H-009 A Semi-Supervised MarginBoost Algorithm Applicable for Dissimilarity-Based Classifications

Sang-Woon Kim and Thanh Binh Le

Dept. of Computer Engineering, Myongji University,
Yongin, Gyeonggi, 449-728 South Korea
{*kimsw, binhle*}*@mju.ac.kr*

1 Introduction

Dissimilarity-based classifications (DBCs) are a way of defining classifiers among the classes. The process is not based on the feature measurements of individual patterns, but rather on a suitable dissimilarity measure among the individual patterns [1], [2]. One of the major questions we encountered when designing DBCs is how to design classifiers in the dissimilarity space. On this subject, the use of many traditional decision classifiers, including the k-NN rule and the linear/quadratic normal-density-based classifiers has been reported [1], [2]. On the other hand, boosting has been developed as a very successful learning technique for solving the problem of classifying object samples in the input feature space. Furthermore, semi-supervised learning algorithms, such as semi-supervised MarginBoost (SSMB) [4], have been proposed for exploiting the unlabeled data in addition to the labeled data to improve performance on the classification task; performance improvement of any supervised learning algorithm with a multitude of unlabeled data [4]. In this paper, designing DBCs using a semi-supervised boosting algorithm, by which the semi-supervised DBC learning is implemented efficiently, is considered.

2 AdaBoost and MarginBoost

In AdaBoost, which is the most well known boosting algorithm, an iterative learning procedure that combines many weak (base) classifiers to approximate the Bayes classifiers is performed. Starting with the unweighted training samples, the algorithm builds a classifier that produces class labels. If a training sample, x_i, is misclassified, the weight of that sample, $w_t(i)$, is increased (boosted). A second classifier is built using the new weights, $w_{t+1}(i)$, which are no longer equal. Again, misclassified samples have their weights boosted and the procedure is repeated - the hard examples receive high weights. After executing the above repetition, the final classifier is defined as the linear combination of the classifiers from each stage.

MarginBoost [3], an extension of AdaBoost, aims at improving the performance of an ensemble classifier, g_t, designed with weak base classifiers, $h_\tau \in \mathcal{H}$, by linear combination as follows: $g_t(x) = \sum_{\tau=1}^{t} \bar{\alpha_\tau} h_\tau(x)$, where $\bar{\alpha_\tau} = \frac{\alpha_\tau}{|\alpha_\tau|}$. The algorithm minimizes the cost function C defined with any scalar decreasing function c of the margin ρ: $C(g_t) = \sum_{i=1}^{l} c(\rho(g_t(x_i), y_i))$, where $\rho(g_t(x_i), y_i) = y_i g_t(x_i)$. Instead of taking exactly $h_{t+1} = -\bigtriangledown C(g_t)$, h_{t+1} is chosen, such that the inner product, $- < \bigtriangledown C(g_t), h_{t+1} >$, is maximal. On the basis of what we have briefly discussed, for $w_0(i) = 1/l$, $i = 1, \cdots, l$, and $g_0(x) = 0$, the MarginBoost algorithm is summarized in the following:

1. Learn a gradient direction h_{t+1} with a high value of $J_t^S = \sum_{i \in S} w_t(i) y_i h_{t+1}(x_i)$, where $S = \{(x_i, y_i)\}_{i=1}^{l}$.
2. If $J_t^S \leq \sum_{i \in S} w_t(i) y_i g_t(x_i)$ then return g_t exit.
3. Choose a step-length, α_t, for the obtained direction by a line-search or by fixing it as a constant, ϵ.
4. Add the new direction to obtain the following: $g_{t+1} = \frac{(|\alpha_t| g_t + \alpha_{t+1} h_{t+1})}{|\alpha_{t+1}|}$.
5. Update the weight, $w_{t+1} = \frac{c'(\rho(g_{t+1}(x_i), y_i))}{\sum_{j \in S} c'(\rho(g_{t+1}(x_j), y_j))}$.

3 Semi-Supervised MarginBoost

In semi-supervised learning, a large amount of unlabeled data, together with labeled ones, are used to build better classifiers. By introducing the way of generating the MarginBoost algorithm in semi-supervised approach, semi-supervised MarginBoost (SSMB) has been proposed in the literature [4]. A conventional algorithm for SSMB is formalized as follows, where labeled data (and labeled class), unlabeled data, and iteration number, which are input parameters for the algorithm, are given by $L = \{(x_i, y_i)\}_{i=1}^{n_l}$, $U = \{(x_j)\}_{j=1}^{n_u}$, and t_0, respectively. Also, as outputs of the algorithm, the class labels for the data and model of classifier are obtained:

1. $g_0(x) = 0$; $w_0(i) = \frac{1}{n_l + n_u}, i = 1, \cdots, n_l + n_u$.
2. Compute predicted labels of unlabeled data, U, using a nearest neighbor rule.
3. Do the following steps with increasing t by unity from 1 to t_1 per epoch: (a) Learn the gradient direction $h_t \in \mathcal{H}$ for the training data, $T (= L \cup U)$, with maximizing a value, $J_t^T = J_t^L + J_t^U$, where J_t^L and J_t^U are computed with L and U, respectively. (b) If $J_t^T \leq 0$, then exit and return $g_t(x)$; otherwise, go to the next sub-step. (c) After choosing a step-length α_t, update $g_t(x)$ as $g_{t+1}(x) \leftarrow g_t(x) + \alpha_t h_t(x)$. (d) Update the weight, w_{t+1}, for the next iteration t.

However, it is also well known that the unlabeled data do not always help the SSMB learning. From this consideration, in this paper, we report a preliminary experimental result, as shown in Table 1, obtained with an idea of incrementally using a set of strong unlabeled samples selected from the training set.

4 Experiments

The idea mentioned above has been tested and compared with conventional methods. This was done by

Table 1: A numerical comparison of the estimated error rates (standard deviations) of the original feature based classifiers (FBCs) and dissimilarity based classifiers (DBCs) trained with the four learning schemes. Here, the values underlined are the *lowest* error rates in the four ensemble classifiers per each iteration.

data sets	ensemble	number of	supervised learning		semi-supervised learning	
(*dim.*/#/*classes*)	classifiers	iterations	AdaBoost	MarginBoost	SSMB	SSMB2
Dataset1 (1024/1500/3)	FBC	10	0.3496 (0.0553)	0.1063 (0.0232)	0.1033 (0.0146)	<u>0.0581</u> (0.0096)
		30	0.2244 (0.0439)	0.0715 (0.0225)	0.0785 (0.0205)	<u>0.0474</u> (0.0166)
		50	0.2200 (0.0300)	0.0656 (0.0161)	0.0700 (0.0189)	<u>0.0456</u> (0.0122)
	DBC	10	0.4052 (0.0805)	0.1348 (0.0258)	0.1496 (0.0248)	<u>0.0819</u> (0.0166)
		30	0.2615 (0.0570)	0.1070 (0.0266)	0.0941 (0.0201)	<u>0.0863</u> (0.0169)
		50	0.2422 (0.0247)	0.0985 (0.0131)	0.0930 (0.0183)	<u>0.0844</u> (0.0123)
Dataset2 (216/2000/10)	FBC	10	0.7411 (0.0601)	0.0249 (0.0040)	0.0229 (0.0045)	<u>0.0140</u> (0.0027)
		30	0.4978 (0.0695)	0.0148 (0.0026)	0.0170 (0.0025)	<u>0.0110</u> (0.0027)
		50	0.4506 (0.0566)	0.0149 (0.0020)	0.0169 (0.0023)	<u>0.0119</u> (0.0023)
	DBC	10	0.7478 (0.0673)	0.0506 (0.0077)	0.0529 (0.0079)	<u>0.0380</u> (0.0074)
		30	0.5783 (0.0957)	0.0369 (0.0042)	0.0406 (0.0039)	<u>0.0314</u> (0.0030)
		50	0.4950 (0.0767)	0.0362 (0.0061)	0.0401 (0.0053)	<u>0.0333</u> (0.0051)

performing experiments on two benchmark databases, namely, Dataset1 and Dataset2.

The data set captioned Dataset1 (1024/1500/3) chosen from the NIST database [1] consists of three kinds of digits for a total of 1500 binary images. The data set named Dataset2 (216/2000/10) [2] consists of handwritten numerals represented in terms of the 216 profile correlations. Here, three numbers in brackets represent the numbers of dimensions, samples, and classes, respectively.

First, the data sets are split into three parts: labeled training sets, labeled test sets, and unlabeled data in the ratio 20 : 10 : 70. Then, the training and testing procedures, namely AdaBoost, MarginBoost, SSMB, and a modified SSMB (shortly SSMB2), are repeated 10 times and the results obtained are averaged. Especially, the four algorithms are performed in two fashions: feature-based classifications (FBCs) and dissimilarity-based classifications (DBCs). In FBCs, boostings have been performed in the input feature space, while the processes of DBCs have been carried out in the dissimilarity space.

Also, the scalar decreasing function and the step-length used for all the SSMB algorithms are $c(x) = e^{(-x)}$ and $\alpha_t = \frac{1}{4}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$, where ϵ_t is the error rate estimated with the test data at iteration t.

The observations obtained from the table are the following:

First, it should be observed that classification accuracy of the SSMB2 learning algorithm is generally better that those of the other method when utilizing the unlabeled data as well as labeled data (see the underlined numbers). Then, the classification accuracies of ensemble classifiers of FBCs are usually better than those of DBCs. Thus, finding an optimal or near optimal SSMB learning method for the DBC approach remains unchallenged.

In addition, it should also be pointed out that the error rates of AdaBoost and MarginBoost algorithms generally decrease as the number of the learning iterations increases. For SSMB2, however, no significant decrease in the error rate was shown as the number of iterations increased. Finally, in FDC, the reduction of dimensionality lead to a noticeable decrease in the error rate, while, in DBC, the reduction did not affect the error rate as much.

[1] http://www.nist.gov/srd/nistsd19.cfm

[2] http: //www.ph.tn.tudelft.nl/$\tilde{\ }$duin

5 Conclusion

In this paper, a semi-supervised MarginBoost algorithm applicable for dissimilarity-based classifications has been considered. The learning method, which is based on an idea of using strong unlabeled samples in an incremental fashion, was experimented and compared with the conventional methods for two well-known benchmark databases. The experimental results obtained demonstrated the possibility that the boosting algorithms could be utilized for the learning of dissimilarity-based classifiers. Especially, future research will address the improvement of the semi-supervised learning of DBCs.

Acknowledgments This work was supported by the National Research Foundation of Korea funded by the Korean Government (NRF-2011-0002517).

References

[1] Pekalska, E., Duin, R. P. W.: *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*. World Scientific, 2005.

[2] Kim, S. -W., Oommen, B. J.: On using prototype reduction schemes to optimize dissimilarity-based classification. *Pattern Recognition*, 40 2946–2957 (2007).

[3] Mason, L., Baxter, J., Bartler, P. L., Frean, M.: Functional gradient techniques for combining hypotheses. In *Advances in Large Margin Classifiers*, MIT, (2000).

[4] d'Alche-Buc, F., Grandvalet, Y., Ambroise, C.: Semi-Supervised MarginBoost. In *Advances in Neural Information Processing Systems*, 553–560 (2002).

[5] Mallapragada, P. K., Jin, R., Jain, A. K., Liu, Y.: SemiBoost: Boosting for Semi-Supervised Learning. *IEEE Trans. Pattern Anal. and Machine Intell.*, 31(11) 2000–2014 (2009).