Proceedings of the 25th IASTED International Multi-Conference
ARTIFICIAL INTELLIGENCE AND APPLICATIONS
February 12–14, 2007, Innsbruck, Austria
ISBN Hardcopy: 978-0-88986-629-4 / CD: 978-0-88986-631-7

INCREMENTAL CLASSIFIER BASED ON A LOCAL CREDIBILITY CRITERION

H. Prehn
Institute of Computer Science and Applied Mathematics
Christian-Albrechts-University of Kiel, Germany
email: hp@ks.informatik.uni-kiel.de

G. Sommer
Institute of Computer Science and Applied Mathematics
Christian-Albrechts-University of Kiel, Germany
email: gs@ks.informatik.uni-kiel.de

ABSTRACT

In this paper we propose the Local Credibility Concept (LCC), a novel technique for incremental classifiers. It measures the classification rate of the classifier's local models and ensures that the models do not cross the borders between classes, but allows them to develop freely within the domain of their own class. Thus, we reduce the dependency on the order of training samples, an inherent problem of incremental methods, and make the classifier robust w.r.t. selecting the algorithm's parameters. These only influence the number of models, whereas the performance is controlled by the LCC automatically on a local scale. In contrast to other algorithms, the models of our method are more adaptable as they can also shrink and vanish. This allows classes to move their domains in the data space making the LCC-Classifier also applicable to drifting data concepts. We present experiments to demonstrate these capabilities as well as some benchmark tests that show the algorithm's competitive performance.

KEY WORDS
Incremental Classification, Local Credibility Criterion

1 Introduction[1]

For data classification incremental algorithms have become inevitable. Either the size of databases prohibits the use of batch learning or constant arrival of new data forces systems to life-long learning. A great variety of incremental classifiers [2, 6, 8, 12, 7, 5] has been developed and established as state-of-the-art methods that show high performance. However, there are still two typical difficulties with incremental methods that have not been solved sufficiently, so far. The first problem is the dependency on the order in which samples are presented when building the data representation incrementally. For two different sequences of the same training samples the topology and classification quality of the resulting data representation can vary significantly. The second problem is estimating the method's parameters like e.g. the size for local models or learning rates. These values depend on the inherent scale and distribution of the data, which is only revealed during the training process. But as many algorithms react very sensitive to these parameters their selection is a non-trivial task.

Here we propose an incremental classifier that alleviates both problems. For an easier parameter selection we developed the Local Credibility Criterion (LCC), a quality measure that uses the classification rate of the individual local models to control their development. For robustness w.r.t. the order of training samples we adapted our no-competition strategy for incremental clustering [10] to the scenario of supervised learning.

In the remainder of the paper, we describe related incremental classification methods in Section 2. Section 3 introduces the Local Credibility Criterion and our classifier based on it. In Section 4 we show the quality of our method on various data sets and its robustness w.r.t. parameter selection. Section 5 concludes the paper.

2 Related Work

In this section, we briefly review related incremental classification technologies. These algorithms are also based on local models like our method which means that they approximate a data distribution with a set of local models. This is quite similar to a mixture of Gaussians, but as many algorithms do not use Gaussian distributions we prefer the more general term of local models.

In [2] Carpenter and Grossberg extended their Adaptive-Resonance-Theory-technology (ART) for incremental clustering to the scenario of supervised classification (of maps), calling it ARTMAP. Based on that technology a whole family of incremental neural network classifiers [2, 5] has been developed. In all these ARTMAP-like approaches single neurons model local domains in the input space. Depending on the applied metric, these can be hyper rectangles, hyper spheres or hyper ellipsoids, either suited for binary or continuous-valued samples, incorporating Gaussian distribution or not. For all these approaches, the process of updating is very similar. Depending on whether the distance between a neuron's centroid and a data sample is greater or less than the global *vigilance* parameter a neuron is activated i.e. it responds or not. The neuron with the highest response is updated by including the sample into its representation. If no neuron (model) responds, a

[1] The work presented here was supported by the European Union, grant COSPAL (IST-2004-71567). However, this paper does not necessarily represent the opinion of the European Community, and the European Community is not responsible for any use which may be made of its contents.

new one is created with the new sample chosen as centroid.

In [12] Salzberg proposed the Nested Generalized Exemplar (NGE) using hyper rectangles as local models. In contrast to ARTMAP these are organized as stand-alone models attached with class labels. Models that respond to new samples either grow to integrate the sample into their representation in the case of correct classification or shrink to enlarge the distance between different classes. If there is no correct classification, a new model is instantiated. Note, that the possibility of shrinking is not shared by many other methods, but increases the algorithm's adaptivity.

Another well-known classification technique is the Learning Vector Quantization (LVQ) [7]. An incremental version (iLVQ) was presented by Kirstein in [6]. In LVQ models are represented by single data prototypes (centers). Classification results from a nearest-neighbor decision. Centers are updated by either pulling them towards the sample or pushing them away, depending on correct or false classification. The size of the impact on the center is determined by the product $\epsilon(t)(x_t - bmu_t)$ where x_t is the sample presented at time t, bmu_t is the corresponding *best matching unit* (responding center) and $\epsilon(t)$ is the learning rate modeled by a function decreasing over time, to decide the plasticity/stability dilemma in favor of stability. Also in the incremental version a new center is set to x_t, if no correct bmu can be determined w.r.t. a certain threshold. In contrast to standard LVQ, Kirstein made $\epsilon(t)$ individual to every center, so that each center has a different age depending on the center's number of positive classifications to ensure a balanced development of old and new centers.

The thresholds that have to be chosen in the aforementioned methods are responsible for the number and size of the models or the resolution of the classifier. Once they are set, the performance of the classifier and thus the quality of the chosen values can only be evaluated after the training phase. There is no on-line assessment that controls the development of models. This makes the correct choice of parameters comparably difficult and crucial like e.g. choosing k in k-means. In the next section we introduce the Local Credibility Criterion which reduces the importance of a perfect parameter selection.

3 LCC-Classification Algorithm

3.1 The Local Credibility Criterion

The Local Credibility Criterion is a mechanism to control the development of a classifier's local models by evaluating the classification rate on a local scale. The classifier's performance is measured for each individual model whenever a new training sample is presented, yielding very detailed information for the models to adapt to the topology of the training data.

The LCC is based on a set of parameters and the credibilities (classification rates) of the individual models. The credibility γ of a model m is defined by the ratio of the number of correct and total responses: $\gamma = R^c/R^t$. It is used for two purposes. First, a model's response is weighted by its credibility and second, together with the parameters the credibility controls the adaptation of the classifier i.e. the development of the local models (Section 3.2).

Before introducing the parameters of the LCC we define the symbols of the data set and the classifier as well as its equations. Given a data set $X = \{x_1, ..., x_N\}$ with $x_t \in \mathbb{R}^d$ and its labels $L = \{l_1, ..., l_N\}$ with $l_t \in \{1, ..., J\}$ we define a classifier for J classes C_j with $j \in \{1, ..., J\}$, where each class is represented by M_j local models m_{ji} with $i = \{1, ..., M_j\}$. Each model m_{ji} is defined by its centroid $c_{ji} \in \mathbb{R}^d$ and a weight matrix $w_{ji} \in \mathbb{R}^{d \times d}$ describing the range of its local domain. In our case we have chosen the models to be hyper spheres so that $w_{ji} = I \cdot w_0^2$, where w_0 is the initial radius of the hypersphere. The value for w_0 is estimated from the first samples that are available.

We then define the similarity s of sample x_t and model m_{ji} to be

$$s(m_{ji}, x_t) = 1 - (x_t - c_{ji})^T \cdot w_{ji}^{-1} \cdot (x_t - c_{ji}).$$

A positive similarity states that the sample lies inside the model's domain. The model's response is defined as

$$r(m_{ji}, x_t) = s(m_{ji}, x_t) \cdot \gamma_{ji},$$

the response of class C_j as

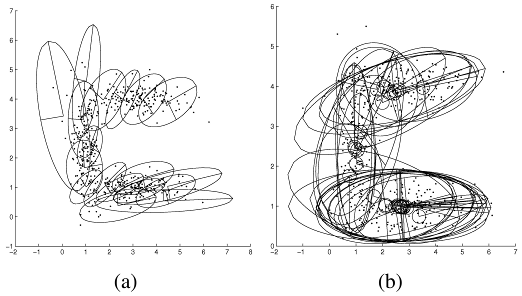
$$r(C_j, x_t) = \sum_{i=1,\, r(m_{ji}, x_t) > 0}^{M_j} r(m_{ji}, x_t)$$

and the result of the classifier as

$$r(x_t) = \arg_j \max r(C_j, x_t),\ j = 1, ..., J.$$

The parameters of the LCC control the creation or deletion of models and the growing or shrinking of their domains. A new model shall be added to class C_j if the response $r(C_j, x_t)$ for sample x_t with $l_t = j$, is below the threshold $\theta_{new} = 0.55$. The threshold for initializing a new model could be any positive number, but as the highest possible response of a model having full credibility and minimal distance to a sample is 1 motivates a value of less than 1 for θ_{new}. The deletion of a model depends on its credibility. If it falls below $\gamma_{del} = 0.3$ the model's classification rate is considered to be too low and the model is deleted. Also the thresholds for growing or shrinking a model's domain are credibility-orientated. A model is only allowed to extend if its credibility is at least $\gamma_{grow} = 0.97$ and should be shrunk if the credibility is below $\gamma_{shrink} = 0.5$. Furthermore, we define a growing factor $f_{grow} = 1.1$ and a shrinking factor $f_{shrink} = 1/f_{grow}$ to determine how much the range of the domain is altered in either case.

All numerical values are chosen arbitrarily just following the expectation that a model should have a certain credibility when stating that a sample belongs to its class. The fact that these values lead to good results in all our experiments as presented in Section 4 supports the assumption that the LCC reduces the problem of parameter selection significantly. In Section 4.2 we present a further discussion and experiments on the parameter selection.



(a) (b)

Figure 1. Model development following different assignment strategies: Updating only best (a) or all (b) matching models.

Algorithm 1: Train Classifier

```
1  foreach $x_t$ do
2      foreach $m_{ji}$ do with $j = 1...J, i = 1...M_j$
3          if $l_t = j$ then update model of same class
4              if $s(m_{ji}, x_t) \geq 0$ then
5                  $\gamma_{ji} = (R^c_{ji} + 1)/(R^t_{ji} + 1)$
6              else if $\gamma_{ji} \geq \gamma_{grow}$ then
7                  $w'_{ji} = w_{ji} \cdot f_{grow}$
8                  if $s(m'_{ji}, x_t) \geq 0$ then
9                      extend model
10                     $\gamma_{ji} = (R^c_{ji} + 1)/(R^t_{ji} + 1)$
11         else if $l_t \neq j$ then update model of
           different class
12             if $s(m_{ji}, x_t) \geq 0$ then
13                 $\gamma_{ji} = R^c_{ji}/(R^t_{ji} + 1)$
14             if $\gamma_{ji} \leq \gamma_{del}$ then
15                 delete model
16             else if $\gamma_{ji} \leq \gamma_{shrink}$ then
17                 $w_{ji} = w_{ji} \cdot f_{shrink}$
18         if $r(C_j, x_t) \leq \theta_{new}$ then
19             initialize new model for class $C_j$ with
               $j = l_t$
```

3.2 The Training Algorithm

When training the classifier all samples x_t are presented individually. For each x_t all models m_{ji} are updated either following the steps in lines 4...10 in Algorithm 1 or lines 12...17 if the model is of the same class as the sample or not, respectively. In both cases it is first determined whether the model responds to the sample or not by computing the similarity $s(m_{ji}, x_t)$. If it responds the model's credibility is increased for being of the same class or decreased otherwise. For non-responding models of the correct class whose credibility allows for growing (i.e. $\gamma_{ji} \geq \gamma_{grow}$) it is further checked whether they would respond if their radius was enlarged by the growth factor f_{grow}. In the case they do respond they are updated by moving the center towards the new sample, enlarging the radius and updating the credibility. Otherwise, no action is taken. For responding models of false classes it is checked whether decreasing their credibility has lead to the case that it has fallen below either the shrinking or deleting threshold γ_{shrink} or γ_{del}, respectively. In either case the appropriate action is taken (cf. Algorithm 1 lines 17 and 15). After updating all models the response for class C_j with $j = l_t$ is computed and for the case that it is below θ_{new} , a new model is added to C_j with the initial values set to $c_{ji} = x_t$, $w_{ji} = I \cdot w_0$, $R^c_{ji} = 1$, $R^t_{ji} = 1$ and $\gamma_{ji} = 1$ with $i = M_j + 1$.

3.3 Benefits of the LCC

In this section the features supported by the LCC are discussed. For robustness w.r.t. to the order of the training samples we adopted the key idea of our incremental clustering algorithm presented in [10]. In contrast to the more often applied winner-takes-all strategy, we suggest to update all responding models. Thus, without competition the models do not bar each other from aligning with the topology of the data. For making the effect even more apparent, in Figure 1 we give an example of incremental clustering using hyper ellipsoids as models. The fact that without competition eventually almost every model converges to cover the same domain as its neighbors proves that the order of training samples has lost most of its significance and makes the resulting classifier more reproducible. Another logical consequence is also the resulting redundancy which additionally allows for an improvement of the computational complexity as less of these well aligned models are needed. In the LCC we do not allow competition between models of the same class but between those of different classes. Thus, models can develop freely within the domain of their own class, but their growth is stopped at the border to the next class when they start responding to samples of different classes and their credibility sinks below the growing threshold θ_{grow}.

The main guidance for the models' adaptation is their own performance. The thresholds of the LCC only influence the number of training samples needed to converge and the resolution of the classifier i.e. the number of models. In other words, the impact of the thresholds is buffered by an internal control mechanism mainly steered by the classification rate. So in general, the parameters only have a small effect on the final performance making the LCC-Classifier robust w.r.t. their selection.

This internal control mechanism offers even more advantages: Although models cannot excessively grow into domains of other classes, they can be instantiated in regions that are covered by foreign models. This is because models react to the real data samples but not to the existing model domains which might be outdated. LCC-models are also able to withdraw themselves from falsely covered domains by either shrinking or being replaced by smaller more accurate ones. Thus, resolution is automatically increased in uncertain domains of the input space. In Figure 2, the process of model refinement at class borders is de-

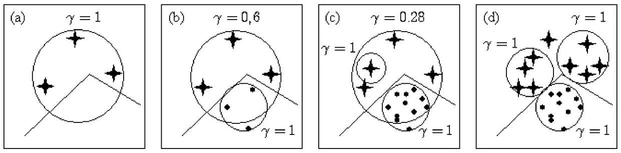


Figure 2. LCC controls resolution adaptation at class borders. (a) In absence of circle samples the model for stars expands over the class border. (b) Responding to circle samples reduces the credibility of the star model. (c) The support for star samples is below θ_{new}, therefore a new model is instantiated. (d) The first star model has been deleted as its credibility was below γ_{del}.

picted in an idealized scene. In Figure 2(a) the star class, in absence of counter examples, has developed a big model that already significantly crosses the class border. In Figure 2(b), a new model is created for the first circle samples. As also the star model responses, its credibility decreases making it more likely for all samples to be correctly classified. With more circle samples falling into the domain of the star model (Figure 2(c)), its response to the new star sample is lower than θ_{new}, so that a new model is created. Finally, the first star model's credibility is below the vanishing threshold causing it to give room for new models (Figure 2(d)). For simplicity we omitted the possibility of shrinking in this example.

The ability of the LCC-Models to abandon covered domains or to grow into foreign domains enables the classifier to handle non-stationary data concepts, where the domains of classes change over time. And although, the integral character of the credibility works towards stability of the classifier it is still able to adapt quickly to a constant concept drift. That is because new models have maximum credibility and ad once insure correct classification even in a domain covered by another model. In the case of an outlier the new model will vanish almost as fast as it was created, so that disturbance is kept low. But, in the case of concept drift it will mature and eventually replace the other. See Section 4.3 for experimental results on non-stationary data sets.

4 Experimental Results

In this section we analyze the influence of the order of training samples, and whether selecting the algorithm's parameters is critical. We also demonstrate the method's ability to cope with drifting data concepts and its competitive performance on some known benchmark tests.

4.1 Influence of Sample Order

The following experiment supports our hypothesis that the allocation and adaptation of models becomes more robust w.r.t. the order of the training samples using the no-competition policy. We used three benchmark data sets (pendigits [4], twonorm [11] and the circle-in-square) and split them into training (70%) and testing (30%) sets. Then we produced 100 different permutations of the training sets, trained a classifier on each and tested it on the testing set. In Figure 3 it can be seen that the performance of the resulting classifiers stays very constant for each data set.

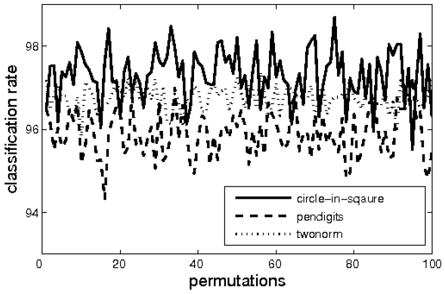


Figure 3. Irrespective of the order of training samples the classification rate stays on a high level.

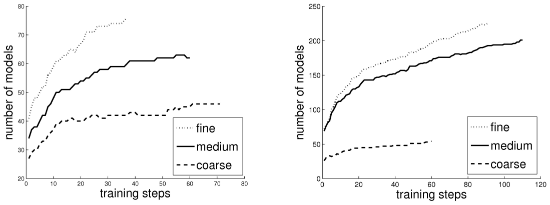
	θ_{new}	f_{grow}	γ_{grow}	γ_{shrink}	γ_{del}
fine	0.8	1.05	0.99	0.7	0.2
medium	0.55	1.1	0.97	0.5	0.3
coarse	0.4	1.2	0.95	0.4	0.4

Table 1. Three parameter configurations to produce a fine, medium or coarse resolution of models.

4.2 Parameter Selection

Many algorithms react very sensitive to small changes of their parameter values which makes finding the right configuration a difficult task. Therefore, we analyzed the parameters' influence on the classifier's topology and its performance. We set up three different configurations of the method's thresholds (Table 1). The first one fosters a finer resolution i.e. the development of more but smaller models, the second is our usual setup and the third one encourages a more coarse resolution having fewer but larger models. Considering the meaning of the thresholds, it is easy to understand that low shrinking γ_{shrink} and deleting γ_{del} thresholds let models persist longer and stay larger. A low growth threshold γ_{grow} lets models grow more often and a high growth factor f_{grow} makes them grow faster. Fewer initializations of new models can be achieved by choosing a low value for θ_{new} making the adaptation process more tolerant to uncertainty.

Using these configurations we trained three classifiers incrementally starting with 1% of the training set and updating the classifier with 0.05% each step until a prespecified performance (97%) on a disjoint testing set is achieved. We performed this test on the twonorm and a circle-in-square data set. In Figure 4 one can clearly see that the configuration for a finer/coarser resolution always yields a classifiers with more/less models than the setting for medium resolution. By steering the number of models, one has an instrument to influence the computational complexity as it is linearly proportional to the number of mod-



(a) (b)

Figure 4. Results of training three classifier with different parameter configurations, showing that one can tune the resolution of the classifier (number of models). Irrespective of the configuration, all classifiers achieve the desired performance. Comparing the results on the circle-in-square (a) and the twonorm (b) data, there is no tendency for any configuration to converge in fewer steps.

els and to decide on the degree of generalizability. There is no clear tendency which configuration converges faster as this can be different from case to case. But the more important fact is that the classifier always reaches the desired performance (if it is chosen realistically for the specific data set). This proves that the LCC-Classifier is not sensitive to the parameter selection which makes it easy to use, even without special expertise of its inner workings.

4.3 Non-Stationary Data Sets

As non-stationary data we created a two dimensional ring-shaped data set that can be rotated around its center. The data set is divided into two classes, the upper half containing 608 and the lower half containing 632 samples. For showing the LCC-Classifier's capabilities with drifting data concepts two experiments were conducted on this set. In the first one the data set is rotated into one direction up to 90 degrees. The experiment was repeated for drifting rates of 1 and 10 degrees per rotation step. In the second experiment the data oscillates i.e. alternately takes five steps of one degree into either direction. For both experiments the classifier was updated on each rotation step with 70% of the data and tested on the remaining 30%. Figure 5 shows the results for the constantly drifting data. Although this is a quite strong distortion of the data concept, classification rate stays on a high level and to the end the number of models increases slower as models start to drop out when their credibility falls below γ_{del}. In the case of oscillating data (Figure 6) the classification rate stays even higher and the number of models converges. This is due to the fact that new models are created until resolution in the uncertain regions has reached its optimum. Deletion of models is rather unlikely as models periodically get positive feedback.

4.4 Benchmark Tests

For comparing the LCC-Classifier to other algorithms we have chosen the widely used iris [4], pendigits and the twonorm data sets. On the iris data set we compared our method with two batch-learning algorithms, a multi-class

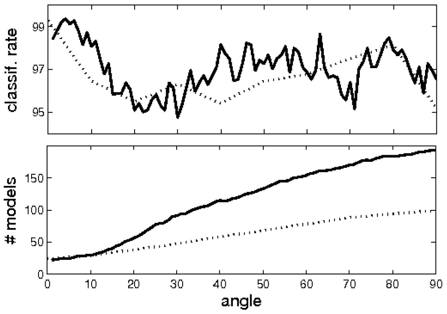


Figure 5. Classification results for rotating dataset. Classification rate and number of models for rotating dataset at 1 degree (solid line) and 10 degrees (dotted line) at each step.

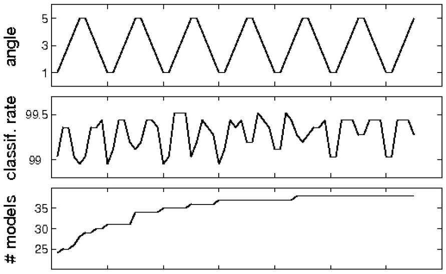


Figure 6. Classification results for rotating dataset. Angle, classification rate and number of models when rotating ring-shaped dataset with alternating direction of rotation.

SVM and a kNN method employing Mahalanobis distance [13], and an incremental classifier, the Fuzzy Generalized Exemplar System (FuGES) [8]. As the other authors we performed 100 runs on different randomly generated 70/30 (training/testing) splits of the data. The iris data set contains 150 samples of 4 dimensions equally distributed in 3 classes. In Table 2 it can be seen that our algorithm shows comparable and good results.

The pendigits data set contains 10992 16 dimensional feature vectors of handwritten digits. Table 3 shows our results compared to two methods based on boosting and bagging: Nearest-Neighbor Classifiers (NNC) by [1] and Learning Ensembles (LE) by [3]. Like [1] and [3], we also used 10-fold cross-validation.

The twonorm data set is an artificial 20 dimensional two class classification example being frequently used as a benchmark. It consists of 7400 instances. For comparison (Table 4) we use the results of the C2-SDMO method presented in [9] which is an incremental version of a support vector machine and a reference value from a benchmark repository[2]. Again we use the same number of runs (100), training (400) and testing samples (7000).

5 Conclusion

In this paper we proposed the Local Credibility Criterion which is a novel technique for incremental classifiers. By

[2] Benchmark Repository: http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm

	runs	train / test	classif. rate
multi-class SVM	100	105 / 45	95.6
LCC-Classifier	100	105 / 45	95.4
kNN Mahal.-dist.	100	105 / 45	95.3
FuGES	?	105 / 45	93.3

Table 2. Classification results on iris data set.

	condition	classif. rate
LCC-Classifier	10-fold CV	97.8
LE	10-fold CV	96.4
NNC	10-fold CV	96.1

Table 3. Classification results on pendigits data set.

measuring the classification rate of the local models the LCC ensures that the models do not cross the borders between classes, but allows them to develop freely within the domain of their own class. We could show that this control mechanism has two major advantages: First, the fact that there is no competition between models of the same class allows all models to adopt the topology of the data. Thus, the dependency on the order of the training samples is reduced significantly. Second, the difficulty of selecting good parameter values for a high classification performance could be minimized as well. Parameters of the LCC do not directly influence the classification performance. This is controlled by the LCC automatically by measuring the classification rate of the individual models. Via the parameters one can only influence the number and size of the models, which is far less critical. Besides these major contributions, we could also show the competitive performance of our method on some well-known benchmark tests. In contrast to many other algorithms that only extend the coverage of their classes, LCC models have also the ability to shrink or vanish allowing classes to move their domains and follow drifting data concepts.

Future work will include extending the local models to hyper ellipsoids to fully exploit the benefits of the no-competition policy. Thus, a method to reduce the redundancy of overlapping models becomes even more desirable.

References

[1] V. Athitsos and S. Sclaroff. Boosting nearest neighbor classifiers for multiclass recognition. In *Workshop on Learning in Computer Vision and Pattern Recognition*, pages III: 45–45, 2005.

[2] G. A. Carpenter, S. Grossberg, N. Markuzan, J. H. Reynolds, and D. B. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Trans. On Neural Networks*, 3(5):698–712, 1992.

[3] Nitesh V. Chawla, Lawrence O. Hall, Kevin W. Bowyer, and W. Philip Kegelmeyer. Learning ensembles from bites: A scalable and accurate approach. *Journal of Machine Learning Research*, 5:421–451, 2004.

[4] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[5] H.Xu and M. Vuskovic. Mahalanobis distance-based artmap network. In *International Joint Conference on Neural Networks*, volume 3, pages 2353–2359, Budapest, Hungary, 2004.

[6] S. Kirstein, H. Wersing, and E. Korner. Rapid online learning of objects in a biologically motivated recognition architecture. In *German Pattern Recognition Symposium*, pages 301–308, Wien, Austria, 2005.

[7] Teuvo Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, third edition, 1989.

[8] Flavia O. S. Sa Lisboa, Maria do Carmo Nicoletti, and Arthur Ramer. A fuzzy classifier system based on generalized exemplars. In *FUZZ-IEEE*, pages 73–76, Melbourne, Australia, 2001.

[9] Thomas Martinetz, Kai Labusch, and Daniel Schneegaß. Softdoubleminover: A simple procedure for maximum margin classification. In *International ConferenceArtificial Neural Networks*, volume 3697, pages 301–306, Warsaw, Poland, 2005.

[10] H. Prehn and G. Sommer. An adaptive classification algorithm using robust incremental clustering. In *International Conference on Pattern Recognition*, volume 1, pages 896–899, Hongkong, 2006.

[11] Carl Edward Rasmussen, Radford M. Neal, G. E. Hinton, D. van Camp, M. Revow, Z. Ghahramani, R. Kustra, and Robert J. Tibshirani. *The DELVE Manual*. University of Toronto, 1.1 edition, 1996.

[12] Steven Salzberg. A nearest hyperrectangle learning method. *Machine Learning*, 6(3):251–276, 1991.

[13] Kilian Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. In Y. Weiss, B. Schőlkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1473–1480. MIT Press, Cambridge, MA, 2006.

	runs	train / test	classif. rate
C2-SDMO	100	5% / 95%	97.6
Reference	100	5% / 95%	97
LCC-Classifier	100	5% / 95%	96.8

Table 4. Classification results on twonorm data set.