Computational Mathematics and Mathematical Physics, Vol. 45, No. 7, 2005, pp. 1276–1282.
Translated from Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, Vol. 45, No. 7, 2005, pp. 1321–1328.

Convex Cluster Stabilization of Classification Algorithms as a Means for Finding Collective Solutions with High Generalization Ability

D. P. Vetrov and D. A. Kropotov

Computing Center, Russian Academy of Sciences, ul. Vavilova 40, Moscow, 119991 Russia
e-mail: vetrovd@yandex.ru, dkropotov@yandex.ru

Received September 13, 2004

Abstract—A collective solution method in pattern recognition based on the simultaneous improvement of the stability and efficiency (the percentage of correctly classified objects in the learning sample) is generalized. The relationship between the procedure described in the paper and several available methods for constructing collective algorithms that are particular cases of a more general approach is revealed. The practical value of the method is confirmed by solving some well-known classification problems.

Keywords: pattern recognition, collective solutions, correction of algorithms, stability of classifiers.

1. INTRODUCTION

There are many parametric families of pattern recognition algorithms. When a particular practical problem is solved, it is usually not clear which algorithm is better suited for this problem. Often, a combination of the results obtained by several methods improves the quality of the recognition. For this reason, methods for the construction of collective solutions over several families of recognition algorithms (see [1, 2]) and methods for the correction of algorithms (see [3, 4]) have received widespread use. It is worth noting that the correction methods can be used to produce collective solutions in the framework of the same (maybe, large) family of algorithms.

The use of algorithms belonging to different families of classifiers for constructing a collective solution makes it possible to overcome the limitations of a particular parametric model. However, this approach involves considerable difficulties that stem from the fact that the results produced by different algorithms must be unified and then combined. Moreover, the resulting collective solution may suffer from overfitting, especially, if its dependence on the underlying algorithms is complicated. Therefore, some other factors, in addition to the efficiency, must be taken into account when constructing the collective solution. Indeed, the ultimate goal of pattern recognition is to minimize the generalization error, i.e., to minimize the probability of the incorrect classification of an arbitrary object from the universal set. In recent years, advances have been made in the evaluation of the relationship between the efficiency and the generalization error; this progress is based on the concept of the classifier capacity (VC dimension [5]). Despite the fundamental nature of the statistical learning theory based on this concept, the estimates based on the capacity of classifiers have certain drawbacks, for example, the excessive size of training samples, pessimistic estimates of the generalization error, and difficulties in determining the capacity of complicated classifiers. Moreover, when the sets of algorithms belonging to different parametric families are used, it seems impossible to find the capacity of the resulting collective solution. Therefore, other characteristics are needed that enable one to at least indirectly evaluate the generalization ability of particular algorithms and their combinations. One of such characteristics is the requirement of stability of an algorithm. This means that when the features of an object slightly vary, the classification results should also change insignificantly. Note that this is not the only possible definition of stability. For example, in the Bayesian regularization of machine learning, one of the restrictions is the requirement for the classification results to change insignificantly when the parameters of the recognition algorithm slightly vary (see [6, 7]). When collective solutions are constructed, this approach seems difficult to implement because the number of parameters of the decision rules is very large (these parameters include the parameters of each algorithm in the set and the parameters of the collective solution itself). Moreover, the influence of these parameters can be different in different regions of the feature space. Note that the stability condition inevitably requires that the output of the classification algorithms be represented in the form of estimates for classes. However, it was shown in [3] that the requirement

to have intermediate estimates for classes, which is needed for the correction of recognition algorithms as well, is not very stringent. This is because any classification method can be represented as the composition of a recognizing operator that possesses the desired properties and a decision rule that produces the index of the class to which the object is assigned by the algorithm.

In [8], a method of the convex stabilization of algorithms was suggested, and theoretical results concerning the construction of a collective solution satisfying the correction algorithm properties were obtained. The basic concepts of convex stabilization are presented in the following section. In Section 3, we propose a new method of convex cluster stabilization that is a generalization of the convex stabilization method. In Section 4, results of comparative experiments are presented and discussed.

2. CONVEX STABILIZER

We consider the following pattern recognition problem. Suppose there is a sample of m objects each of which belongs to one of l nonoverlapping classes. Each object $\{x_i\}_{i=1}^m$ is considered as a vector in the n-dimensional space of features. Given a new object, we want to assign it to one of the l classes in such a way that the classification conforms to the given sample of objects in the best possible way. We assume that all the recognition algorithms considered in this paper can be represented in the form

$$A(x) = \{P(\omega_j|x)\}_{j=1}^l = \mathbf{P}(\boldsymbol{\omega}|x). \tag{1}$$

In other words, every algorithm produces posterior probabilities of the fact the object belongs to the corresponding class. Note that not all algorithmic models operate in terms of probabilities. However, it is usually possible to adequately represent the results produced by an algorithm in probabilistic terms (for example, for the algorithms based on the construction of a hyperplane in the space of features, one can take the logistics function of the distance between the object and the hyperplane). Below, we assume that the features are somehow normalized (for example, have a unit variance).

Definition 1. The instability of the pattern recognition algorithm A at the point x is defined as

$$Z_A(x, \varepsilon) = \sum_{j=1}^{l}\left[P\left(\omega_j \mid x + \varepsilon\sum_{i=1}^{n}\mathbf{e}_i\right) - P(\omega_j \mid x)\right]^2,$$

where $\mathbf{e}_i$ is the unit vector of the corresponding coordinate in the feature space.

Thus, the instability is a characteristic of the variability of the classification results depending on the coordinates of the object to be classified. For sufficiently small ε, it is approximately given by the rule

$$Z_A(x, \varepsilon) \approx \varepsilon^2\|\nabla\mathbf{P}(\boldsymbol{\omega} \mid x)\|^2.$$

Assume that there are p classifiers $A_1, \ldots, A_p$, which may belong to different families of algorithms. Assume that these algorithms produce posterior probabilities for classes. To construct a collective solution, we use a control sample $\{y_k\}_{k=1}^q$, which may coincide with the training sample. Note that the use of an additional sample for improving the quality of trained algorithms is a widely used technique (see, for example, the pruning sets used in the postprocessing of decision trees in [9]).

Define

$$\Theta(k) = \{t \mid A_t \text{ correctly classifies } y_k\},$$

$$R(k) = \arg\min_{t\in\Theta(k)} Z_{A_t}(y_k), \quad T(k) = \arg\min_{t\in\{1,2,\ldots,p\}} Z_{A_t}(y_k),$$

$$P(k) = \begin{cases} R(k) & \text{for } \Theta(k) \neq \emptyset, \\ T(k) & \text{otherwise.} \end{cases}$$

Definition 2. The algorithm A is said to be obtained from the algorithms $A_1, \ldots, A_p$ by using the *convex*

stabilizer if it can be represented by the following convex combination of the initial algorithms:

$$P_A(\omega_j \mid x) = \frac{\sum_{k=1}^{q} w_k(x) P_{A_{F(k)}}(\omega_j \mid x)}{\sum_{k=1}^{q} w_k(x)}. \tag{2}$$

Here, $F : \{1, 2, \ldots, q\} \longrightarrow \{1, 2, \ldots, p\}$ is a function that determines the index of the "best" classification algorithm for each object in the control sample and $w_k : \mathbb{R}^n \longrightarrow \mathbb{R}$ are weighting functions possessing the following properties:

$$\begin{gathered} w_k(x) \geq 0 \quad \forall k = 1, 2, \ldots, q, \\ w_k(x) \longrightarrow 0 \text{ as } \rho(x, y_k) \longrightarrow \infty, \\ \frac{w_k(x)}{\sum_{j=1}^{q} w_k(x)} \longrightarrow 1 \text{ as } \rho(x, y_k) \longrightarrow 0. \end{gathered} \tag{3}$$

It was proposed in [8] to use $F(k) = P(k)$ and to define the weighting functions by

$$w_k(x) = \begin{cases} 1, & \rho(x, y_k) \leq \varepsilon, \\ 0, & \exists y_i \neq y_k : \rho(x, y_k) \leq \varepsilon, \\ \dfrac{1}{\rho(x, y_k) - \varepsilon}, & \rho(x, y_k) > \varepsilon \quad \forall y_i. \end{cases}$$

The convex stabilization turned out to be very efficient for solving problems with small and very small samples (see [8]). However, the application of this stabilizer for problems that involve samples consisting of a hundred or more objects is computationally inefficient. Moreover, a large control sample causes an excessive "switching" between the algorithms, which makes the collective solution unstable despite all the efforts.

3. CONVEX CLUSTER STABILIZATION

The convex stabilization is based on the two following ideas. First, a collective solution is sought in the form of a convex combination of the classifiers that produce the "best" results in a certain region of the feature space. The weights in this combination depend on the location of the object to be classified. The closer the object to a region, the greater is the weight of the corresponding algorithm in the convex combination. Actually, this reminds us of the use of the classifier selection procedure with fuzzy boundaries. Second, in addition to efficiency, the best algorithm for each region is generally determined using a stability requirement with respect to small variations of the objects' coordinates. It will be shown in the next section that it is the simultaneous use of these two ideas that improves the generalization ability of the algorithm. We extend the concept of a convex stabilizer using the considerations above.

Divide the control sample into r clusters $D_1, \ldots, D_r$, and define the optimal algorithm for the kth cluster.

Definition 3. The algorithm A_{t_0} is said to be optimal for the kth cluster if it maximizes the objective function

$$t_0 = \arg \max_{t = 1, 2, \ldots, p} [E_k(A_t) + \alpha S_k(A_t)],$$

where $E_k(A_t)$ is the part of the objects in the kth cluster that are correctly classified by the algorithm A_t and $S_k(A_t)$ is the stability component defined by

$$S_k(A) = \frac{1}{q_k} \sum_{y_i \in D_k} \exp\left(-\frac{Z_A(y_i, \varepsilon)}{2\lambda^2}\right).$$

Here, q_k is the number of objects in the control sample that belong to the kth cluster, but λ and α are real numbers such that $\lambda > 0$ and $\alpha \geq 0$.

Note that this definition of the optimal algorithm for the region of the space formed by the corresponding cluster is a compromise between the requirement for the algorithm to produce good results on objects with an a priori known result and the stability requirement, which enables one to hope that the algorithm also works well on arbitrary objects of the universal set. The compromise is determined by the parameters α, ε, and λ. The optimal value of each of these parameters can be either determined in the process of learning. or it is almost independent of the particular problem.

Definition 4. The algorithm A is said to be obtained from the algorithms $A_1, \ldots, A_p$ by using the *convex cluster stabilizer* if it can be represented by the following convex combination of the initial algorithms:

$$P_A(\omega_j \mid x) = \frac{\sum_{k=1}^{r} w_k(x) P_{A_{G(k)}}(\omega_j \mid x)}{\sum_{k=1}^{r} w_k(x)}. \tag{4}$$

Here, $G: \{1, 2, \ldots, r\} \longrightarrow \{1, 2, \ldots, p\}$ is a function that determines the index of the "optimal" classification algorithm for the corresponding cluster and $w_k: \mathbb{R}^n \longrightarrow \mathbb{R}$ are weighting functions possessing the following properties:

$$\begin{gathered} w_k(x) \geq 0 \quad \forall k = 1, 2, \ldots, r, \\ w_k(x) \longrightarrow 0 \text{ as } \rho(x, C_k) \longrightarrow \infty, \\ \frac{w_k(x)}{\sum_{j=1}^{q} w_k(x)} \longrightarrow 1 \text{ as } \rho(x, C_k) \longrightarrow 0. \end{gathered} \tag{5}$$

In (5), $C_k = \frac{1}{q_k} \sum_{y_i \in D_k} y_i$ are the centers of the clusters.

Here is the simplest example of the weighting functions:

$$w_k(x) = \begin{cases} 1 \text{ for } \rho(x, C_k) < \rho(x, C_i) \quad \forall i \neq k, \\ 0 \text{ otherwise}. \end{cases} \tag{6}$$

It is easy to see that this system of weighting functions specifies a crisp decomposition of the space into nonoverlapping regions. Such a decomposition can be useful in solving essentially discrete problems (when the features are nominal or can take a small number of discrete states).

We now show that the scheme described above is a generalization of some well-known collective solution methods.

Theorem 1. *A collective solution obtained by using the convex cluster stabilizer* (4), (6) *with* $\alpha = 0$ *coincides with the solution obtained by the application of the classifier selection procedure* (*see* [10, 11]) *to the initial set of algorithms.*

Proof. Denote by A the result obtained by the application of the convex cluster stabilizer with the parameters indicated in the condition of the theorem to the set of algorithms $A_1, \ldots, A_p$. Due to (6), it is clear that, for each object, this stabilizer produces the same result as that obtained by the application to this object of the optimal algorithm for the given cluster ($A(x) = A_{t_0}(x)$). When $\alpha = 0$, the stability component is not taken into account in finding the optimal algorithm. Therefore, the optimal algorithm for a given cluster is defined as the algorithm that minimizes the number of errors on the objects of the control sample belonging to the cluster. We see that this method of constructing the collective solution coincides with the classifier selection procedure, which is also known as the Clustering & Selection procedure (see [11]). This completes the proof of the theorem.

Theorem 2. *A collective solution obtained by using the convex cluster stabilizer* (4), (5) *with* $r = q$ *and* $0 < \alpha < 1$ *coincides with the application of the convex stabilizer* (2), (3) *with the same system of weighting functions and* $F(k) = P(k)$ *to the initial set of algorithms.*

Proof. It is clear that, if the number of clusters equals the number of objects in the control sample, each cluster consists of a single object, which is the cluster center. Therefore, conditions (3) are equivalent to conditions (5). Hence, we can use the same weighting functions in both cases; i.e., each system of the weighting functions used by the cluster stabilizer can be assigned an equivalent system of functions used by the convex cluster stabilizer and conversely.

It remains to prove that $G(k) = P(k)$. Suppose that there is at least one algorithm in the set that correctly classifies the object $y_k = C_k$. Then, $P(k)$ is equal to the index of the most stable of such algorithms, which has the minimal value of $Z(y_k, \varepsilon)$. Consider the expression $V(A_t) = E_k(A_t) + \alpha S_k(A_t)$. For all the algorithms that classify y_k incorrectly, we have $V(A_t) < 1$, since $E_k(A_t) = 0$ and $S_k(A_t) = \exp\left(-\frac{Z_{A_t}(y_k, \varepsilon)}{2\lambda^2}\right) \in (0, 1]$ and $0 < \alpha < 1$. For any algorithm that correctly classifies y_k, we have $V(A_t) > 1$, since $E_k(A_t) = 1$. Therefore, the optimal algorithm for the kth cluster is sought among the algorithms with the indexes from the set $\Theta(k)$. The optimal algorithm is the one that has the maximum value $S_k(A)$. Since $S_k(A)$ monotonically decreases with respect to $Z_A(y_k, \varepsilon)$ for any λ, its maximum value corresponds to the minimum value of the instability on y_k. Therefore, if $\Theta(k) \neq \emptyset$, we have $G(k) = R(k) = P(k)$.

If none of the given algorithms can correctly classify the object, we can use similar reasoning, taking into account that $E_k(A_t) = 0\ \forall t$, to prove that $G(k) = T(k) = P(k)$.

Theorem 2 shows that the convex stabilizer is a particular case of the convex cluster stabilizer.

4. EXPERIMENTAL RESULTS

To test the utility of the method described above, we conducted a series of experiments. We used some recognition problems from the UCI repository (see [12]), which are often used to compare various recognition paradigms. Actually, this repository provides a benchmark for testing various ideas. The experiments were conducted using the RECOGNITION software package (see [13]), which is a universal tool for the investigation of various problems and data analysis methods. In all the cases, the set consisting of six methods was used: the linear Fisher discriminant, the one-layer perceptron with ten neurons in the layer, the q nearest neighbors method, and the three support vector machines with different parameters (a linear separating hyperplane, a cubic hypersurface, and Gaussian kernel functions). The outputs of all these methods were modified taking into account condition (1). As alternative collective solution methods, we used such effective methods as the naïve Bayes approach [14] and the decision templates method with the Euclidean metrics [15]. The other collective methods implemented in RECOGNITION (committee methods, the Woods method, classifier selection with crisp boundaries, and others) are inferior to the above-mentioned methods as applied to the problems in the UCI repository that were examined in this study.

In all the problems, the control sample was identical to the training sample. The weighting functions were defined by

$$w_k(x) = \frac{1}{\rho^2(x, C_k)} \exp\left(-\frac{\rho^2(x, C_k)}{2\sigma^2}\right). \tag{7}$$

The scaling parameters were set to the typical distance in the sample:

$$\lambda = \sigma = \varepsilon = \max_{i \neq j} \min \rho(y_i, y_j).$$

The number of clusters r and the value α were chosen using an independent control sample or cross validation.

The results of the experiments are presented in the table. Column SB shows the percentage of correctly classified objects for the best algorithm in the set. The following columns show the change of the result upon the application of the naïve Bayes approach (NB), decision templates (DT), and the convex cluster stabilizer (CCS). We also examined the influence of fuzziness of subclusters of the feature space and the stability principle on the generalization ability. Column CA shows the change of the result when the classical classifier selection scheme (i.e., the convex cluster stabilizer with $\alpha = 0$ and the weighting functions defined by (6)) is used. Column CC shows the change of the result when the convex cluster stabilizer with the weighting

Table

Problem	SB	NB	DT	CCS	CA	CC	SA
Melanoma	59.4	3.1	–3.1	6.2	0	3.1	–6.3
Breast	95.5	–1.4	0	–1.1	–1.1	–1.1	–1.1
Credit	86.2	–5.5	–3.7	0.3	–10.9	–1.1	–5.9
Eco	82.7	–2.8	1.1	0.5	–3.9	–0.6	–11.2
Hea	57.4	–5.9	–3	0	–0.1	–0.5	–6.7
Image	86.7	4.4	3.1	4.2	–0.2	1.1	–13.2
Yea	59.7	–1.3	–0.5	–0.1	–5.8	–0.1	–2.8

functions (7) and $\alpha = 0$ is used. Finally, column SA shows the change of the result when the stabilizer with crisp boundaries between the clusters (i.e., the weighting functions (6) and the value of α equal to that used in the CCS method) is used.

The results of the experiments suggest the following conclusions. First, the quality of the classification is significantly improved when a fuzzy decomposition of the feature space is used. Second, taking into account the stability condition also improves the performance. Furthermore, the simultaneous use of a fuzzy decomposition and the stability condition also improves the quality even more. It is worth noting that the convex cluster stabilizer produced good results in all the cases; often the quality was higher than the results produced by the best collective methods. The computational complexity of the convex cluster stabilizer is significantly less than that of the ordinary convex stabilizer. Third, there are problems (e.g., Breast) on which none of the collective methods could improve the result obtained by the best algorithm in the set. Moreover, the use of collective methods even deteriorates the quality. We believe that this is due to the fact that the best algorithm takes into account all the significant regularities in the data, and the use of the results produced by other methods does not add any useful information, while making the model more complicated and prone to random fluctuation. The experiments revealed one more interesting thing. It turned out that the optimal number of clusters is approximately proportional to the square root of the size of the control sample:

$$r \sim \beta\sqrt{q},$$

where $\beta \in (0.5, 3)$. We think that there is an analogy between this fact and the problem of reconstructing probability densities in nonparametric statistics using the dynamic windows method. In this method, it is recommended that the number of objects is proportional to the square root of the size of the sample.

5. CONCLUSIONS

In this paper, we suggested a generalized scheme for constructing collective solutions over the set of algorithms belonging to different families. The main goal in the development was to improve the generalization ability of the resulting algorithm without using additional precedent samples. To this end, we introduce a correction to the criterion function that determines the appropriateness of an algorithm in a certain domain. This correction accounts for the stability of the result when the coordinates of an object vary. For constructing a collective solution, the classifier selection paradigm is used, which was extended for the case of fuzzy boundaries between the clusters. Numerous experiments confirmed that taking into account the stability of a method and the use of fuzzy clusters specified by a system of weighting functions make it possible to improve the quality of collective solutions compared both with the particular algorithms in the set and with other hybrid schemes.

ACKNOWLEDGMENTS

This work was supported in part by the Russian Foundation for Basic Research, project nos. 04-01-00161, 04-01-08045, and 03-01-00580; the Targeted Program of the Division of Mathematics, Russian Academy of Sciences, project no. 2; and grant NSh-1721.2003.1 of the President of the Russian Federation.

REFERENCES

1. J. Kittler, M. Hatef, R. Duin, and J. Matas, "On Combining Classifiers," IEEE Trans. Pattern Anal. Mach. Intell. **20**, 226–239 (1998).

2. A. M. Veshtort, Yu. A. Zuev, and V. V. Krasnoproshin, "A Two-Level Recognition Scheme with a Logical Corrector," Raspoznavanie, Klassifikatsiya, Prognoz: Mat. Metody Ikh Primen., No. 2, 73–98 (1989).
3. Yu. I. Zhuravlev, *Selected Works* (Magistr, Moscow, 1998) [in Russian].
4. R. Shapire, Y. Freund, P. Bartlett, and W. Lee, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," *Proceedings of the 14th International Conference on Machine Learning, 1997* pp. 322–330.
5. V. N. Vapnik and A. Ya. Chervonenkis, *Theory of Pattern Recognition* (Nauka, Moscow, 1974) [in Russian].
6. S. A. Shumskii, "Bayesian Regularization of Learning," in *IV Vserossiiskaya Nauchno-Tekhnicheskaya Konferentsiya Neiroinformatika 2002,* vol. 2, p. 30 [in Russian].
7. D. MacKay, "Bayesian Interpolation," Neural Comput. **4**, 415–447 (1992).
8. D. P. Vetrov, "Design of Correct Pattern Recognition Algorithms with Minimal Instability," Zh. Vychisl. Mat. Mat. Fiz. **43**, 1754–1760 (2003) [Comput. Math. Math. Phys. **43**, 1687–1693 (2003)].
9. F. Esposito, D. Malerba, and G. Semeraro, "A Comparative Analysis of Methods for Pruning Decision Trees," IEEE Trans. Pattern Anal. Mach. Intell. **19**, 476–492 (1997).
10. L. A. Rasstrigin and R. Kh. Erenshtein, *A Method of Collective Recognition* (Energoizdat, Moscow, 1981) [in Russian].
11. A. Lipnickas, "Classifier Fusion with Data Dependent Aggregation Schemes," *Proceedings of the 7th International Conference on Information Networks, Systems, and Technologies, 2001*, vol. 1, p. 147–153.
12. P. Murphy and D. Aha, *UCI Repository of Machine Learning Databases* (Univ. California, Dept. Informat. and Comput. Sci., Irvine, Calif., 1996).
13. Yu. I. Zhuravlev, V. V. Ryazanov, O. V. Sen'ko, *et al.*, "Development of a Universal Program for the Intelligent Data Analysis, Recognition, and Forecasting," in *Doklady XI Vserossiiskoi konferentsii Matematicheskie Metody Raspoznavaniya Obrazov, 2003*, p. 311 [in Russian].
14. L. Xu, A. Krzyzak, and C. Suen, "Methods of Combining Multiple Classifiers and Their Application to Handwriting Recognition," IEEE Trans. Syst., Man, Cybernetics **22**, 418–435 (1992).
15. L. Kuncheva, J. Bezdek, and R. Duin, "Decision Templates for Multiple Classifier Fusion: An Experimental Comparison," Pattern Recogn. Image Anal. **34**, 299–314 (2001).