



Local feature selection for heterogeneous problems

I. Skrypnik, A. Tsymbal & S. Puuronen

*Department of Computer Science and Information Systems,
University of Jyväskylä, Finland.*

Abstract

Current electronic data repositories contain enormous amount of data including also unknown and potentially interesting patterns and relations. One approach commonly used is supervised machine learning, in which a set of training instances is used to train one or more classifiers that map the space formed by different features of the instances into the set of class values. The classifiers are later used to classify new instances with unknown class values. The multidimensional data is sometimes feature-space heterogeneous so that different features have different importance in different subareas of the whole space. In this paper we describe a technique that searches for a division of the feature space identifying the best subsets of features for each instance. Our technique is based on the wrapper approach where a classification algorithm is used as the evaluation function to differentiate between several feature subsets. On the other hand, to make the feature selection dynamic or local, we apply the recently developed technique for dynamic integration of classifiers to determine what classifier and with what feature subset is applied for each new instance. Our technique can be applied in the case of implicit heterogeneity when the regions of heterogeneity cannot be easily defined by a simple dependency. We make experiments with well-known datasets of the UCI machine learning repository using ensembles of simple base classifiers. The base classifiers discussed in the experiments are developed by C4.5 based on only one feature each but the combining method itself is allowed to take into account unlimited amount of present features. The results achieved are promising and show that the local data mining in comparison with mining the whole space can be advantageous. In many cases, the mining accuracy is better as well as the time of processing is shorter, because only a small set of features is used to classify each new instance.



1 Introduction

Current electronic data repositories contain enormous amount of data including also unknown and potentially interesting patterns and relations, which are tried to be revealed using knowledge discovery and data mining methods [1]. One approach commonly used is supervised machine learning, in which a set of training instances is used to train one or more classifiers that map the space formed by different features of the instances into the set of class values [2]. Each training instance is usually represented by a vector of the values of the features and the class value of the instance. An induction algorithm is used to learn a classifier, which maps the space of feature values into the set of class values. The classifiers are later used to classify new instances with unknown class values.

Data, which is the subject of analysis and processing in knowledge discovery and data mining, is usually multidimensional, and presented by a number of features for better representation of the problem domain. One significant problem in data mining is the presence of irrelevant features. When several such features are present in the example descriptions, many data mining techniques can be confused and it may result in severe degradation of accuracy. A natural solution to this problem is to identify the irrelevant features, and discard them before storing the examples for future use [3].

The multidimensional data is sometimes feature-space heterogeneous so that different features have different importance in different subareas of the whole space. Many methods have been proposed for the purpose of feature selection, but almost all of them ignore the fact that some features may be relevant only in context (i.e. in some regions of the instance space). This ignorance may consist in discarding the features that are highly relevant in a restricted region of the instance space because this relevance is swamped by their irrelevance everywhere else. The ignorance may also consist in retaining the features that are relevant in most of the space, but still unnecessarily confuse the classifier in some regions of the space [4].

In this paper we describe a technique that searches for a division of the feature space identifying the best subsets of features for each instance. To make the feature selection local, we apply the recently developed technique for dynamic integration of classifiers to determine what classifier and with what feature subset is applied for each new instance [5]. Our technique can be applied in the case of implicit heterogeneity when the regions of heterogeneity cannot be easily defined by a simple dependency. We make experiments with well-known datasets of the UCI machine learning repository using ensembles of quite simple base classifiers [6]. Our results are promising and show that the local data mining in comparison with mining the whole space can be advantageous.

In Chapter 2 we consider the dynamic integration of multiple classifiers. Chapter 3 discusses local feature selection with the dynamic classifier integration. In the next chapter we consider our experiments with the local feature selection technique on different datasets. We conclude briefly in Chapter 5 with a summary and further research topics.



2 Dynamic integration of multiple classifiers

In this chapter the dynamic integration of multiple classifiers is discussed, and a variation of stacked generalization, which uses a metric to locally estimate the errors of the component classifiers, is considered.

Recently the integration of multiple classifiers has been under active research in machine learning, and different approaches have been considered [7]. The challenge of integration is to decide which classifier to rely on or how to combine classifications produced by several classifiers. The integration of an ensemble of classifiers has been shown to yield higher accuracy than the most accurate component classifier alone in different real-world tasks. The two main approaches to the integration are: first, *combination approach*, where the base classifiers produce their classifications and the final result is composed using those classifications and second, *selection approach*, where one (the best one) of the classifiers is selected and the final result is the result produced by it.

The most popular and simplest method of combining classifiers has been voting (also called majority voting and Select All Majority, SAM) [2]. In this simple method the classification produced by a base classifier is considered as a vote for particular class value, the votes are considered as equally weighted, and class value with most votes is selected as the final classification result. The weighted voting (WV) [2], stacked generalization [8], the SCANN method [9], combining the minimal nearest neighbor classifiers [10], and different versions of resampling (as boosting, bagging, and cross-validated resampling) [11] are examples of more sophisticated combination algorithms.

One very popular but simple selection approach is CVM (Cross-Validation Majority) [12], which estimates the accuracy of each base classifier using cross-validation and selects a classifier with the highest accuracy. CVM is an example of the *static* selection methods that select one classifier for the whole data space. More sophisticated selection approaches use estimates of the local accuracy of the base classifiers by considering errors made with the most similar instances [13] or the meta-level classifiers, which predict the correctness of the base classifiers for a new instance [14]. These more sophisticated selection methods are examples of the *dynamic* selection methods.

We have elaborated a dynamic approach that uses estimation of the local accuracy of the base classifiers by analyzing the accuracy in near-by instances [5]. Instead of training a meta-level classifier that will derive the final classification using the classifications of the base classifiers as in stacked generalization, we propose to train a meta-level classifier that will estimate the errors of the base classifiers for each new instance and then use these errors to derive the final classification. To predict the errors of the base classifiers, we use weighted nearest neighbor classification (WNN) [15].

The dynamic integration technique contains two main phases [5]. First, in the learning phase, the training set is partitioned into folds. The cross-validation technique is used to estimate the errors of the base classifiers on the training set and the meta-level training set is formed. It contains all the attributes of the training instances and the estimates of the errors of the base classifiers on those



instances. Second, in the application phase, the combining classifier is used to predict the performance of each component classifier for a new instance. We have considered two different functions implementing the application phase [5]: the dynamic selection (DS) and the dynamic voting (DV). In the application phase the DS selects a classifier with the smallest predicted classification error using the WNN procedure. On the other hand the DV uses the weights of the base classifiers and weighted voting. The weights themselves depend on the local performance of the classifiers.

3 Local feature selection

In knowledge discovery and data mining the object of processing is usually multidimensional data presented by a number of features. Commonly there are present also many irrelevant features. In this chapter the feature selection problem is discussed based on local considerations of the relevance of each feature and this forms background for the experimental part that is considered in the next chapter.

A solution to this problem is to identify the irrelevant features and discard them generating smaller (minimum) subset of features retaining all information needed for further processing [3]. The task to identify the irrelevant features has mostly been tried to solve globally by covering the whole domain space by a single subset of features. This may result in two kinds of problematic situations. First, when some feature is relevant only inside a small subspace, it may globally be evaluated irrelevant or almost irrelevant and discarded from the set of relevant features even if it is highly relevant inside the subspace. Second, when some feature is almost globally relevant, it can still be totally irrelevant inside some small subspace and may confuse the classifiers totally inside this subspace [4]. Both of these situations are hard to solve using the global set of relevant features with a heterogeneous feature-space, where the features that are important for data mining are different in different regions of the feature space [16].

There are two main approaches to solve the above problem [17]. First, a more complex global model can be generated to include the necessary information, for example by adding extra artificial features. Second, the data mining problem can be divided into subproblems and the solution of the whole classification task can be guided by the heterogeneity of the feature space. This decomposition approach is potentially highly beneficial, because most widely used data mining techniques (including decision trees, rules, and neural networks) rely on measures that are computed over all the features and all the examples at hand, and are inevitably diffused by an averaging effect over the entire problem [17]. We apply the second approach in this paper trying to fit a simple model to local regions instead of trying to build a single global model for the whole feature space.

Differentiation between feature subsets can be based on a heuristic measure, as the quality of the class separation, or on the actual accuracy obtained by applying the classifier using those features [3]. A feature selection algorithm applying a heuristic measure acts as a *filter* extracting features from feature set



before its use by the main algorithm. A feature selection algorithm applying the actual accuracy acts as a *wrapper* around that the main algorithm. We consider a dynamic feature selection technique that uses wrapper approach [3] because it has been found to yield the best results. The evaluation function in our case is the local classification accuracy obtained by applying the classification algorithm with different feature subsets. We propose to apply a technique for dynamic integration of classifiers [5]. It allows us to determine what classifier and with what feature subset should be applied for each new instance. This technique can also be applied in the case of implicit heterogeneity when the regions of heterogeneity cannot be defined by a simple dependency. In our method the classifiers included in the ensemble (the base classifiers) are built based on various subsets of the original feature set. We considered this method previously in [18].

In this paper we consider an advanced version of the method presented in [18]. We previously analyzed all possible combinations of features to build the ensemble of base classifiers. However, this can be very expensive computationally and can lead to overfitting. The big number of feature subsets and, consequently, integrated classifiers dramatically increases the number of degrees of freedom in the training process, leading to increased variance of predictions and an increased risk of overfitting the data. To reduce the risk, we propose to limit the number of considered feature subsets in our local wrapper technique. The base classifiers in the ensemble can be built using combinations of only potentially locally relevant features, discarding features that are definitely irrelevant at that region. Some recursive partitioning techniques or some heuristic measures can be used to discard features that are locally irrelevant with a high probability. Thus, we propose to combine our wrapper-based method with a filter approach, using it in advance for restricting the possible feature combinations.

In [19] a decision tree was proposed to be used for local feature selection, where only those features are considered to be locally relevant for a test case, which lie on the path taken by the test case in the decision tree. We propose to combine this filter feature selection with our method that should speed up the learning and classification processes and raise the accuracy of classification.

4 Experimental results

In this chapter, we present our experiments with the use of the C4.5 decision tree algorithm to guide the local feature selection process conducted using the local wrapper approach with the help of dynamic integration of classifiers. First we describe the experimental setting and then present results of our experiments. We conduct the experiments on ten datasets taken from the UCI machine learning repository [6] and on the Dystonia dataset considered in [20]. Previously we have experimentally evaluated the dynamic classifier integration [5] and the unguided local feature selection [18]. Here we use a similar experimental environment, but this time the algorithm builds C4.5 decision trees with and without pruning [21] at the end of the training phase. These decision trees are

used to guide the selection or voting during the application phase in order to restrict the number of feature combinations being analyzed. For each test instance we consider only those feature combinations that include features present in the path taken by the test instance in the decision tree.

For each dataset 30 test runs are made. In each run the dataset is first split into the training set and the test set by random sampling. Each time 30 percent of the instances of the data set are first randomly picked up to the test set. The rest 70 percent of the instances are then passed to the learning algorithm where they are divided into 10 folds using stratified random sampling. In our test environment we apply 10-fold cross-validation to build the cross-validation history for the dynamic integration of the base classifiers. The base classifiers themselves are learned using the C4.5 decision tree algorithm with pruning [21] for each single feature separately. Thus in the case of DNA dataset with 180 features there exist 180 base classifiers. In the estimation of the classification errors of the base classifiers for a new instance we use the collected classification information about classification errors for seven nearest neighbors of the new instance. The selection of the number of folds and the number of nearest neighbors has been discussed in [5]. Based on the comparisons between different distance functions for dynamic integration presented in [22] we decided to use the Heterogeneous Euclidean-Overlap Metric, which produced good test results. The test environment was implemented within the MLC++ framework (the machine learning library in C++) [23].

The left-hand side of Table 1 shows the averages of the classification accuracies. The first five columns include the average of the minimum accuracies of the base classifiers (min), the average of average accuracies of the base classifiers (aver), the average of the maximum accuracies of the base classifiers (max), the average percentage of test instances where all the base classifiers managed to produce the right classification (agree), and the average amount of test instances where at least one base classifier during each run managed to produce the right classification (cover). The four next columns of the right-hand side of Table 1 include the accuracies for the four types of integration of the base classifiers. These are: CVM – Cross-Validated Majority, WV – Weighted Voting, DS – Dynamic Selection, and DV – Dynamic Voting. The three columns of the right-hand side of Table 1 include the minimum (min), average (aver), and maximum (max) number of features used to classify test instances. All the above columns are averaged over the 30 test runs.

In vertical direction the main body of Table 1 is divided into groups of three rows each corresponding to one dataset. The first row of each dataset contains accuracies received with unguided local feature selection, the second row contains accuracies received when the local feature selection is guided by the C4.5 decision tree with pruning, and the third one contains accuracies received when the local feature selection is guided by the C4.5 decision tree without pruning. The last group in the table shows accuracies averaged over all the datasets.



Table 1: Accuracy values and numbers of analysed features for our algorithm of local feature selection.

DB	Classification accuracy									Number of features		
	min	aver	max	Agree	cover	CVM	WV	DS	DV	min	aver	max
Breast	0.673	0.706	0.737	0.473	0.887	0.695	0.709	0.709	0.709	9	9	9
	0.332	0.669	0.906	0.551	0.831	0.697	0.723	0.700	0.730	1.833	3.294	6.000
	0.561	0.720	0.875	0.524	0.840	0.695	0.725	0.697	0.725	1.933	4.432	7.433
Diabetes	0.561	0.637	0.713	0.179	0.968	0.709	0.670	0.700	0.664	8	8	8
	0.529	0.626	0.717	0.209	0.961	0.710	0.682	0.678	0.667	2.133	6.755	8.000
	0.536	0.629	0.717	0.201	0.962	0.710	0.681	0.678	0.667	2.133	6.855	8.000
DNA	0.503	0.527	0.685	0.104	1.000	0.640	0.524	0.594	0.524	180	180	180
	0.000	0.292	1.000	0.305	0.997	0.731	0.521	0.731	0.517	2.000	5.723	12.73
	0.000	0.328	1.000	0.301	0.997	0.728	0.521	0.728	0.517	3.000	7.261	14.20
Dystonia	0.892	0.977	1.000	0.847	1.000	1.000	1.000	1.000	1.000	7	7	7
	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.133	1.200
	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.133	1.200
Glass	0.274	0.430	0.570	0.004	0.858	0.517	0.590	0.530	0.570	9	9	9
	0.256	0.440	0.631	0.046	0.810	0.506	0.535	0.502	0.548	2.500	6.085	7.867
	0.278	0.465	0.713	0.042	0.812	0.506	0.542	0.512	0.556	2.500	6.435	8.533
Heart	0.493	0.641	0.793	0.013	0.999	0.720	0.770	0.641	0.741	13	13	13
	0.364	0.641	0.843	0.145	0.983	0.715	0.732	0.621	0.704	2.533	5.933	8.233
	0.354	0.640	0.840	0.138	0.984	0.716	0.730	0.621	0.703	2.533	6.032	8.233
Iris	0.502	0.775	0.965	0.351	0.986	0.942	0.935	0.930	0.934	4	4	4
	0.735	0.879	0.965	0.890	0.974	0.941	0.945	0.948	0.945	1.000	1.578	2.300
	0.412	0.764	0.966	0.848	0.978	0.941	0.942	0.939	0.934	1.000	1.752	3.000
Liver	0.496	0.557	0.623	0.066	0.967	0.563	0.596	0.597	0.616	6	6	6
	0.500	0.565	0.628	0.099	0.965	0.574	0.618	0.595	0.629	1.133	5.659	6.000
	0.500	0.565	0.628	0.099	0.965	0.574	0.618	0.595	0.629	1.133	5.659	6.000
MONK-1	0.403	0.491	0.757	0.091	0.957	0.757	0.509	0.453	0.462	6	6	6
	0.308	0.490	0.762	0.338	0.927	0.757	0.612	0.623	0.575	1.000	2.963	4.967
	0.314	0.490	0.757	0.332	0.937	0.757	0.613	0.625	0.577	1.000	3.053	5.167
MONK-2	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669	0.669	6	6	6
	0.566	0.646	0.701	0.669	0.669	0.669	0.669	0.669	0.669	2.767	4.509	6.000
	0.572	0.643	0.692	0.669	0.669	0.669	0.669	0.669	0.669	2.967	4.778	6.000
MONK-3	0.473	0.604	0.813	0.290	1.000	0.781	0.570	0.787	0.566	6	6	6
	0.469	0.706	0.872	0.591	1.000	0.781	0.763	0.786	0.943	1.967	2.104	3.000
	0.469	0.706	0.872	0.591	1.000	0.781	0.763	0.786	0.943	1.967	2.104	3.000
Average	0.536	0.631	0.751	0.271	0.942	0.717	0.680	0.681	0.667			
	0.458	0.625	0.813	0.430	0.923	0.728	0.703	0.706	0.708			
	0.458	0.628	0.816	0.422	0.925	0.726	0.703	0.706	0.709			

When the numbers of features are compared inside the three lines of the groups corresponding to all the datasets, one can see that the use of the C4.5 decision tree algorithm with or without pruning for local feature selection significantly reduces the number of locally analyzed features. When the



corresponding accuracies achieved are compared one can see that this has often happened without a loss in the classification accuracy. For many datasets, the number of locally analyzed features is several times less in average than the total number of features. For example, in the DNA dataset there are totally 180 features, whereas the average number of locally analyzed features is 5.723 for pruned C4.5 and 7.261 for C4.5 without pruning. The accuracy with this dataset is even higher with the guided feature selection, reaching 0.731 with C4.5 without pruning for CVM and DS. Thus, sometimes the guided feature selection helps to improve the classification accuracy, as on the Breast, DNA, Iris, Liver, MONK-1, and MONK-3 datasets. However, some datasets are complex, containing many locally relevant features, even more than there are levels in the decision tree. In those cases, the guided local feature selection usually produces slightly lower accuracy than the unguided feature selection, as on the Diabetes and Glass datasets.

For some datasets, the dynamic integration (local feature selection) is clearly better than the static approaches, as on the Liver and MONK-3 datasets. The MONK-3 dataset is a good example of the benefit from the local feature selection guided by a decision tree. On this dataset, the guided feature selection is better than the unguided one, and the dynamic integration is better than the static integration. The dynamic integration of classifiers built on different feature subsets guided by the C4.5 decision tree is the best choice for this domain. The C4.5 decision tree clearly helps to reject the three irrelevant features present in the dataset. One can see that at maximum, just three relevant features are selected as locally relevant with the C4.5 on this dataset.

According to the last group of the table, the guided feature selection is better on average than the unguided one. The C4.5 without pruning naturally generates bigger trees, and it leads to bigger average number of locally relevant features selected. However, the accuracies of the algorithm with and without pruning for C4.5 usually are almost undistinguishable. Only on the Glass dataset pruning gives clearly better results.

5 Conclusions

In knowledge discovery and data mining the object of processing is usually multidimensional data presented by a number of features. The multidimensional data is sometimes feature-space heterogeneous so that different features have different importance in different subareas of the whole space.

In this paper we described a technique that searches for a division of the feature space identifying the best subsets of features for each instance. The technique is based on the wrapper approach, and uses the method for dynamic integration of classifiers to determine what classifier and with what feature subset is applied for each new instance. The technique can be applied in the case of implicit heterogeneity when the regions of heterogeneity cannot be easily defined by a simple dependency. At the application phase, in order to restrict the number of feature combinations being analyzed, we used the C4.5 decision trees built on the whole feature space. For each test instance we considered only those



feature combinations that included features present in the path taken by the test instance in the decision tree.

We conducted experiments on different datasets of the UCI machine learning repository using ensembles of simple base classifiers each generated on only one feature. The results achieved are promising and show that the local data mining in comparison with mining the whole space can be advantageous. In many cases, the mining accuracy is better as well as the time of processing is shorter, because only a small set of features is used to classify each new instance.

Further experiments can be conducted to make deeper analysis of applying recursive partitioning and the dynamic integration of classifiers for local feature selection. Decision trees built on the whole instance set were used in our experiments. However, using local decision trees could improve the selection of locally relevant features. Another potentially interesting topic for further research is the analysis of feature subsets containing more than one feature.

Acknowledgements

This research is partly supported by the COMAS Graduate School of the University of Jyväskylä and partly by the CIMO organization. We would like to thank the UCI machine learning repository of databases, domain theories and data generators for the datasets, and the machine learning library in C++ for the source code used in this study.

References

- [1] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. & Uthurusamy, R. *Advances in Knowledge Discovery and Data Mining*, AAAI/ MIT Press, 1997.
- [2] Aivazyan, S.A. *Applied Statistics: Classification and Dimension Reduction*, Finance and Statistics: Moscow, 1989.
- [3] Dash, M., & Liu, H. Feature selection for classification. *Intelligent Data Analysis*, **1** (3), Elsevier Science, 1997.
- [4] Domingos, P. Context-sensitive feature selection for lazy learners. *Journal of Artificial Intelligence Review*, **11**(1-5), pp. 227-253, 1997.
- [5] Puuronen, S., Terziyan, V. & Tsymbal, A. A dynamic integration algorithm for an ensemble of classifiers. *Foundations of Intelligent Systems: 11th Int. Symp. ISMIS'99. Lecture Notes in AI*, eds. Z.W. Ras, & A. Skowron, Vol. 1609, Springer-Verlag: Warsaw, pp. 592-600, 1999.
- [6] Blake, C.L., & Merz, C.J. *UCI Repository of Machine Learning Databases* [[http:// www.ics.uci.edu/~mllearn/ MLRepository.html](http://www.ics.uci.edu/~mllearn/MLRepository.html)], Department of Information and Computer Science, Un-ty of California, Irvine CA, 1998.
- [7] Dietterich, T.G. Machine learning research: four current directions. *AI Magazine*, **18** (4), pp. 97-136, 1997.
- [8] Wolpert, D. Stacked generalization. *Neural Networks*, **5**, pp. 241-259, 1992.



- [9] Merz, C.J. Dynamical selection of learning algorithms. *Learning from Data, Artificial Intelligence and Statistics*, eds. D. Fisher, & H.-J. Lenz, Springer-Verlag: New York, 1996.
- [10] Skalak, D.B. *Combining Nearest Neighbor Classifiers*, Ph.D. Thesis, Dept. of Computer Science, University of Massachusetts, Amherst MA, 1997.
- [11] Schapire, R.E. Using output codes to boost multiclass learning problems. *Machine Learning: Proc. of the 14th Int. Conf.*, pp. 313-321, 1997.
- [12] Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proc. of IJCAI-95 the 14th Int. Joint Conf. on Artificial Intelligence*, ed. C. Mellish, Morgan Kaufmann, 1995.
- [13] Merz, C.J. Using correspondence analysis to combine classifiers. *Machine Learning*, **36** (1-2), pp. 33-58, 1999.
- [14] Koppel, M., & Engelson, S.P. Integrating multiple classifiers by finding their areas of expertise. *AAAI-96 Work. On Integrating Multiple Learning Models*, pp.53-58, 1996.
- [15] Cost, S., & Salzberg, S. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, **10** (1), pp. 57-78, 1993.
- [16] Apte, C., Hong, S.J., Hosking, J.R.M., Lepre, J., Pednault, E.P.D. & Rosen, B.K. Decomposition of heterogeneous classification problems. *Advances in Intelligent Data Analysis (IDA-97)*, eds. X. Hiu, P.Cohen & M. Berthold, Springer-Verlag: London, pp. 17-28, 1997.
- [17] Atkeson, C.G., Moore, A.W. & Schaal, S. Locally weighted learning. *Artificial Intelligence Review*, **11**(1-5), pp. 11-73, 1997.
- [18] Skrypnik, I., Terziyan, V., Puuronen, S. & Tsymbal, A. Learning feature selection for medical databases. *Proc. 12th IEEE Symp. on Computer-Based Medical Systems CBMS'99*, IEEE CS Press: Stamford CT, pp.53-58, 1999.
- [19] Cardie, C., & Howe, N. Improving minority class prediction using case-specific feature weights. *Proc. 14th Int. Conf. on Machine Learning*, Morgan Kaufmann, pp. 57-65, 1997.
- [20] Terziyan, V., Tsymbal, A. & Puuronen, S. The decision support system for telemedicine based on multiple expertise. *Int. J. of Medical Informatics*, **49** (2), pp. 217-229, 1998.
- [21] Quinlan, J.R. *C4.5 Programs for Machine Learning*, Morgan Kaufmann: San Mateo CA, 1993.
- [22] Puuronen, S., Tsymbal, A. & Terziyan, V. Distance functions in dynamic integration of data mining techniques. *Data Mining and Knowledge Discovery: Theory, Tools, and Techniques*, ed. B.V. Dasarathy, SPIE-The International Society for Optical Engineering, USA, 2000. (to be published)
- [23] Kohavi, R., Sommerfield, D. & Dougherty, J. Data mining using MLC++: A Machine Learning Library in C++. *Tools with Artificial Intelligence*, IEEE CS Press, pp. 234-245, 1996.