http://www.newtheory.org

Journal of New Theory

Received: 26.11.2014 *Year*: **2015,** *Number*: **1**, *Pages*: **17-23**
Accepted: 05.12.2014 *Original Article*[**]

A Novel Supervised Learning Based on Density

Umut Orhan[1,*] (uorhan@cu.edu.tr)
Mahmut Hekim[2] (mahmut.hekim@gop.edu.tr)

[1]*Cukurova University, Department of Computer Engineering, 01130 Adana, Turkey*
[2]*Gaziosmanpasa University, Department of Electrical and Electronics Engineering, 60240 Tokat, Turkey*

Abstract – Because prototype based classifiers are both easy and reasonable methods, there have been many studies on similarity based supervised learning. In order to detect each class region, they should not only appropriately locate the prototypes, but also deal with overfitting and instability. In this study, by considering all these criteria, we develop a new classifier method based on the prototypes selected from dense patterns. While the method determines details of the prototypes, it evades overfitting according to relation of the correct classification accuracy and the number of prototypes. Because of its similarity in point of architecture, we compare it with learning vector quantization (LVQ) method by using some synthetic and benchmark datasets. This comparison shows that our method is better than the other, and it may cause new suggestions on classification and some real applications.

Keywords – *Supervised learning, prototype classifier, learning vector quantization (LVQ), overfitting.*

1. Introduction

The simplest supervised learning method classifies samples based on similarity according to their class labels. Because prototype based classifiers are both easy and reasonable methods, there have been many studies on similarity based supervised learning. Each prototype represents a group of patterns with the same class. There are different viewpoints to prototype term. In the some of the approaches, each pattern acts as a prototype, but the approaches using fewer prototypes are more widespread. Learning Vector Quantization (LVQ) which finds prototypes using cluster analyze [7] and self-generating neural tree [19] are well-known prototype classifiers. Some other prototype classifiers are hyper-spheres [13], hyper-ellipsoids [8] and hyper-rectangles [14] based methods.

[**] *Edited by* Naim Çağman (*Editor-in- Chief) and* Mehmet Akar *(Area Editor).*
[*] *Corresponding Author.*

In some studies, special terms are used instead of prototypes. For instance Expectation Maximization (EM) and Gaussian Mixture (GM) models use components instead of prototypes in estimation of the class densities [6]. Minimum enclosing axis-parallel boxes in the method of Kudo et al. [9] and Takigawa et al. [15] represent also prototypes. In two algorithms offered by Takigawa et al. [16], convex balls are regarded as prototypes. Besides some researchers proposed some methods on logical analysis of box-based data [1, 4], ball-based combinatorial classifier [2, 3, 10, 12], and support vector machines (SVM) [18]. SVM is a classifier which selects the vectors on the margin of the classes. If the selected vectors are considered as the prototypes, SVM can be regarded as a prototype based classifier. In contrast to other prototype classifiers, SVM selects the weakest patterns as prototypes. Fayed et al. [5] suggest starting with one prototype for each class, assigns patterns into prototypes, and reduces prototypes.

The commonest handicaps of all prototype based methods are determining the number and starting positions of prototypes. In order to prevent these disadvantages, the most methods are run for different numbers of prototypes. In this paper, we offer a novel supervised learning algorithm called supervised learning based on the prototypes selected from dense patterns (SLDP) which no need running more times.

The paper is organized as follows: section 2 describes the structure of new method, section 3 presents some applications on some artificial and real datasets to verify the effectiveness of the proposed method, and finally, section 4 gives the conclusions.

2. Supervised Learning Based on the Prototypes Selected from Dense Patterns

A dataset usually includes many regions which have different densities based on the distances among the patterns. Dense regions within each class can be symbolized by the prototypes.

Inspired by gravity, we suppose that each pattern has potential energy, and this energy (or weight) can be computed by using neighborhood among patterns. If we accept that each pattern has unit mass [11], the potential weight can be easily calculated as follow.

$$w_j = \sum_{i=1}^{n} \frac{1}{\left\| x_i - x_j \right\|^2}, \quad \text{for } x_i, x_j \in C_k \text{ where } k \in \{1,2,...,c\} \qquad \textbf{(1)}$$

where n is the number of patterns, $\left\| x_i - x_j \right\|$ is Euclidean distance between patterns x_i and x_j, C_k is class k and c is the number of classes in the dataset.

The pattern with the maximum potential weight is selected as a prototype. The location (x_i) and the potential weight (w_i) of the selected pattern are assigned the location (X_i) of the prototype and its absolute weight (W_i) respectively. According to the second step of the study, the classification process is operated by the determined prototypes. The classification effect of each class (C_k) on a new pattern (Y) is calculated by

$$\varepsilon_k = \sum_j \frac{W_j}{\left\| X_j - Y \right\|^2}, \quad \text{for all } X_j \in C_k \text{ where } k = 1, 2, \ldots, c \tag{2}$$

where *j* values list is increased as new prototypes are discovered. Then the class of the new pattern (*Z*) can be estimated by

$$Z = \arg max(\varepsilon_k), \text{ for } k = 1, 2, \ldots, c \tag{3}$$

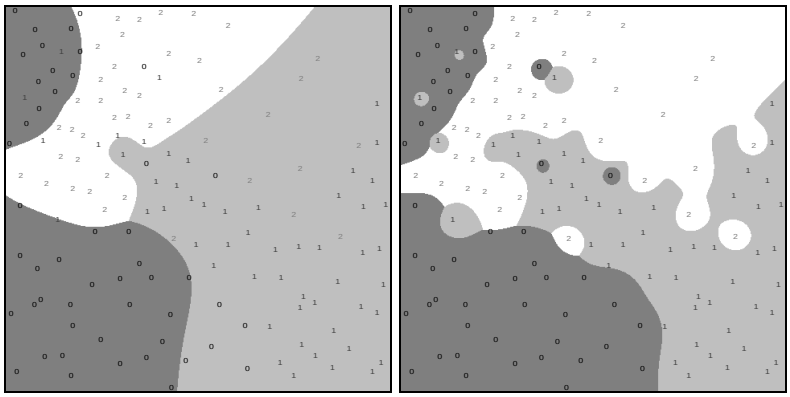
where *Z* is the estimated class of pattern *Y*. In training process, the pattern is identified as a misclassified pattern, if its desired class is not the same with *Z*. Then, the one with maximum potential weight among the misclassified patterns is selected as a new prototype. Training process stays in this loop until the gradient between number of prototypes and misclassification error get less than a predefined threshold value. Our learning algorithm consists of the following steps.

Step 1. Calculate w_j using Equation 1, and choose one prototype for each class.
Step 2. For each pattern in dataset, calculate $\varepsilon_{i=1,..,k}$ using Equation 2, and decide *Z* by Equation 3. If $Z \neq C_Y$, signify *Y* as a misclassified pattern.
Step 3. Compute the gradient between misclassification error and the number of prototypes. If it is less than a predefined threshold value, stop the algorithm.
Step 4. Set the misclassified pattern with the maximum weight as new prototype.
Step 5. Go to Step 2.

Unlike a common LVQ network, the locations and weights of the prototypes does not change in SLDP. In each iteration of the training process, only one prototype is discovered.

3. Numerical Results and Comparisons

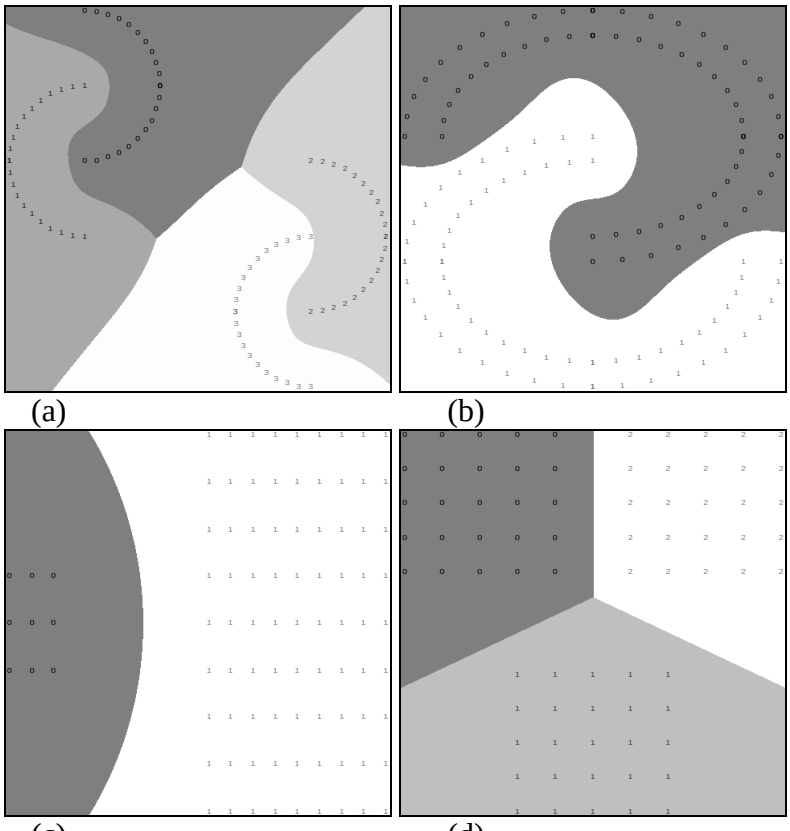
In the numerical experiments, we use some synthetic and real datasets. To show behaviors of the proposed method, five synthetic datasets are preferred as two dimensional, and to prove success of it, four real datasets are chosen from multidimensional benchmark datasets. The classification behaviors of the method are illustrated in Figure 1(a) and 1(b). In the first experiment, the method classifies an asymmetric dataset with 171 patterns. In Figure 1(a), the algorithm reaches success 81.38% with 12 prototypes by avoiding overfitting. As seen in Figure 1(b), if the algorithm does not consider overfitting, it can reach 100% success with 44 prototypes.



(a) (b)

Figure 1. The classification maps of SLDP for a dataset including outliers (a) by avoiding overfitting (b) by not avoiding overfitting.

Frequently, high success brings to mind overfitting. The success of the algorithm is 81.38% for 12 prototypes in Figure 1(a). Even though 32 new prototypes are discovered, the success increases only 18.62%. This increase values cannot be accepted as consistent.
We have also prepared four synthetic datasets which are discrete, complex, symmetric and asymmetric. Figure 2 shows the classification maps of the method for these four synthetic datasets.



(a) (b)

(c) (d)

Figure 2. The classification maps of SLDP for synthetic datasets with (a) 4 symmetric classes (b) chain shaped 2 classes (c) 2 asymmetric classes (d) 3 symmetric classes

In order to compare our algorithm with some other methods, we applied it to some benchmark datasets listed in Table 1 [17].

Table 1. Summary of datasets.

Dataset	Instances	Attributes	The number of classes
Iris	150	4	3
Parkinson	195	22	2
Spect Heart	267	44	2
Statlog (Landsat)	6435	36	6

The numbers of prototypes found by SLDP for Iris, Parkinson, Spect Heart, and Statlog are 5, 6, 3, and 7 respectively. They are also used as the number of prototypes of LVQ for real datasets. For example, the new method finds 1, 2, and 2 prototypes for each class of Iris dataset. LVQ is started with the same distribution of prototypes for each class. Table 2 shows the correct classification values of LVQ and SLDP for real datasets.

Table 2. The correct classification values of LVQ and SLDP for real datasets.

Datasets	**LVQ** (%)	**SLDP** (%)
Iris	95,33	96,00
Parkinson	60,00	88,72
Spect Heart	53,18	81,27
Statlog (Landsat)	20,45	68,47

The novel method reaches the same result each time. But LVQ is run 500 times and 500 iterations for each dataset and the highest results are selected for this comparison. As seen in Table 2, the most successful method is SLDP.

There is no method which is able to reach high success for every dataset without overfitting. The relation between the misclassification error and the number of prototypes is very important in dealing with overfitting. In Figure 3, we can see the regions with overfitting for four real datasets.

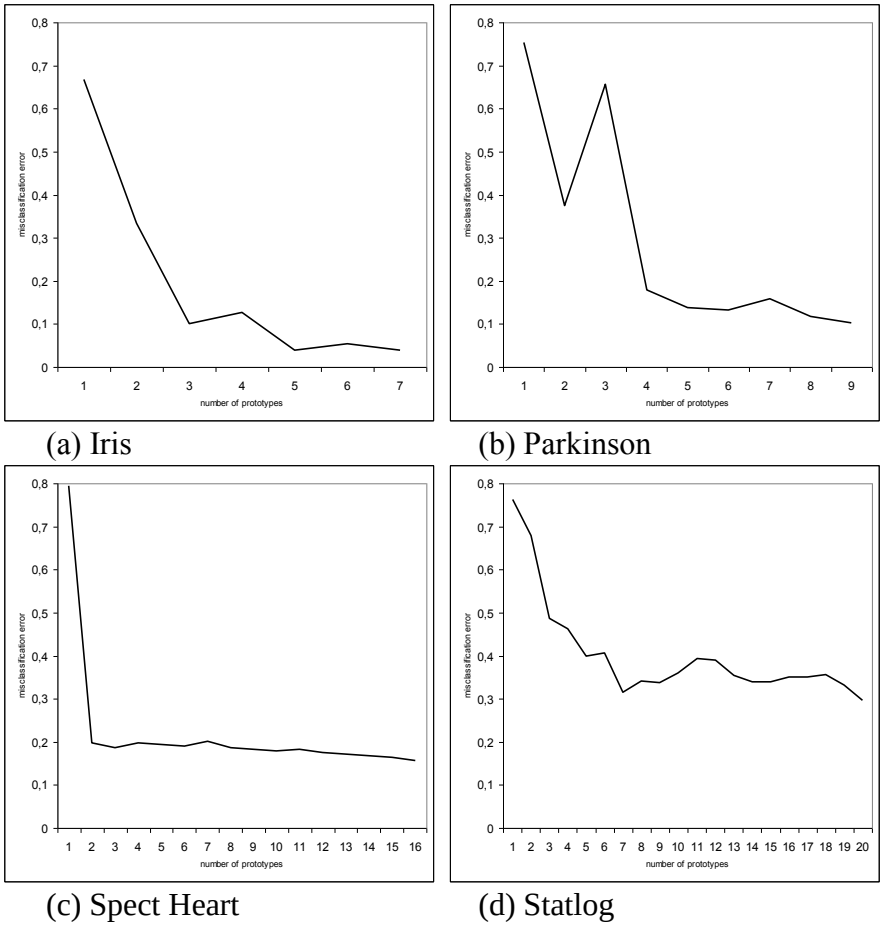


Figure 3. The relation between misclassification error and the number of prototypes for real datasets.

In Figure 3, if gradient of the relation is less than 0.01, overfitting starts. In this situation, the algorithm must be stopped. Otherwise, it will continue overfitting. Thus there is no learning after 5 prototypes for Iris, 6 prototypes for Parkinson, 3 prototypes for Spect Heart, and 7 prototypes for Statlog.

4. Conclusion

In this paper, we proposed a new method called supervised learning based on the prototypes selected from dense patterns (SLDP). While the method learns the number of prototypes and its locations by avoiding overfitting, it determines prototypes by using the potential weights of each pattern. Its other two advantages are that it does not depend on the sequence of patterns in a dataset and does not require any input parameters. The method also offers a new approach to control overfitting. Although it needs many more experiments, we hope that SLDP will be a source of inspiration for new methods.

References

[1] G. Alexe, P.L. Hammer, *Spanned patterns for the logical analysis of data*, Discrete Applied Mathematics 154-7 (2006) 1039–1049.

[2] A.H. Cannon, , L.J. Cowen, *Approximation algorithms for the class cover problem*, Annals of Mathematics and Artificial Intelligence 40 (2004) 215–223.

[3] A.H. Cannon, J.M. Ettinger, D. Hush, C. Scovel, *Machine learning with data dependent hypothesis classes*, Journal of Machine Learning Research 2 (2002) 335–358.

[4] J. Eckstein, P.L. Hammer, Y. Liu, M. Nediak, B. Simeone, *The maximum box problem and its application to data analysis*, Computational Optimization and Applications 23-3 (2002) 285–298.

[5] H.A. Fayed, S.R. Hashem, A.F. Atiya, *Self-generating prototypes for pattern classification*, Pattern Recognition 40 (2007) 1498-1509.

[6] T. Hastie, R. Tibshirani, J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, Springer, Stanford, CA, 2001.

[7] T. Kohonen, *Self organizing maps*, Springer Series in Information Sciences, Berlin, Springer Verlag, 2001.

[8] M. Kositsky, S. Ullman, *Learning class regions by the union of ellipsoids*, Proceedings of the 13th International Conference on Pattern Recognition, 1996, Volume 4, 750–757.

[9] M. Kudo, S. Yanagi, M. Shimbo, *Construction of class regions by a randomized algorithm: a randomized subclass method*, Pattern Recognition 29 (1996) 581–588.

[10] D.J. Marchette, *Random Graphs for Statistical Pattern Recognition*, Wiley, NewYork, 2004.

[11] U. Orhan, M. Hekim, T. Ibrikci, *Supervised gravitational clustering with bipolar fuzzification*, Lecture Notes in Artificial Intelligence, ICIC 2008, 5227, 667-674.

[12] C.E. Priebe, D.J. Marchette, J.G. DeVinney, D.A. Socolinsky, *Classification using class cover catch digraphs*, Journal of Classification 20 (2003) 3–23.

[13] D. Reilly, L. Cooper, C. Elbaum, *A neural model for category learning*, Biological Cybernetics 45 (1982) 35–41.

[14] S. Salzberg, *A nearest hyperrectangle learning method*, Machine Learning 6 (1991) 277–309.

[15] I. Takigawa, N. Abe, Y. Shidara, M. Kudo, *The boosted/bagged subclass method*, International Journal of Computing Anticipatory Systems 14 (2004) 311–320.

[16] I. Takigawa, M. Kudo, A. Nakamura, *Convex sets as prototypes for classifying patterns*, Engineering Applications of Artificial Intelligence 22 (2009) 101-108.

[17] *UCI Machine Learning Repository*, http://archive.ics.uci.edu/ml/ (Dec 1, 2008).

[18] V.N. Vapnik, *Statistical learning theory*, Wiley, New York, 1998.

[19] W.X. Wen, A. Jennings, H. Liu, *Learning a neural tree*, International Joint Conference on Neural Networks 1992, Beijing, 751–756.