFN construction algorithm has only the growing (top-down) phase. The top-down construction is terminated (pre-pruned) by

a statistical significance test (see below), and, consequently, there is no need in bottom-up post-pruning of the network branches. The detailed process of building the network is explained below.

M-IFN construction is an *iterative* rather than a *recursive* process. At every iteration, the algorithm utilizes the entire set of training instances to choose an input (predicting) feature (from the set of unused "candidate input" features), which maximizes the decrease in the total conditional entropy of *all* class dimensions. The conditional entropy decrease, also called *conditional mutual information* [9] or *information gain* [25], is a very common feature selection criterion in single-objective and multi-objective decision-tree algorithms (see [4] [6] [18] [27] [29], etc.).

In information theory (see [9]), conditional entropy measures the degree of uncertainty of a random variable Y given the values of other random variables $X_1, \ldots, X_n$ and it is calculated as $H(Y / X_1, \ldots, X_n) = -\Sigma\, p(x_1,\ldots, x_n, y)\ log\ p(y / x_1,\ldots, x_n,)$. If a given function is deterministic (noiseless) the conditional entropy of every output is zero.

The conditional mutual information of the class dimension Y_i and the input feature X_n given the features $X_1, \ldots, X_{n-1}$ is calculated by [9]:

$$MI(Y_i; X_n / X_1,..., X_{n-1}) = H(Y_i / X_1,..., X_{n-1}) - H(Y_i / X_1,..., X_n) = \sum_{x_1 \in X_1,..., x_n \in X_n, y_i \in Y_i} p\ (x_1,..., x_n, y_i) \log \frac{p(y_i, x_n / x_1,..., x_{n-1})}{p(y_i / x_1,..., x_{n-1})\, p(x_n / x_1,..., x_{n-1})} \quad (1)$$

At *n-th* iteration, the M-IFN algorithm chooses the input feature X_{j^*}, which maximizes the sum of information gains over all class dimensions by finding

$$j^* = \arg\max_j \sum_{i=1}^{m} MI(Y_i; X_j / X_1,..., X_{n-1}) \quad (2)$$

In a multi-objective info-fuzzy network having *n-1* layers, each internal node in the last layer represents a conjunction of values of *n-1* input features $X_1, \ldots, X_{n-1}$. Consequently, the conditional mutual information of a class dimension Y_i and an input feature X_n given the features $X_1, \ldots, X_{n-1}$ can be calculated as a sum of information gains of Y_i and X_n over all terminal nodes z in the last layer L_{n-1}:

$$MI(Y_i; X_n / X_1,..., X_{n-1}) = \sum_{z \in L_{n-1}} MI(Y_i; X_n / z) \quad (3)$$

The algorithm evaluates nominal and continuous features in a different way. Thus, the conditional mutual information of each nominal input feature X_j and the class dimension Y_i given a terminal node z is calculated by the following formula:

$$MI(Y_i; X_j / z) = \sum_{x_j \in X_j, y_i \in Y_i} p\ (z, x_j, y_i) \log \frac{p(y_i, x_j / z)}{p(y_i / z)\, p(x_j / z)} \quad (4)$$

Where x_j and y_i are distinct values of variables X_j and Y_i respectively.

In the M-IFN algorithm, we use the *Likelihood-Ratio Test* to evaluate the actual capability of an internal node to decrease the conditional entropy of an output by splitting it on the values of a particular input feature. The likelihood-ratio statistic of a

nominal input feature X_j and the class dimension Y_i given a terminal node z is measured by the following expression (based on [28]):

$$G^2(Y_i; X_j / z) = 2 \sum_{x_j \in X_j, y_i \in Y_i} N_z(x_j, y_i) \ln \frac{N_z(x_j, y_i)}{p(y_i / z) E_z(x_j)} \quad (5)$$

where $N_z(x_j, y_i)$ is the number of instances taking an input value x_j and an output value y_i at the node z and $E_z(x_j)$ is the total number of instances taking an input value x_j at the same node.

The Likelihood-Ratio Test is a general-purpose method for testing the null hypothesis H_0 that two random variables are statistically independent. Following our previous experience with single-objective IFN [23], the default significance level (*p-value*) for rejecting the null hypothesis by the M-IFN algorithm is set to 0.1%. If the likelihood-ratio statistic is significant *for at least one class dimension*, the algorithm marks the node z as "split" on the values of an input feature X_j. However, the conditional mutual information of Y_i and X_j (see Eq. (3) above) is incremented by the result of Eq. (4) only if splitting z on X_j proved to be statistically significant with respect to the class dimension Y_i. In other words, the algorithm treats statistically insignificant values of information gain as zeros. As mentioned above, M-IFN is based on the pre-pruning approach: when no input feature causes a statistically significant decrease in the conditional entropy of *any* class dimension, the top-down network construction is terminated.

Unlike EODG [19], the M-IFN induction algorithm uses *multi-way splits* on continuous input features. The threshold splits are identical for all nodes of a given layer and they are determined by a procedure similar to the information-theoretic heuristic of Fayyad and Irani [11]: recursively find a binary partition of an input feature that minimizes the total conditional entropy of all class dimensions. However, the stopping criterion we are using is different the minimum description length principle of [11]. Like in the case of nominal features (see above), we make use of the likelihood-ratio test [28] with respect to the conditional entropy of every class dimension. The search for the best partition of a continuous attribute is dynamic: it is performed each time a candidate input attribute is considered for inclusion in the network. After discretization, each hidden node in the new layer of the network is associated with an interval of the selected feature.

In Table 1 below, we show the main steps for constructing a multi-objective info-fuzzy network from a set of candidate input features.

As indicated above, the *multi-objective classification task* is to find an accurate model (or models) for predicting the values of m equally important class dimensions. The M-IFN induction procedure shown in Table 1 is a greedy algorithm that builds a *single model* aimed at minimizing the *sum of conditional entropies* of all dimensions. In [30], we show the M-IFN algorithm to have the following information-theoretic properties:

- The average conditional entropy of m class dimensions in an n-input m-dimensional model M is not greater than the average conditional entropy over m single-objective models S_i $(i=1, \ldots, m)$ based on the same n input features. This inequality is strengthened if the multi-objective model M is trained on more features than the single-objective models. Consequently, we may expect that the *aver-*

Table 1. Multi-objective Network Construction Algorithm

Input:	The set D of training examples; the set R of candidate input features; the set O of class dimensions; the minimum significance level *sign* for splitting a network node (default: *sign* = 0.1%).
Output:	A dependency subset I of input features and an info-fuzzy network. Each input feature has a corresponding hidden layer in the network.
Step 1	Initialize the info-fuzzy network (single root node representing all examples, no hidden layers, and a target layer for all values of the class dimensions). Initialize the set I of selected inputs as an empty set: $I = \varnothing$.
Step 2	While the number of layers $\|I\| < n$ (total number of candidate input features) **do**

Step 2.1	**For each** candidate input X_j / $X_j \in R$; $X_j \notin I$ **do**
	If X_j is continuous **then**
	find the best threshold splits of X_j over all class dimensions O
	Calculate the total conditional mutual information between X_j and the class dimensions O: $cond\_MI_j = \sum_{Y_i \in O} MI(Y_i; X_j / I)$
	End Do

Step 2.2	Find the candidate input X_{j*} maximizing $cond\_MI_j$

Step 2.3	**If** $cond\_MI_{j*} = 0$, **then**
	End Do.
	Else
	Expand the network by a new hidden layer associated with the feature X_{j*}, and add X_{j*} to the set I of input features $I = I \cap X_{j*}$.

Step 2.4	End Do
Step 3	Return the set of input features I and the network structure

age accuracy of a multi-objective model in predicting the values of m class dimensions will not be worse, or even will be better, than the average accuracy of m single-objective models that use the same set of input features.

- If all class dimensions are either mutually independent or totally dependent on each other, the input feature selected by the algorithm will minimize the joint conditional entropy of all class dimensions. The first case extends the scope of multitask learning [7], where "extra" tasks are assumed to be related to the main task.

4 Case Studies

Most datasets stored in the UCI Machine Learning Repository [3], UCI KDD Archive [16], and other collections of benchmark data have only one class dimension, which makes them inappropriate for the multi-objective classification task. After a careful search, we have located at [16] three datasets, which apparently have more than one class dimension. These datasets belong to three distinct domains: analysis of WWW user surveys (web mining), prediction of weather conditions (meteorology), and the impact of water quality on algae concentration (microbiology). For each data set, we run the single-objective IFN algorithm against each class dimension and compare the

average classification accuracy of the single-objective models to the accuracy of the M-IFN model. For benchmark purposes, we also present the results of additional classification / prediction algorithms that were applied in literature to these tasks. Finally, we compare the size of the multi-objective model, in terms of nodes and prediction rules, to the overall size of the single-objective models.

The *Internet Usage* dataset contains selected results of the 8th WWW User Survey conducted by the Graphics and Visualization Unit (GVU) at Georgia Tech in 1997 [13]. More than 10,000 respondents could check any number of answers out of a list of 19 not-purchasing reasons. This is a typical multi-objective classification task, where we have 19 binary-valued class dimensions. Table 2 shows the overall misclassification rates of three algorithms: C4.5 [27], single-objective IFN, and M-IFN. All three algorithms were used with their default settings. One can see that the average performance of IFN appears to be slightly better than C4.5, while there is no overall difference between IFN and M-IFN. We may conclude that the Internet Usage task agrees with the M-IFN information-theoretic properties: the multi-objective model does not decrease the average predictive accuracy, which compares fairly with the accuracy of a state-of-the-art classification algorithm (C4.5). At the same time, M-IFN has reduced the total number of nodes by 70% and the number of rules by nearly 68%.

Table 2. Internet Usage Data:Summary of Results

	C4.5	IFN	M-IFN	Change vs. IFN
Average Error Rate	0.1580	0.1524	0.1524	0.0%
Internal Nodes		342	102	-70.2%
Prediction Rules		268	86	-67.9%

The *El Nino Data Set* includes 533 meteorological measurements taken between May 23 and June 5, 1998. To find potential relationships between the measured variables, we have identified three class dimensions. Since IFN and M-IFN algorithms can handle discrete class dimensions only, the values of every continuous output have been discretized to ten intervals of equal frequency. Multiple linear regression, which was used as a benchmark method, can directly handle the continuous dependent variables. Table 3 shows the Root Mean Square Error (RMSE) of multiple linear regression, single-objective IFN, and M-IFN on the three class dimensions of El Nino dataset. Despite discretization, the single-objective IFN was superior to regression on all three class dimensions. M-IFN has further improved the average predictive performance of the single-objective IFN algorithm. Thus, the results of the El Nino task support M-IFN information-theoretic properties by showing an improvement in the average predictive accuracy of M-IFN vs. the single-objective models.

Table 3. El Nino Data: Summary of Results

	Regression	IFN	M-IFN	Change vs. IFN
RMSE	2.860	2.520	2.107	-16.4%
Internal Nodes		82	41	-50.0%
Prediction Rules		63	33	-47.6%

The data used in 1999 Computational Intelligence and Learning (*COIL*) competition comes from a microbiological study. The collected data included 340 water quality samples each containing 18 values. The seven class dimensions of each observation are the distribution of different kinds of algae. Since info-fuzzy networks can handle discrete class dimensions only, the continuous values of every output have been discretized to ten intervals of equal frequency. The results of a multi-objective artificial neural network (ANN) [8] were used as a benchmark. These results have been awarded the runner-up prize at the COIL competition. Table 4 shows the Root Mean Square Error (RMSE) of artificial neural network, single-objective IFN, and M-IFN on the seven class dimensions (algae kinds) of COIL 1999 dataset. Apparently, the performance of info-fuzzy models was slightly worse than the performance of the neural network. However this gap may be explained by the power transformations applied to the original variables before training the ANN algorithm [8]. The results of IFN algorithms presented here are based on the raw, untransformed values of all features. In any case, these results confirm again the information-theoretic properties of M-IFN by showing a slight decrease in the average error of M-IFN vs. the overall error of the seven single-objective models. We also observe a 54% decrease in the number of hidden nodes and a 50% decrease in the number of rules as a result of using a multi-objective info-fuzzy model, which on average provides us with more accurate predictions than the single-objective info-fuzzy models.

Table 4. COIL 1999 Data: Summary of Results

	Neural Network	IFN	M-IFN	Change vs. IFN
RMSE	9.069	9.841	9.657	-1.9%
Internal Nodes		37	17	-54.1%
Prediction Rules		24	12	-50.0%

5 Conclusions

In this paper, we have introduced a novel classification algorithm called M-IFN (Multi-objective Info-Fuzzy Network) for inducing an oblivious decision graph from a multi-objective data set. Using theoretical analysis and empirical evaluation, we have shown that multi-objective algorithms in general and the M-IFN algorithm in particular have a sound potential for producing compact and accurate classification models in a complex and multi-faceted learning environment.

Adaptation of other classification algorithms, such as C4.5, for the multi-objective classification task has yet to be explored. In addition, it would be interesting to see applications of M-IFN and other multi-objective classification algorithms to real-world learning tasks in health care, time series analysis, and other areas. Multi-objective models can also contribute to design of black-box test cases for multi-output software systems [22]. The information-theoretic properties of M-IFN suggest that multi-objective classification can be enhanced by analyzing dependency relations between class dimensions. This is another important direction for future research.

Acknowledgement. This work was partially supported by the National Institute for Systems Test and Productivity at University of South Florida under the USA Space and Naval Warfare Systems Command Grant No. N00039-01-1-2248.

References

1. Babu H. & Sasao T. (1998). Shared Multi-Terminal Binary Decision Diagrams for Multiple-Output Functions. *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E81-A, No.12, pp. 2545-2553.
2. Babu, H. & Sasao T. (1999). Representations of Multiple-Output Functions Using Binary Decision Diagrams for Characteristic Functions. *IEICE Trans. Fundamentals*, Vol. E82-A, No. 11, pp. 2398 – 2406.
3. Blake, C. & Merz C. J. (2000). UCI Repository of Machine Learning Databases. Machine-readable data repository, Department of Information and Computer Science, University of California at Irvine, Irvine, CA. Available at [http://www.ics.uci.edu/~mlearn/MLRepository.html]
4. Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, P.J. (1984). Classification and Regression Trees, Wadsworth.
5. Bryant, R. E. (1986). Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Transactions on Computers*, C-35-8, 677-691.
6. Caruana, R. (1993). Multitask Learning: A Knowledge-Based Source of Inductive Bias. Proceedings of the 10th International Conference on Machine Learning, ML-93, University of Massachusetts, Amherst, pp. 41-48.
7. Caruana, R. (1997). Multitask Learning. *Machine Learning*, 28, pp. 41–75.
8. Chan, R. (1999). Protecting Rivers & Streams by Monitoring Chemical Concentrations and Algae Communities. In: The 3rd International Competition of Data Analysis by Intelligent Techniques [http://www.erudit.de/erudit/competitions/ic-99/]
9. Cover T. M. & Thomas, J.A. (1991). *Elements of Information Theory*, Wiley.
10. Dietterich, T. G., Hild, H., & Bakiri, G. (1995). A Comparison of ID3 and Backpropagation for English Text-to speech Mapping. Machine Learning, 18 (1), pp. 51-80.
11. Fayyad U. & Irani, K. (1993). Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. Proc. Thirteenth Int'l Joint Conference on Artificial Intelligence, pp. 1022-1027, San Mateo, CA.
12. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery: An Overview. In U., Piatetsky-Shapiro, G., Smyth, P. & Uthurusamy, R. (Eds.), *Advances in Knowledge Discovery and Data Mining*, Fayyad, AAAI/MIT Press.
13. GVU's WWW User Survey (1998). Georgia Tech Research Corporation. [www.gvu.gatech.edu/user_surveys].
14. Han, T.S. (1978). Nonnegative Entropy Measures of Multivariate Symmetric Correlations, *Information and Control*, 36 (2):133-156.
15. Han J. & Kamber, M. (2001). *Data Mining: Concepts and Techniques*, Morgan Kaufmann.
16. Hettich, S. & Bay, S. D. (1999). The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science.
17. Kohavi, R. (1994). Bottom-Up Induction of Oblivious Read-Once Decision Graphs, Proceedings of the European Conference on Machine Learning.
18. Kohavi R. & Li, C-H. (1995). Oblivious Decision Trees, Graphs, and Top-Down Pruning, Proc. of International Joint Conference on Artificial Intelligence (IJCAI), pages 1071-1077.
19. M. Last, Y. Klein, A. Kandel (2001). Knowledge Discovery in Time Series Databases. *IEEE Transactions on Systems, Man, and Cybernetics*, Volume 31: Part B, No. 1, pp. 160-169.
20. Last, M. (2002). Online Classification of Nonstationary Data Streams", *Intelligent Data Analysis*, Vol. 6, No. 2, pp. 129-147.
21. Last M. & Maimon, O. (2004). A Compact and Accurate Model for Classification. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 2, pp. 203-215.

22. Last, M., Friedman, M. & Kandel, A. (2003). The Data Mining Approach to Automated Software Testing. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*. pp. 388 - 396.
23. Maimon O. & Last, M. (2000). *Knowledge Discovery and Data Mining – The Info-Fuzzy Network (IFN) Methodology*. Kluwer Academic Publishers, Massive Computing, Boston.
24. Minato, S. (1996). Graph-Based Representations of Discrete Functions. In Sasao T. & Fujita M. (Eds.), *Representations of Discrete Functions*. Kluwer Academic Publishers, pp. 1 – 28,
25. Mitchell, T.M. (1997). *Machine Learning*, McGraw-Hill.
26. Provost, F. & Domingos P. (2003). Tree Induction for Probability-Based Ranking. Machine Learning, 52, pp. 199–215.
27. Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
28. Rao C.R. & Toutenburg, H. (1995). Linear Models: Least Squares and Alternatives. Springer-Verlag.
29. E. Suzuki, M. Gotoh, and Y. Choki (2001). Bloomy Decision Tree for Multi-objective Classification. L. De Raedt and A. Siebes (Eds.): *PKDD 2001*, LNAI 2168, pp.436 –447.
30. Last M. & Friedman M. (2004). Black-Box Testing with Info-Fuzzy Networks. In Last, M., Kandel, A., and Bunke H. (Eds.), *Artificial Intelligence Methods in Software Testing*, World Scientific, pp. 21-50.