On the Stability of Pattern Recognition Algorithms[1]

D. P. Vetrov

Dorodnitsyn Computer Center, Russian Academy of Sciences, ul. Vavilova 40, GSP-1, Moscow, 119991 Russia
e-mail: vetrovd@yandex.ru

Abstract—In this work, an attempt is made to introduce a measure of stability for a wide class of pattern recognition algorithms. Based on this notion, constructive methods for the synthesis of correct (or close to correct) stable algorithms are built. Such algorithms can be derived either from a set of parametric algorithms of the same model or from the algorithms which belong to different models.

1. INTRODUCTION

In pattern recognition theory, a quality functional which depends on the number of errors of the algorithm in a test sample is regarded as a commonly accepted efficiency factor of the recognition algorithm. One of the formulations of the pattern recognition problem involves the construction of an algorithm that does not commit errors on the test sample [1]. Such algorithms are known as *correct algorithms*. When solving practical problems, one assumes that the correct algorithm exhibits error-free operation upon classification of originally unclassified objects. However, this assumption is not always valid. If the sizes of the learning and test samples are small and the number of the parameters optimized is large, algorithm overfitting takes place. This means that the algorithm correctly recognizes the objects from the test sample and degenerates at the remaining objects (yields incorrect results, refuses recognition, etc.). Conventional means of eliminating such a phenomenon are deliberate limitation of the parametric family of the algorithms [2] and simultaneous use of several algorithms [3, 4]. In this work, we propose an approach that makes it possible to simultaneously increase the efficiency of the algorithm on the test sample and its stability against variations in the attributes of the object.

Below, we analyze the algorithms represented as $A = B \circ C$, where B is the *recognition operator* that calculates the *classification estimate* and C is the *decision rule* making the estimate-based classification [1]. Let us consider a standard problem of the recognition of objects represented by vectors in an n-dimensional real space with l classes. In this case, the recognition operator is given by

$$B = \{\Gamma_B^k(S)\}, \quad k = 1 \ldots l, \quad S \in R^n,$$

$$\text{where } \Gamma_B^k(S)\colon R^n \longrightarrow [0, 1],$$

where $\sum_{k=1}^{l} \Gamma_B^k(S) = 1$.

Apparently, any algorithm calculating the attributes of the classification estimate can be represented in such a way owing to the normalization of the estimates obtained. This is valid not only for conventional models of estimate calculations but also for neural networks, algorithms using potential functions, and for various types of fuzzy classifiers. Further, without loss of generality, we assume that appropriately scaled attributes do not correlate with each other. We define the objects of the test sample by S_j, $j = 1, \ldots, q$.

2. INSTABILITY OF RECOGNITION OPERATORS

Normally, the fitting of the parameters of the recognition algorithm employs a test sample. It is expedient to perform the correction of algorithms in the space of the classification estimates rather than in the space of the decision, because the family of the correction operators in the latter case is poor [1]. Similar reasoning is valid for the stabilizing operations. Therefore, the main objects for the further analysis are the recognition operators. We assume that a recognition operator consists of l functions of n variables. In this case, it is expedient to use an analog of the difference gradient to characterize the stability of the operator in the point.

Definition 1. The instability of recognition operator B on the jth object of the test sample is defined by

$$Z_B(S_j, \varepsilon) = \sum_{k=1}^{l} \frac{1}{\varepsilon_k^2} \sum_{i=1}^{n} [\Gamma_B^k(S_j + \varepsilon_k e_i) - \Gamma_B^k(S_j)]^2,$$

where e_i is a unit vector of the corresponding coordinate and $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_l)$.

[1] This work was supported by the Russian Foundation for Basic Research, project nos. 02-07-90134, 02-07-90137, and 02-01-08007.

Received February 28, 2003

Remark. The instability of the recognition operator depends on the shift ε_k. This parameter can be interpreted as a mean distance from recognized objects of the kth class to the nearest object of the test sample of their own class. This parameter can be chosen based on either *a priori* assumptions or the information contained in the test sample, for example, in the following way:

$$\varepsilon_k = \frac{1}{2}\min[\max_{S_j \in K_k} \min_{S_i \in K_k} \rho(S_i, S_j), \min_{m \neq k} \rho(K_i, K_m)],$$

where $\rho(S_i, S_j)$ is the distance between the corresponding objects and $\rho(K_k, K_m)$ is the distance between the classes calculated by using the objects of the test sample.

Definition 2. The instability of recognition operator B (on the test sample) is defined as

$$Z_B(\varepsilon) = \sum_{j=1}^{q} Z_B(S_j, \varepsilon).$$

Definition 3. The recognition operator B_1 (and the corresponding recognition algorithm A_1) are more stable than the recognition operator B_2 (and the corresponding algorithm A_2) if

$$Z_{B_1}(\varepsilon) < Z_{B_2}(\varepsilon).$$

Further, we assume that ε is constant and omit the corresponding symbol.

It is easy to construct an absolutely stable recognition operator. For example, an operator all whose functions are constants in the entire space of objects is absolutely stable. It is clear from this example that the instability of the recognition operator considered separately from its efficiency (i.e., the quality of operation on the test sample) cannot be a single criterion for choosing an algorithm. Therefore, one can use the instability of the recognition operator (and, hence, of the recognition algorithm) to compare two correct algorithms. In this case, preference should be given to a more stable algorithm. The further purpose is to construct a new operator which is more stable and, at least, no less effective using the given set of recognition operators. We call this process *stabilization* of the recognition algorithms. Let us demonstrate that, in this case, one can preserve the correctness of the algorithms.

3. SYNTHESIS OF A CORRECT STABLE RECOGNITION OPERATOR FROM A FAMILY OF CORRECT OPERATORS

The formulation of the problem is as follows. We have p recognition operators $B_1, \ldots, B_p$. The task is to construct a recognition operator B such that $Z_B \leq \min_{t=1\ldots p} Z_{B_t}$. Let $T(j) = \arg\min_{t=1\ldots p} Z_{B_t}(S_j)$ be the index of the most stable (on the jth object) recognition operator.

Definition 4. The recognition operator B is constructed from $B_1, \ldots, B_p$ using a *convex stabilizer* if it can be represented by a convex combination of the original recognition operators:

$$\Gamma_B^k(S) = \frac{\sum_{j=1}^{q} w_j(S)\Gamma^k_{B(F(j))}(S)}{\sum_{j=1}^{q} w_j(S)}, \quad k = 1\ldots l, \quad S \in R^n, \tag{3.1}$$

where $F: \{1\ldots q\} \longrightarrow \{1\ldots p\}$ is a function that determines the index of the recognition operator for each object of the test sample and $w_j: R^n \longrightarrow R$ are weighting functions featuring the following properties:

$$w_j(S) \longrightarrow 0 \text{ at } \rho(S, S_j) \longrightarrow \infty,$$
$$\frac{w_j(S)}{\sum_{j=1}^{q} w_j(S)} \longrightarrow 1 \text{ at } \rho(S, S_j) \longrightarrow 0.$$

Definition 5. Recognition operator B is *continuous* if all the corresponding functions $\{\Gamma_B^k(S)\}_{k=1}^{l}\}$ are continuous in R^n.

Remark. Assuming that $w_j(S)$ equals unity if $\forall k$ $\rho(S, S_j) < \rho(S, S_k)$ (otherwise, it equals zero), we obtain a convex stabilizer working by the nearest neighbor rule, so that for the object recognized we use the recognition operator which is best, in a sense, for the nearest object of the test sample. In this case, the space of the objects is split into nonintersecting preferable areas of the corresponding algorithms. Such a scenario resembles separating a space into competence areas [5]. It is obvious that, generally, the resulting recognition operator is not continuous. The continuity can be violated at points equidistant with respect to a few objects of the test sample. Such an approach is acceptable if the original recognition operators are represented by sets of functions that take no more than a denumerable number of values.

Below, we assume that

$$\varepsilon_k = \varepsilon < 0.5 \min_{T(j) \neq T(i)} \rho(S_i, S_j). \tag{3.2}$$

Consider the convex stabilizer obtained by substituting the following values into Eq. (3.1):

$$F(j) \equiv T(j), \tag{3.3}$$

$$w_j(S) = \begin{cases} 1 & \text{if } \rho(S, S_j) \leq \varepsilon \\ 0 & \text{if } \exists S_i \neq S_j : \rho(S, S_i) \leq \varepsilon \\ \dfrac{1}{\rho(S, S_j) - \varepsilon} & \text{if } \forall S_i\ \rho(S, S_i) > \varepsilon. \end{cases} \tag{3.4}$$

Lemma 1. Function $w_i(S)/\sum_{j=1}^{q} w_j(S)$ is continuous in R^n.

Proof. To prove this statement, it suffices to analyze the behavior of the function at the points lying at the distance ε from a certain object of the test sample S_i. In accordance with Eq. (3.2), if such an object exists, it is unique. By making a forward substitution in expression (3.4), we arrive at

$$\lim_{\rho(S_i, S) \to \varepsilon} \frac{w_r(S)}{\sum_{j=1}^{q} w_j(S)} = \delta_{ir}.$$

The continuity of the function at the remaining points is obvious due to the continuity of the corresponding functions in expression (3.4). Therefore, the assumed function is continuous over the entire space R^n and the lemma is proven.

Theorem 1. The instability (on the test sample) of recognition operator B constructed by applying a convex stabilizer given by expressions (3.3) and (3.4) to the original family of recognition operators does not exceed the instability of the most stable of these operators.

Proof. Let us consider the instability of B on an arbitrary object S_i of the test sample. In accordance with expressions (3.3) and (3.4), in the ε neighborhood of this object, the action of the recognition operator B is identical to the action of the operator $B_{T(i)}$, which is the most stable (among the operators of the original family) operator on the ith object. The instability of the operator on the given object is completely determined by its behavior in the corresponding ε neighborhood. Therefore, the instability of the operator B on the ith object of the test sample is exactly equal to the instability of the operator $B_{T(i)}$. Thus, the instability of B on the entire test sample is

$$Z_B = \sum_{j=1}^{q} Z_{B(T(j))}(S_j) = \sum_{j=1}^{q} \min_{t=1\ldots p} Z_{B_t}(S_j)$$

$$\leq \min_{t=1\ldots p} \sum_{j=1}^{q} Z_{B_t}(S_j) = \min_{t=1\ldots p} Z_{B_t}$$

and the theorem is proven.

Statement 1. The following equality is satisfied for any recognition operator B obtained by applying any convex stabilizer:

$$\Gamma_B^k(S_j) = \Gamma_{B(F(j))}^k(S_j) \quad k = 1\ldots l \quad j = 1\ldots q.$$

Proof. In accordance with expression (3.1) and the conditions of weighting functions, the action of operator B at point S_j is identical to the action of operator $B_{F(j)}$. This means that the classification estimates coincide and the statement is proven.

Theorem 2. If recognition operators $B_1\ldots B_p$ are correct, the recognition operator B constructed by applying the convex stabilizer given by expressions (3.3) and (3.4) is also correct.

Proof. It follows from the previous statement that, at any point of the test sample, the action of the operator B is identical to that of one of the operators of the original family. Each of these operators correctly recognizes any object of the test sample. Therefore, the operator B also correctly recognizes all the objects of the test sample. The theorem is proven.

Remark. Let all the recognition operators of the original family possess continuous estimating functions in the object space. Then, in accordance with Lemma 1, the resulting recognition operator also exhibits this property.

Thus, by applying the aforementioned convex stabilizer to the family of correct recognition operators, one can obtain a more stable operator without violating the correctness.

4. SYNTHESIS OF AN EFFECTIVE STABLE RECOGNITION OPERATOR FROM A FAMILY OF INCORRECT OPERATORS

In practical applications, we have to deal with families of incorrect algorithms. Such families often belong to the same parametric model of algorithms. These can be algorithms at which the local extrema of the quality functional are reached. The question arises if it is possible to construct correct algorithms on this family.

Definition 6. The object of a test sample is called *regular* with respect to the family of the recognition operators $B_1\ldots B_p$ if at least one of these operators correctly recognizes the given object. In the opposite case, the object is irregular.

Further, we assume that the family of algorithms is sufficiently rich and the test sample does not contain irregular objects. Let us introduce the following quantities:

$$\Theta(j) = \{t | B_t \text{ correctly classifies } S_j\},$$

$$R(j) = \arg\min_{t \in \Theta(j)} Z_{B_t}(S_j).$$

We consider a convex stabilizer obtained by substituting expression (3.4) and

$$F(j) \equiv R(j) \tag{4.1}$$

into expression (3.1).

Theorem 3. The recognition operator B constructed by applying the convex stabilizer given by expressions (3.4) and (4.1) to the family of operators $B_1\ldots B_p$ is correct, and its instability on each object of the test sample is no greater than the instability of the most stable operator that correctly recognizes the corresponding object.

Proof. The correctness of the constructed recognition operator follows from the representation of the set $\Theta(j)$ and Statement 1. Let us consider its stability. It follows from expressions (3.2) and (3.4) that, in a certain ε neighborhood of each object S_j, the operator B is identical to a certain operator $B_{R(j)}$. Such a neighborhood completely determines the instability at the jth object. Thus,

$$\mathrm{Z}_B(S_j) = \mathrm{Z}_{B(R(j))}(S_j) \le \mathrm{Z}_{B_t}(S_j),$$
$$\forall j = 1 \ldots q \ \forall t \in \Theta(j),$$

and the theorem is proven.

The last theorem offers a constructive way to build correct stable algorithms in the absence of irregular objects in the test sample. This procedure makes it possible to construct an algorithm complying with two requirements (correctness and stability). In the presence of irregular objects, we introduce the notation

$$\mathrm{P}(j) = \begin{cases} \mathrm{T}(j) \text{ if } \Theta(j) = \varnothing \\ \mathrm{R}(j) \text{ otherwise.} \end{cases} \quad (4.2)$$

Let us consider a convex stabilizer in which the weighting functions are given by expression (3.4) and F(j) is represented by P(j). The efficiency of the recognition operator is interpreted as the number of errors on the test sample. Then, the following theorem is valid.

Theorem 4. The recognition operator B constructed by applying the convex stabilizer given by expressions (3.4) and (4.2) to the family of operators $B_1 \ldots B_p$ is no less effective than the most effective operator of this family, and its instability on each object of the test sample is no greater than the instability of the most stable (at this object) operator which correctly recognizes the corresponding object provided it is a regular object of the sample, and is no greater than the instability of the most stable (at this object) algorithm of the original family in the case of an irregular object.

Proof. It follows from Statement 1 and formula (4.2) that the constructed recognition operator correctly classifies all regular (with respect to the given family) objects of the test sample. Therefore, the errors are made only on irregular objects. However, all the algorithms of the original family commit errors on these objects. The first statement is proven.

For the irregular (regular) object, the second statement follows from Theorem 1 (3). Thus, the theorem is proven.

It follows from the last two theorems that one can simultaneously increase the efficiency and stability of modern recognition algorithms applied to a specific problem.

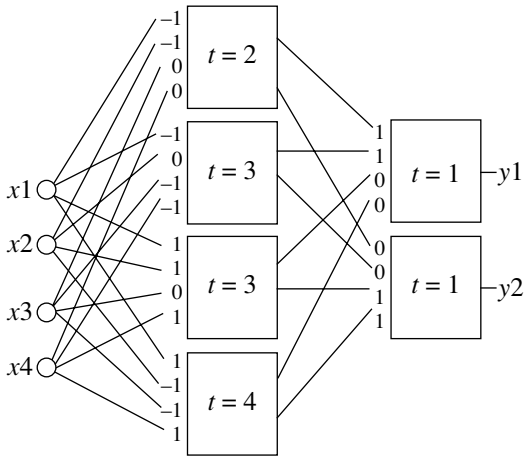

Fig. 1. Structure and weighting coefficients of the NN_1 network.

5. EXAMPLE OF SYNTHESIS OF A STABLE RECOGNITION OPERATOR BY APPLYING A CONVEX STABILIZER TO NEURAL NETWORKS

Let us consider a simple example of application of the above theory. Suppose that it is necessary to classify objects with features taking the values 1 and –1 into two nonintersecting classes. The dimension of feature space equals four. Let the learning sample contain four objects of the first class $S^1 = (-1, -1, -1, -1)$, $S^2 = (-1, 1, -1, -1)$, $S^3 = (-1, -1, 1, -1)$, and $S^4 = (-1, -1, -1, 1)$ and three objects of the second class $S^5 = (1, -1, -1, 1)$, $S^6 = (1, 1, -1, 1)$, and $S^7 = (1, 1, 1, 1)$. For simplicity, we assume that the test sample coincides with the learning one. Suppose that we have two neural networks NN_1 and NN_2 which correctly classify the original objects and whose activation functions are given by

$$\text{output} = \begin{cases} +1 \text{ if } \langle \mathrm{w}, \text{input} \rangle \ge t \\ -1 \text{ if } \langle \mathrm{w}, \text{input} \rangle < t. \end{cases}$$

Figures 1 and 2 show the structure of these functions and their weighting coefficients (obtained, for example, by random search).

The decision rule is written as

$$C(B(S)) = \begin{cases} 1 \text{ if } B(S) = (1, 0) \\ 2 \text{ if } B(S) = (0, 1) \\ ? \text{ if } B(S) = (1, 1) \text{ or } B(S) = (0, 0). \end{cases}$$

Figure 3 shows the work of both networks at the remaining points. The objects of the test sample are in bold face. For the remaining objects, the variants of their classification by the first and the second networks are indicated. Specifically, they yield opposite results in point $S = (-1, -1, 1, 1)$. A simple calculation of the instability of the neural networks at $\varepsilon = 1$ yields the fol-

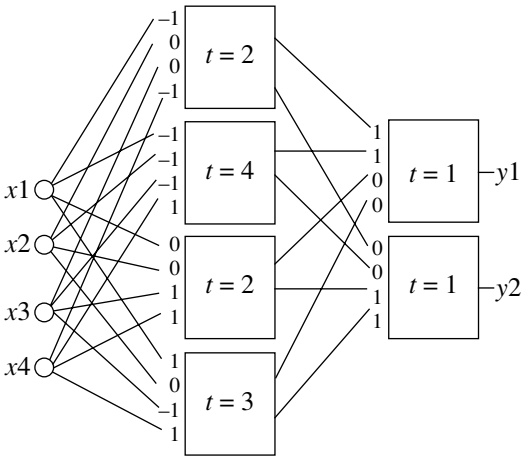

Fig. 2. Structure and weighting coefficients of the NN_2 network.

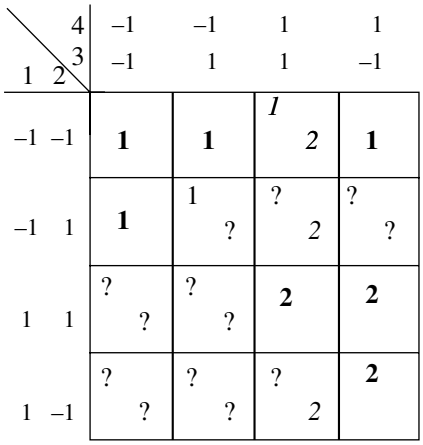

Fig. 3. Results of classification using original neural networks.

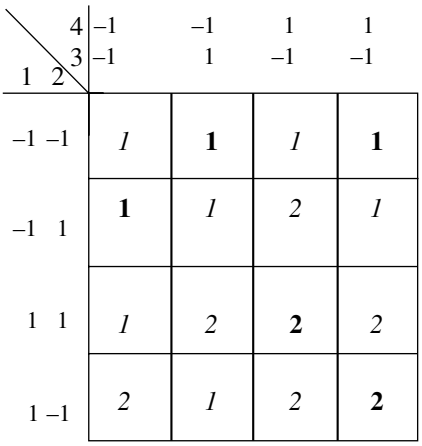

Fig. 4. Stability regions of the original neural networks.

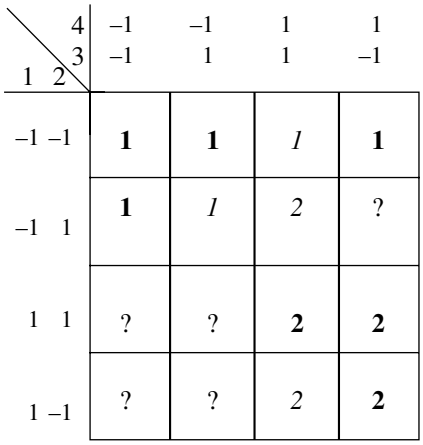

Fig. 5. Final variant of classification of the objects.

lowing values: $Z_1 = 17$ and $Z_2 = 19$. Based on the stability criterion, preference must be given to the first neural network.

One can improve the obtained results by applying the following convex stabilizer to these neural networks:

$$\mathrm{F}(j) \equiv \mathrm{T}(j),$$

$$\mathrm{w}_j(S)$$

$$= \begin{cases} 1/p \ \text{ if } \ \rho(S, S_j) \le \rho(S, S_k)\ \forall k = \overline{1, q} \\ \quad \exists j_1, \ldots, j_p\colon \rho(S, S_j) = \rho(S, S_{j_1}) = \ldots = \rho(S, S_{j_p}) \\ 1 \ \text{ if } \ \rho(S, S_j) < \rho(S, S_k)\ \forall k = \overline{1, q} \\ 0 \quad \text{otherwise.} \end{cases}$$

Note that while building convex stabilizer we do not consider the objects of the test sample on which both networks exhibit equal values of instability. To calculate the distances, we use the following metric:

$$\rho(x, y) = \left|\{i \,|\, x_i \ne y_i\}\right|.$$

Figure 4 shows how to choose the neural network for classification of the given object. The objects of the test sample where one of the networks is more stable than another are boldfaced.

The constructed classifier recognizes a greater number of objects and appears to be more stable in comparison to each of the original networks. Note that object $S = (-1, -1, 1, 1)$ is classified as belonging to the first class, which is in agreement with its intuitive classification. Figure 5 demonstrates the results of the work of the classifier. In bold face, we show the objects of the test sample.

6. CONCLUSIONS

In this work, we propose one of the possible methods for solving the problem of instability of modern effective (in particular, correct) algorithms for pattern recognition. After formalizing the notion of instability of the recognition algorithm, one can employ stabilizing operations aimed at increasing the stability of the algorithm. All operations are applied to recognition operators that calculate the classification estimates, which makes it possible to construct rather rich families of stabilizing operations. We analyze in detail the construction of a convex stabilizer enabling one to increase the stability of the algorithm without changing its local peculiarities. The methods obtained allow construction of more stable but still effective algorithms.

REFERENCES

1. Zhuravlev, Yu.I., *Selected Scientific Works*, Moscow: Magistr, 1998.
2. Vapnik, V., *Statistical Learning Theory*, New York: Wiley, 1998.
3. Kittler, J., Hatef, M., Duin, R.W.P., and Matas, J., On Combining Classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998, vol. 20, no. 3, pp. 226–239.
4. Kuncheva, L.I., Bezdek, J.S., and Duin, R.P.W., Decision Templates for Multiple Classifier Fusion: and Experimental Comparison, *Pattern Recognition*, 2001, vol. 34, no. 2, pp. 299–314.
5. Rastrigin, L.A. and Erenshtein R.Kh., *Method of Collective Recognition*, Moscow: Energoizdat, 1981.

Dmitrii P. Vetrov. Born 1981. Graduated from Moscow State University in 2003. Engineer at the Dorodnitsyn Computer Center, Russian Academy of Sciences. Area of research: pattern recognition, mathematical statistics, expert systems, and game theory. Author of three articles.