Data Dependent Classifier Fusion for Construction of Stable Effective Algorithms.

Vetrov Dmitry
VetrovD@yandex.ru

Kropotov Dmitry
DKropotov@yandex.ru

Dorodnicyn Computing Centre of the Russian Academy of Sciences
Russia, 119991, Moscow
Vavilova str. 40

Abstract

A measure of stability for a wide class of pattern recognition algorithms is introduced to cope with overfitting in classification problems. Based on this concept, constructive methods for designing effective stable algorithms are developed. New algorithm is represented as convex combination of the initial algorithms with weights that depend both from the location of the point being classified and from the degree of local stability of each algorithm. Either a set of parametric algorithms from the same model or algorithms that belong to different models may be used for such fusion.

1. Introduction

Classifier fusion is considered to be a promising way in improving the performance of pattern recognition algorithms. Many authors proposed different ways of fusing classifiers ([5], [8], [10], [6]). Conceptually, we may separate out two main paradigms: data independent and data dependent classifier fusion. The first means that a measure of influence of each algorithm is constant among the whole objects space. These methods are easier to build and generally represent some kind of algorithms voting [5]. They improve classification if the following assumption is right: At any (or at least at most) point from the objects space, the majority of algorithms work "well" in some sense. Then their union may become better than the best of initial algorithms. Otherwise data-independent fusion can hardly improve performance. Data-dependent classifier fusion means that the coefficients which characterize the influence of each classifier depend on the location of the given point. The most simple case is to use "winner-takes-it-all" weighting functions. Such approach is called dynamic classifier selection. In this case only one coefficient has non-zero value i.e. we select only one classifier of the initial algorithms according to some rule and use it for further classification ([8], [10], [2]). There are also more sophisticated concepts with continuous weighting functions, but they require special procedures of training which are not trivial in general case [4]. Recently Jin and his colleagues proposed data-dependent boosting concept for improving robustness of classical AdaBoost algorithm [3]. But the ideas contained in their work are unlikely to be implemented directly to fusion of various classifiers which belong to different families.

In this paper we present an approach, which is based on simultaneous increase of classifiers efficiency (the rate of correctly classified objects from the validation set) and stability. The last requirement should improve generalization capability of the algorithm. The methods of statistical learning theory allow to estimate the tendency to overfit (e.g. by means of VC-dimension [9]) for the given algorithm. Nevertheless we still can't say how much the algorithm is overtuned to the particular task when the training is finished. That's why some indirect values which could characterize the generalization capability are needed. A possible solution is to measure how much the algorithm's output changes if the inputs vary slightly. It seems obvious that large fluctuations of outputs in the regions where the concentration of objects is high, lead to poor generalization. We suggest to consider this information together with algorithms competence when fusing them. During classification of arbitrary object accurate and stable in its neighborhood algorithms should have greater weights. In the next section we define a key notion of the paper. Section 3 describes the way of classifier fusion with the aid of convex stabilizer and section 4 shows some experimental results.

2. Instability of recognition algorithms

In what follows we consider algorithms, which return the vector of estimates. Each of its components is the estimation of posterior probability that a concerned object belongs to the given class.

$$A : \mathbb{R}^n \to P^l \tag{1}$$

where $P^l = \{(p_1, \ldots, p_l) | \sum_{k=1}^{l} p_k = 1, p_k > 0)\}$. Here n is number of features and l is number of classes. So we may consider each recognition algorithm as a set of l functions of n variables

$$A = \{P^k(S)\}_{k=1}^{l}, S \in \mathbb{R}^n$$

If the features posses real values, it seems natural to demand that all $P^k(S) = p_k$ should be continuous. Note that a large number of algorithms can be represented in form (1). All such classifiers may be used for further fusion.

After training is finished, we would like to use those algorithms which are less overtuned. Since the direct measuring of actual overfitting degree is unavailable, some indirect values should be involved. The simplest example of such value is gradient of posterior probabilities in the given point. Similar ideas were used in double-backpropagation conformably to neural networks [1].

Definition 1. The instability of recognition algorithm on the i^{th} object of the sample S_i is expressed by the following formula:

$$Z_A(S_i) = \|\nabla \vec{P}(S_i)\|^2 \tag{2}$$

where $\vec{P}(S) = (P_1(S), \ldots, P_l(S))$.

In cases when the answers of algorithm are not expressed in functional form directly or the inputs are discrete and hence the derivative can't be evaluated analytically, its difference analogue could be used.

$$\hat{Z}_A(S_i) = \sum_{k=1}^{l} \sum_{j=1}^{n} \left(\frac{P^k(S_i + \epsilon_j) - P^k(S_i)}{\epsilon_j} \right)^2$$

If we want to select just one algorithm among the initial set of classifiers, which showed acceptable performance on the training sample, then we should be able to estimate the degree of instability in the entire objects space.

Definition 2. The instability of recognition algorithm (according to μ-measure) is expressed in the following way:

$$Z_A = \int_S Z_A(s) d\mu(s) \tag{3}$$

As there is only finite number of available objects, the integral in (3) turns to summation among the objects from validation set $X = \{S_i\}_{i=1}^{q}$. Then measure μ can be defined like

$$\mu(B) = |\{S_i \in B\}|$$

Generally, validation set should differ from the training sample because many algorithms (e.g. Q-nearest neighbors) do not perform on the training sample the same way as they do on arbitrary objects.

Definition 3. Classifier A_1 is more stable than classifier A_2 if

$$Z_{A_1} > Z_{A_2}$$

An absolutely stable classifier is easy to construct. For instance, this is the one whose posterior estimates are constant in the entire object space. This example clearly shows that the instability of recognition algorithms considered regardless of its effectiveness (i.e. of its performance on validation set) cannot serve as a single measure of the preference of the corresponding classifier. Therefore, the instability value can be used, for example, for comparing two algorithms, which have acceptable error level on the validation set. In this case, a more stable classifier is preferable. Our further goal is, given a set of algorithms, to construct a new one that is more stable and at least as effective as the best of initial ones. This process is called stabilization of recognition algorithms.

3. Convex stabilizer

Consider the following problem. Given p classifiers $A_1, \ldots, A_p$ construct a new algorithm A such that $Z_A \leq \min_{t=1,\ldots,p} Z_{A_t}$. The initial classifiers may belong either to the same parametrical model (e.g. neural networks on which the quality functional has local extremum) or have different concept and structure. The only demand is they operate in the similar terms. In our case this means that they should return posterior probabilities. Let $T(j) = \arg\min_{t=1,\ldots,p} Z_{A_t}(S_j)$ be the index of the most stable recognition algorithm on j^{th} object.

Definition 4. An algorithm A is said to be derived from $A_1, \ldots, A_p$ by applying a convex stabilizer if A can be represented as a convex combination of the original classifiers:

$$P_A^k(S) = \frac{\sum_{j=1}^{q} w_j(S) P_{A_{F(j)}}^k(S)}{\sum_{j=1}^{q} w_j(S)} \tag{4}$$

Where q is number of objects in the validation set, $F : \{1, \ldots, q\} \to \{1, \ldots, p\}$ is a function defining the

index of the best in some sense classifier for each object in the validation set and $w_j : R^n \to R$ are weight functions having the following properties:

$$w_j(S) \to 0 \quad if\ \rho(S, S_j) \to \infty$$

$$\frac{w_j(S)}{\sum_{i=1}^{q} w_i(S)} \to 1 \quad if\ \rho(S, S_j) \to 0$$

Definition 5. A recognition algorithm is called continuous if all corresponding functions $P^k(S)$ are continuous in $\mathbb{R}^n$.

Note, that setting $w_j(S)$ equal to unity if $\rho(S, S_j) < \rho(S, S_k)$ for any k and equal to zero otherwise, we obtain a convex stabilizer that performs according to the nearest-neighbor rule; i.e. the algorithm that is the best for the nearest object of the validation set is applied to the object being recognized. It is easy to see that the resulting classifier is not generally continuous. For example, it may be discontinuous at points equidistant from several objects of the validation set. This approach can be used when the features take no more than a countable number of values (discrete features). Consider another convex stabilizer obtained by substituting the following values into (4)

$$F(j) = T(j) \tag{5}$$

$$w_j(S) = \frac{1}{\rho^2(S, S_j)} exp\left(-\frac{\rho^2(S, S_j)}{2\sigma^2}\right) \tag{6}$$

with

$$\sigma = \max_j \min_i \rho(S_i, S_j)$$

It is easy to see that applying convex stabilizer 5, 6 to continuous algorithms we receive also a continuous algorithm.

Theorem 1. Suppose that algorithm A is obtained from the original family of classifiers by applying a convex stabilizer (5), (6). Then its instability does not exceed the instability of the most stable classifier of this family.

Now consider validation set a little bit closer.

Definition 6. An object in the validation set is regular with respect to set of classifiers $A_1, \ldots, A_p$ if there exists at least one classifier which recognizes the given object correctly. Otherwise the object is irregular.

To continue we introduce some notation. Denote

$$\Theta(j) = \{t | A_t\ recognizes\ S_j\ correctly\}$$

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

Consider a convex stabilizer in which the weight functions are defined by (6) and

$$F(j) = \begin{cases} R(j), & \Theta(j) \neq \emptyset \\ T(j), & otherwise \end{cases} \tag{7}$$

By effectiveness of algorithm, we mean the fraction of errors on the validation set. Then the following theorem is true

Theorem 2. The algorithm A constructed by applying convex stabilizer (6), (7) to a set of classifiers $A_1, \ldots, A_p$ is no less effective than the most effective classifier in this set. Moreover, its instability on each object of the validation set does not exceed the instability of the most stable (on this object) classifier correctly recognizing the given object if the latter is regular, and does not exceed the instability of the most stable (on this object) classifier from the entire initial set if the object is irregular.

This theorem means that by applying convex stabilizer one may get both effective and stable (hence less overtrained) classifier.

4. Experimental results

The concepts described above were successfully implemented as a part of project for creating universal program complex of recognition and data-mining "RECOGNITION (LOREG)" [12]. This system made possible to undertake comparative analysis of classifier fusion methods. As initial families of algorithms, there were used about 10 models including multilayer perceptrons, support vector machines, some logical, combinatorial and statistical methods. For some methods (e.g. linear Fisher discriminant) the necessary changes were made in order them to return the vector of posterior probabilities. The algorithms parameters were selected according to the best performance on the validation set. Convex stabilizer (CS) was compared with several other fusion methods like committee methods (such as taking maximum (MAX), minimum (MIN), product (PRO), mean (AVR) and simple mode majority (MAJ) of posterior probabilities) [5], dynamic Woods (WOO) method [10], Naive Bayes (NB) approach [11], decision templates (DT) with Euclidian distance in the space of profiles [7], clustering and selection (CLU) method [6]. The comparison was held according to three applications, which represent typical kinds of recognition tasks. The first was pneumonia diagnostics (7 classes, 8 features, 158 objects in the training sample), second was melanoma diagnostics (3 classes, 33 features, 48 objects in the training sample) and the last was latin letters optical recognition (26 classes, 16 features, 2200 objects in the training sample). The validation set had approximately the same size and on melanoma task it coincided with the training sample due to the lack of objects. The results of experiments (percent of correctly recognized objects from the independent test sample) are shown in the figure 1.

The last line shows the best classifier (SB) among the

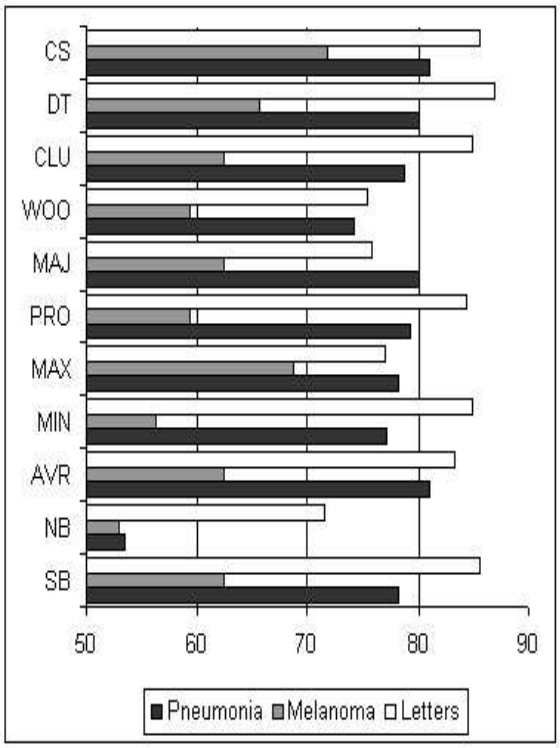

Figure 1. The performance of different aggregation schemes on three real-life tasks.

initial ones. These (and also some other experiments) made clear that a convex stabilizer gives the result at least not worse than best fusion methods in many cases. On the tasks with small samples it tends to outperform other methods significantly improving the recognition. Another good property of convex stabilizer is that despite of other fusion methods, as a rule it doesn't make the performance worse than single best classifier. The main disadvantage is high computational complexity and hence relatively long time of learning and recognition. Choosing another norm in (2) and implementing some heuristics help to reduce the computational time.

5. Conclusion

Stabilization of recognition algorithms is a possible way of building more effective and less overtrained classifiers. The main idea is to perform fuzzy splitting of the objects space into several areas and assign bigger weight to those classifiers that are more stable (have smaller gradient of posterior estimates) and still recognize the objects correctly from some neighborhood. Classifiers are fused by means of taking convex combination of their outputs, with coefficients depending on the point in objects space. The corresponding construction called convex stabilizer was introduced. Some theoretical results found for such concept were briefly discussed. The realization of convex stabilizer proved that this method of classifier fusion can be used for improving the recognition performance especially on the small samples.

The work was partially supported by the Russian Foundation for Basic Research (grants 02-07-90134, 02-01-00558, 02-01-08007, 02-07-90137, 03-01-00580).

References

[1] H. Drucker and Y. LeCun. Improving generalization performance using double backpropagation. *IEEE Transaction on Neural Networks*, 3(6):991–997, 1992.

[2] G. Giacinto and F. Roli. Adaptive selection of image classifiers. *Proceedings of 9th International Conference on Image Analysis and Processing*, pages 38–45, 1997.

[3] R. Jin, Y. Liu, L.Si, J. Carbonell, and A.G.Hauptmann. A new boosting algorithm using input-dependent regularizer. *Proceedings of the 20th International Conference on Machine Learning*, 2003.

[4] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994.

[5] J. Kittler, M. Fatef, R. Duin, and J.Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[6] A. Kopustinkas and A. Lipnickas. Classifiers fusion with data dependent aggregation schemes. *Proceedings of the 7th International Conference on Information Networks, Systems and Technologies*, 1:147–153, 2001.

[7] L. Kuncheva, J. Bezdek, and R. Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34(2):299–314, 2001.

[8] L. Rasstrigin and R. Erenstein. *Method of collective recognition.* Energoizdat, Moscow, 1981.

[9] V. Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.

[10] K. Woods, W. Keelmeyer, and K. Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:405–410, 1997.

[11] L. Xu, A. Krzyzak, and C. Suen. Methods of combining multiple classifiers and their application to handwriting recognition. *IEEE Transactions on Systems, Man, Cybernetics*, 22:418–435, 1992.

[12] Y. Zhuravlev. et al. Development of universal program system of intellectual data analysis, recognition and forecast. *Proceedings of 11th All-Russian Conference on Mathematical Methods of Pattern Recognition*, pages 311–314, 2003.