Annales UMCS Informatica AI 4 (2006) 60-71

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Instance reduction approach to machine learning and multi-database mining

Ireneusz Czarnowski[*], Piotr Jędrzejowicz

Department of Information Systems, Gdynia Maritime University
Morska 83, 81-225 Gdynia, Poland

Abstract

The paper proposes a heuristic instance reduction algorithm as an approach to machine learning and knowledge discovery in centralized and distributed databases. The proposed algorithm is based on an original method for a selection of reference instances and creates a reduced training dataset. The reduced training set consisting of selected instances can be used as an input for the machine learning algorithms used for data mining tasks. The algorithm calculates for each instance in the data set the value of its similarity coefficient. Values of the coefficient are used to group instances into clusters. The number of clusters depends on the value of the so called representation level set by the user. Out of each cluster only a limited number of instances is selected to form a reduced training set. The proposed algorithm uses population learning algorithm for selection of instances. The paper includes a description of the proposed approach and results of the validating experiment.

1. Introduction

One of the application areas of instance reduction algorithms is data mining, where the machine learning algorithm is used as a data mining tool. As it has been observed in [1], in the supervised learning, a machine-learning algorithm is shown a training set, which is a collection of training examples called instances. Each instance has an input vector and an output value. After learning from the training set, the learning algorithm is presented with the additional input vectors, and the algorithm must generalize, that is make a decision as to what the output value should be.

It is well known that in order to avoid excessive storage and time complexity, and possibly, to improve generalization accuracy by avoiding noise and overfitting it is often necessary to reduce the original training set by removing some instances before learning phase or to modify the instances using a new representation.

[*] Corresponding author: *e-mail address*: irek@am.gdynia.pl

Usually, instance reduction algorithms are based on distance calculation between instances in the training set. In such a case selected instances, which are placed near the centers of similar instance groups, serve as the reference instances. The approach requires using some clustering algorithms. Other methods, known as similarity-based methods, remove *k* nearest neighbors from a given category based on the assumption that all instances from the neighbor will be, after all, correctly classified [2]. The third group of methods eliminate training examples based on a classification test.

Although a variety of instance reduction methods has been so far proposed in the literature, no single approach can be considered superior nor guaranteeing satisfactory results and reduction of the learning error or increased efficiency of the supervised learning. Therefore, the problem of the selection of the reference instances is still open.

In a traditional approach machine learning and data mining algorithms are used basied on the assumption that all the training data can be pooled together in a centralized data repository. In the real life there are, however, more and more cases where the data have to be physically distributed due to some constraints. As a consequence recently the distributed data mining has attracted a lot of attention as there are many cases where pooling distributed data for mining is not feasible, due to either huge data volume or data privacy or some other reasons. Applying the traditional data mining tools to discover knowledge from the distributed data sources might not be possible [3]. Hence, knowledge discovery from multi-databases has became an important research field and is considered to be a more complex and difficult task than knowledge discovery from mono-databases [4,5].

A common methodology for distributed machine learning and data mining is of two-stage type – first performing local data analysis and then combining the local results forming the global one [5]. For example, in [6], a meta-learning process was proposed as an additional learning process for combining a set of locally learned classifiers (decision trees in particular) for a global classifier. Another approach to confront the size of datasets is to select out of the distributed databases only the relevant ones. In [7] a relevance measure was proposed to identify relevant databases for mining with an objective to find patterns or regularity within certain attributes.

An interesting solution known as the hierarchical meta-learning was proposed in [8]. Meta-learning starts with a distributed database or set of data subsets from an original database, concurrently runs a learning algorithm on each subset and combines the predictions from classifiers learned from these subsets by recursively learning “combiner” and “arbiter” models in a bottom-up tree manner.

In this paper we propose to deal with the multi-database mining through using an instance reduction approach. The focus is a selection of reference instances from distributed databases. The goal is to reduce a number of instances in each of the distributes data subsets to enable either pooling the data together and using some mono-database mining tools or effectively applying meta-learning techniques.

The paper is organized as follows. Section 2 describes the proposed instance reduction algorithm and its application to the mono-dataset and the multi-database case. Section 3 presents the results of the computation experiment carried to validate the approach. Finally, in the last section some conclusions are drawn and directions for future research are suggested.

2. Instance reduction algorithm

2.1. The general idea of algorithm

The idea of the instance reduction algorithm (IRA) was proposed in the earlier paper of the authors, where it was shown that in the case of the artificial neural network the approach can result in reducing the number of instances and still preserving a quality of the data mining results [9]. It has been also demonstrated that in some cases reducing the training set size can increase efficiency of the supervised learning.

The general idea of the proposed algorithm is based on:

- Calculating for each instance from the original set the value of its similarity coefficient.
- Grouping instances into clusters consisting of instances with identical values of this coefficient.
- Selecting the representation of instances for each cluster and removing remaining instances, thus producing the reduced training set.

The suggested instance reduction algorithm aims at removing a number of instances from the original set T and thus producing the reduced set S. Let N denote the number of instances in T and n – the number of attributes. Total length of each instance (i.e. training example) is equal to n+1, where the element numbered n+1 contains the output value. Let also $X = \{x_{ij}\}$ (i=1…N, j=1…n+1) denote a matrix of n+1 columns and N rows containing values of all instances from T. The algorithm involves the following steps:

Step 1. Transform X normalizing value of each x_{ij} into interval [0, 1] and then round it to the nearest integer, that is 0 or 1.

Step 2. Calculate for each instance from the original set the value of its similarity coefficient I_i:

$$I_i = \sum_{j=1}^{n+1} x_{ij} s_j, \quad i = 1...N ,$$

where:

$$s_j = \sum_{i=1}^{N} x_{ij}, \quad j = 1...n+1 .$$

Step 3. Map instances (i.e. rows from X) into clusters denoted as Y_v, v=1…t. Each cluster contains input vectors with identical value of the similarity coefficient I_i and t is a number of different values of I_i.

Step 4. Set value of the representation level K, which denotes the maximum number of instances to be retained in each of t clusters defined in step 3. Value of K has to be set arbitrarily by the user.

Step 5. Select instances to be retained in each cluster. Let y_v denote a number of instances in the cluster v, v = 1…t . Then the following rules for selecting instances are applied:

- If $y_v \leq K$ *and* K>1 then $S=S\cup Y_v$.
- If $y_v>K$ and K=1 then $S=S\cup\{x^v\}$, where x^v is a selected reference instance from the cluster Y_v, where the distance $d\left(x^v,\mu^v\right)=\sqrt{\sum_{i=1}^{n}\left(x_i^v-\mu_i^v\right)^2}$ is minimal and $\mu^v=\frac{1}{y_v}\sum_{j=1}^{y_v}x^v$ is the mean vector of the cluster Y_v
- If $y_v>K$ and K>1 then $S=S\cup\{x_j^v\}$, where x_j^v $\left(j=1...K\right)$ are reference instances from the cluster Y_v selected by applying the PLA algorithm [10].

2.2. Population learning algorithm

The population learning algorithms, introduced in [10], handle population of individuals, which represent coded solutions of the considered problem. Initially, a massive population of individuals, known as the initial population, is generated. The number of individuals in the initial population should be sufficient to represent adequately the whole space of feasible solutions. A sufficient number of individuals relates to the need of, possibly, covering the neighborhood of all of the local optima. Adequate representation of these neighborhoods is related to the need of assuring that the improvement process, originated at the initial stage, should be effective enough to carry at least some individuals to the highest stages of learning. The number of individuals for the particular PLA implementation is usually set at the fine-tuning phase.

Generating the initial population could be simply based on some random mechanism assuring the required representation of the whole feasible solution space. Once the initial population has been generated, individuals enter the first learning stage. It involves applying some, possibly basic and elementary, improvement schemes or conducting simple learning sessions. The improved individuals are then evaluated and better ones pass to the subsequent stage. A strategy of selecting better or more promising individuals must be defined and duly applied. At the following stages the whole cycle is repeated. Individuals are subject to improvement and learning, either individually or through information exchange, and the selected ones are again promoted to a higher stage with the remaining ones dropped-out from the process. At the final stage the remaining individuals are reviewed with a view to selecting a solution to the problem at hand.

The PLA algorithm implemented for selection of reference instances maps instances x^v from Y_v into K subset D_{vj}, where the sum of the squared Euclidean distances between each instances x^v and the mean vector μ^j of the subset D_{vj} is minimal.

Vectors with the minimal distance to the mean vector in subset are selected as K reference vectors. This selection method can be associated with one of the clustering technique known as *k*-means algorithm [2].

The representation of a solution p is coded as $(K+ y_v)$-element vector, where the first K positions define how many elements from y_v are contained in K-subset, the next y_v positions represent input vector number from Y_v.

If P denotes population of individuals and M denotes size of population, where $P=\{p_1, p_2, \dots, p_M\}$, then fitness of an individual p can by evaluated as:

$$J(p) = \sum_{j=1}^{K} \sum_{z \in T} \left\| p[z] - \mu^j \right\|^2 ,$$

where

$$T = \begin{cases} \left(K, K + p[j]\right) & if \quad j = 1, \\ \left(K + \sum_{i=1}^{j-1} p[i], K + \sum_{i=1}^{j} p[i]\right) & otherwise, \end{cases}$$

where $p[z]$ denotes the instance number which is in one of K subsets, and μ^j denotes the mean vector of the subset j.

The remaining assumptions for the PLA implementation include:

– An initial population is generated randomly.
– Four improvement methods are used (random local search, PMX crossover [11], local search, tabu search [12,13]).
– There is a common selection criterion for all stages. At each stage, individuals with fitness below the current average are rejected.

The learning/improvement procedures require some additional comments.

The first procedure, random local search modifies an individual changing position randomly selected element (i.e. instance number) in an individual. If the fitness function value has improvement then the change is accepted.

The second improvement procedure involves PMX crossover between an individual and another randomly selected one. The crossover produces an offspring of two individuals, which are evaluated. The best of individuals is retained. If the fitness function value has not improved, then the local search procedure is run.

The local search procedure modifies an individual by changing position of a randomly selected element within an individual. This element, representing the instance number, is allocated to subset, where the Euclidean distance to the mean vector is minimal. If the fitness function value of an individual has improved, then the change is accepted.

The fourth improvement procedure is based on the tabu search algorithm [12,13]. In the first step there is selected an element $p[i]$ ($i=K+1,\ldots,y_v$) within an individual. If the selected element is not on the list containing moves that at current stage are not allowed, then the position of this element within an individual is changed. If the fitness function value of an individual has improved, then the change is accepted. If an individual has not improved, then the change position is repeated for two neighbors elements of $p[i]$. If in no case the individual has not improved, then $p[i]$ is added to the list of tabu active moves and by s iterations positions for $p[i]$ and neighbors can not be changed. The parameter s is known as tabu tenure.

The implemented procedure can be run by c iterations. The both parameters, c and s have to be set by the user.

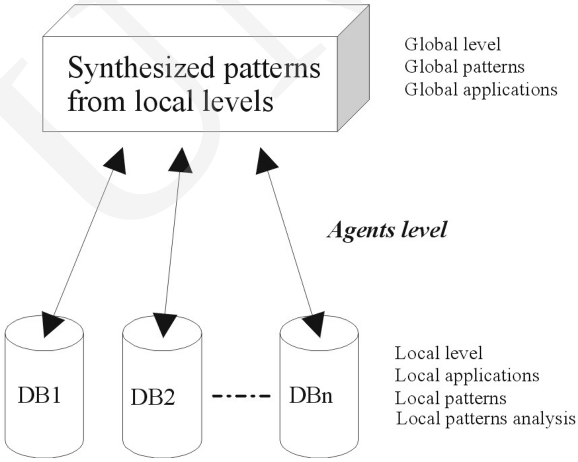

Fig. 1. The architecture of a multi-agent system for multi-database mining

2.3. Selection of reference instances from distributed databases

The proposed approach to the multi-database mining includes two stages [4]. The first involves the selection of reference instances from each of the distributed dataset. The second involves integration of the selected data and application of some data mining tools. To perform the above tasks a multi-agent system has been designed and implemented. Its architecture is shown in Fig. 1. In fact the multi-agent metaphor [14] is used in both stages – to solve the instance selection which is a combinatorial problem and to forward and integrate the distributed instance reference sets.

3. Computational experiment results

To validate the proposed approach it has been decided to carry out an experiment involving comparison of the classifier performance in the case of both: multi-database and mono-database mining. A generalization accuracy was used as a performance criterion. The classification tool used is C 4.5.

The experiment involved dataset from the EUNITE World Competition announced in 2002 [15]. The main target of this competition was to propose an effective tool supporting customer intelligence in the banking sector. The model was needed to estimate customer behavior and to predict whether the client is active or non-active. The available dataset consisted of 24000 instances including 12000 instances of “active” customers and 12000 instances of “non-active” ones. The “Customer Intelligence” problem instance includes 36 values of input attributes and a single value of the output class.

The reported computational experiment was based on the “10 cross validation” approach. At first the available dataset of 24000 instances was randomly partitioned into training and test sets, consisting of 22000 and 2000 instances respectively. The second step involved a random partition of the previously generated training set of 22000 instances into the three independent smaller datasets simulating a multi-database environment with the three distributed and separated databases. Next, each of the three thus obtained datasets was reduced using the proposed IRA approach. The reduction was carried out for 10 different representation levels $k = \{1, 5, 10, 15, 20, 25, 30, 100, 150, 200\}$. The reduced datasets were than integrated into a training set and the C 4.5 classifier was trained. Ten such trials were run ten times each, using different dataset partitions as the test set for each trial and different representation levels. The above described experiment was repeated three times for the three different dataset partitions into a multi-database. The respective number of instances in each of the dataset partitions were set to (5415, 11150, 5435), (6430, 9210, 6360), and (7010, 9630, 5360).

The experiment results are shown in Tables 1-3 and in Fig. 2. The results have been averaged over all experiment runs carried. Tables 1-2 show accuracy of classification performed on the test set by the trained C 4.5 classifier without pruned leaves, with pruned leaves and with the reduced-error pruning. The results in Table 1 concern the centralized dataset. Table 2 shows the results obtained by applying the IRA to the multi-database. The instances reduction was also executed on the global level after combining the local results. These results are shown in Table 3. Additionally, Fig. 2 depicts accuracy of classification for C 4.5 classifier, with pruned leaves, versus value of the representation level.

Table 1. Average accuracy (%) of C 4.5 results obtained for the centralized dataset

	Value of the representation level										
	K=1	K=5	K=10	K=15	K=20	K=25	K=30	K=100	K=150	K=200	Full dataset
unpruned	54.85	63.7	63.95	63.55	65.4	65.75	64.7	67.4	66.8	66.8	73.25
pruned	54.85	62.45	64.15	61.35	65.6	65.28	64.85	67.25	69.05	69.6	75.5
pruned with reduction of error (pruning)	61.85	64.15	64.55	65.4	63.1	64.15	61.5	69.65	66.55	70.6	75.15

Table 2. Average accuracy (%) of C 4.5 results obtained for the distributed datasets

	Value of the representation level										
	K=1	K=5	K=10	K=15	K=20	K=25	K=30	K=100	K=150	K=200	Full dataset
unpruned	67.35	62	65.05	68.35	65	70	68.75	67.3	69.55	70.95	73.25
pruned	68.2	65	66.2	70.54	66.75	71.6	71.5	71.7	73.5	74.55	75.5
pruned with reduction of error (pruning)	58.95	67	65.35	68.65	64.55	71.05	66.8	70.65	73.45	72.55	75.15

Table 3. Average accuracy (%) of C 4.5 results obtained for the distributed datasets with the additional reduction on the global level

	Value of the representation level										
	K=1	K=5	K=10	K=15	K=20	K=25	K=30	K=100	K=150	K=200	Full dataset
unpruned	62.85	65.2	56.6	63.5	63.75	66.25	65.1	68.1	66.05	67.85	73.25
pruned	62.85	64.8	57.1	63.95	65.35	66.35	65.3	67.95	68.25	70.65	75.5
pruned with reduction of error (pruning)	62.85	57.45	59	58.95	64.65	63.7	66.75	69.85	66.85	70.35	75.15

It should be noted that the approach produces reasonable and good results and is independent of the location of datasets. For instance, assuming the representation level of 10 for each of the three distributed datasets and pooling the selected instances together into the training set assures classification accuracy of 66.2%, whereas for the mono-database (that is the whole database without partition into the three parts) produces the classification accuracy of 64.15%. For example, the instances reduction on the global level results 58% of the classification accuracy. The representation level of 200 for the multi-dataset set results 74.55% and 70.65% of the classification accuracy after the additional reduction on the global level. In the case of the mono-database it produces the classification accuracy of 69.6%. For comparison the accuracy of classification for the full dataset equals 75.7%.

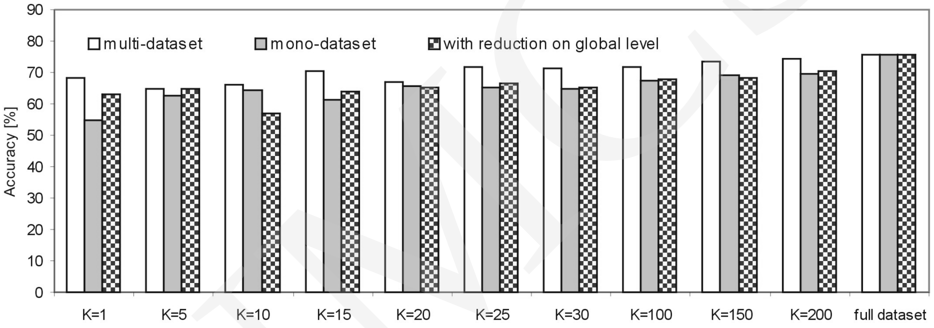

Fig. 2. Accuracy of classification versus the value of the representation level

On the other hand, Figs. 3 and 4 depict percentage and number of the retained instances versus the value of the representation level. As expected, reducing distributed datasets and pooling together the selected instances produce, for a given representation level, larger datasets than in the case of the corresponding mono-database. For instance, assuming the representation level of 10 for each of the three distributed datasets and pooling the selected instances together into the training set results in selecting 533 instances, whereas in the case of the mono-database 205 instances which amounts to 2.42% and 0.93% of the initial population of instances respectively. For comparison, the additional reduction on the global level results in selection of 186 instances that amounts to 0.85% of the initial population of instances. The representation level of 200 for the multi-dataset results in selecting 3462 instances, whereas in the case of reduction of the mono-database it results in selection of 1322 instances, which amounts to 15.74% and 6.01% of the initial population of instances respectively. The instances reduction on the global level results in selecting 1286 instances which is equal to 5.85%. However, based on the comparison of the obtained results for a similar number of instances retained the distributed approach produces, as a rule, better classification results.

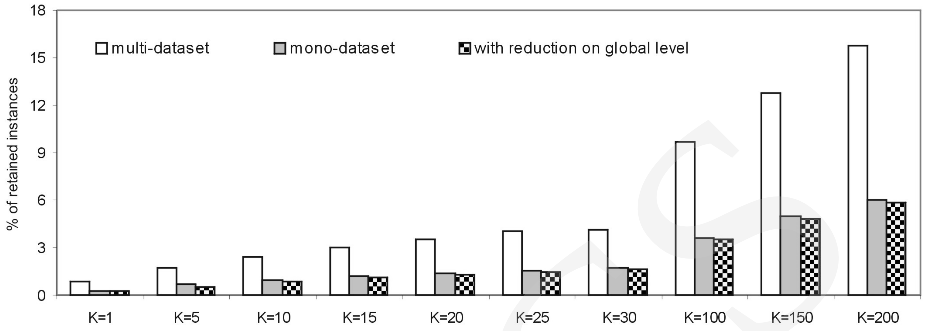

Fig. 3. Percentage of the retained instances versus value of the representation level

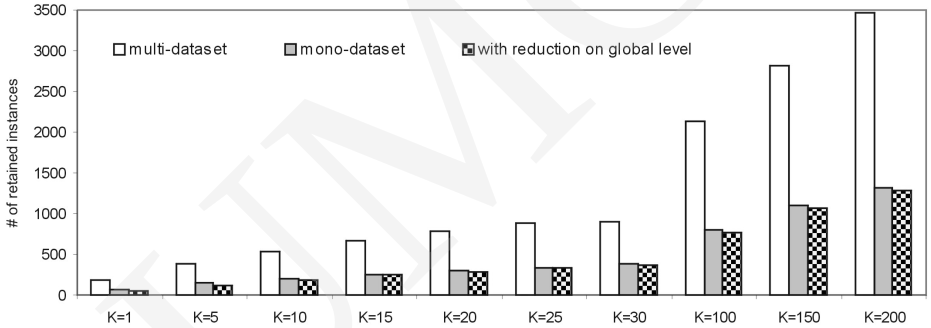

Fig. 4. Number of the retained instances versus value of the representation level

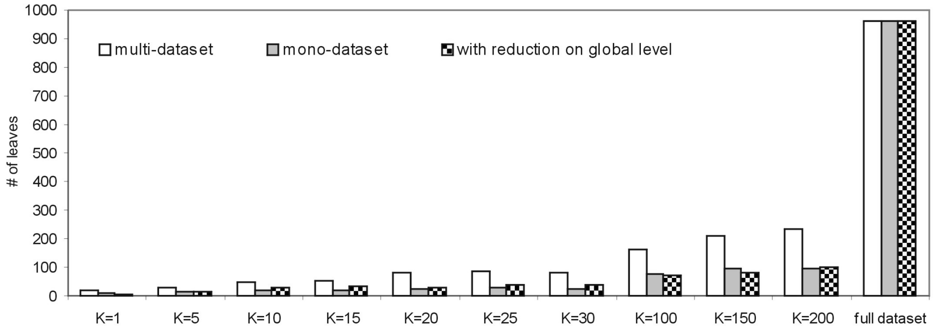

Fig. 5. Number of leaves of the decision tree versus value of the representation level

In Fig. 5 the number of leaves of the decision tree versus values of the representation level is shown. The comparison concerns the three reduction cases: multi-database, mono-database and the additional reduction on global level using C 4.5 classifier with pruned leaves. For instance, assuming the representation level of 200 for the distributed datasets the number of leaves equals 234, whereas in the case of the mono-database 96 leaves. After the reduction on the global level the number of leaves of the decision tree equals 98. These results confirm that reduction of the dataset can lead to a decreased complexity and size of the tree in comparison to the decision tree created based on the full dataset.

4. Conclusions

The paper proposes a simple heuristic approach intended for selection of the reference instances from physically separated and distributed datasets. Instances selected from the distributed databases are pooled together to provide input to data mining tools used in the knowledge discovery processes. Computation experiment results confirm that the proposed approach can be of help when confronting the problem of size of the multi-database.

The suggested algorithm allows selecting important information from many distributed databases, producing a significantly reduced dataset and still assuring a good or very good generalization accuracy on the global level.

The results show that a knowledge representation is less complexity, when created using a reduced dataset. It can result in general profit in the point of view of time computation. This conclusion is true for decision trees and for other machine learning methods.

Computation experiment results confirm also that reducing dataset size still preserves basic features of the analyzing data. Moreover, there are cases where partitioning the mono-database into the corresponding multi-database, and then reducing distributed dataset size and integrating again the selected instances together may improve mining accuracy as compared with a traditional approach. This observation may lead to the development of new data mining tools based on decomposition – integration principle.

The future research will focus on decision rules for finding an appropriate representation level suitable for each cluster allowing different representation levels for different clusters. Another direction of research will focus on validation of other classification tools with a view to overcome a danger of neglecting some important information available at the distributed level.

References

[1] Wilson, D.R., Martinez, T.R., *Reduction techniques for instance-based learning algorithm*. Machine Learning, Kluwer Academic Publishers, Boston, (33) (2000) 257.

[2] Likas, A., Vlassis N., Verbeek J.J., *The global k-means clustering algorithm*. Pattern Recognition, 36(2) (2003).

[3] Kargupta H., Park B., Hershberger D., Johnson E., *Collective Data Mining: A New Perspective Toward Distributed Data Analysis*. Accepted in the Advances in Distributed Data Mining, H. Kargupta and P. Chan (eds.), AAAI/MIT Press, (1999).

[4] Ahang S., Wu X., Zhang C., *Multi-Database Mining*. IEEE Computational Intelligence Bulletin, (2)1 (2003).

[5] Zhang X., Lam C., Cheung W.K., *Mining Local Data Sources For Learning Global Cluster Model Via Local Model Exchange*. IEEE Intelligence Informatics Bulletine, (4) 2 (2004).

[6] Prodromidis A., Chan P.K., Stolfo S.J., *Meta-learning in Distributed Data Mining Systems: Issues and Approaches*. In: Advances in Distributed and Parallel Knowledge Discovery, Kargupta H., Chan P.(ed.), AAAI/MIT Press, Chapter 3, (2000).

[7] Liu H., Lu H., Yao J., *Identifying Relevant Databases for Multidatabase Mining*. In: Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining, (1998) 210.

[8] Prodromidis A., Stolfo S., *Pruning meta-classifiers in a distributed data mining system*. Proceedings of the First National Conference on New Information Technologies, (1998) 151.

[9] Czarnowski I., Jędrzejowicz P., *An Approach to instance reduction in supervised learning*. In Coenen F., Preece A. and Macintosh A. (eds.), Research and Development in Intelligent Systems XX, Springer, London, (2004) 267.

[10] Jędrzejowicz, P., *Social Learning Algorithm as a Tool for Solving Some Difficult Scheduling Problems*. Foundation of Computing and Decision Sciences 24, (1999) 51.

[11] Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*. 3rd edn. Springer-Verlag, Berlin Heidelberg New York, (1996).

[12] Glover F., *Tabu Search – Part I*. ORSA Journal of Computing, 1 (1990) 190.

[13] Glover F., *Tabu Search – Part II*. ORSA Journal of Computing, 2 (1990) 4.

[14] Guo Y, Műller J.P., *Multiagent Collaborative Learning for Distributed Business Systems*. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, New York, (2004).

[15] The European Network of Excellence on Intelligence Technologies for Smart Adaptive Systems (EUNITE) – EUNITE World Competition in domain of Intelligent Technologies – http://neuron.tuke.sk/competition2