A Multi-class Boosting Method for Learning from Imbalanced Data

Xiaohui Yuan* † and Mohamed Abouelenien†

*Faculty of Information Engineering
China University of Geosciences, Wuhan, China
†Department of Computer Science and Engineering
University of North Texas, Denton, Texas, U.S.A. 76203
E-mail: xiaohui.yuan@unt.edu
mohamed@unt.edu
*Corresponding author

Abstract: The acquisition of face images is usually limited due to policy and economy considerations, and hence the number of training examples of each subject varies greatly. The problem of face recognition with imbalanced training data has drawn attention of researchers and it is desirable to understand in what circumstances imbalanced data set affects the learning outcomes, and robust methods are needed to maximize the information embedded in the training data set without relying much on user introduced bias. In this article, we study the effects of uneven number of training images for automatic face recognition and proposed a multi-class boosting method that suppresses the face recognition errors by training an ensemble with subsets of examples. By recovering the balance among classes in the subsets, our proposed multiBoost.imb method circumvents the class skewness and demonstrates improved performance. Experiments are conducted with four popular face data sets and two synthetic data sets. The results of our method exhibits superior performance in high imbalanced scenarios compared to AdaBoost.M1, SAMME, RUSboost, SMOTEboost, SAMME with SMOTE sampling and SAMME with random undersampling. Another advantage that comes with ensemble training using subsets of examples is the significant gain in efficiency.

Keywords: Classification; Imbalanced Data; Boosting.

Reference

Biographical notes: Xiaohui Yuan received B.S. degree in Electrical Engineering from Hefei University of Technology, China in 1996 and Ph.D. degree in Computer Science from Tulane University in 2004. After graduation, he worked at the National Institutes of Health on medical imaging and analysis till 2006. He joined the University of North Texas and is currently an Associate Professor. His research interests include computer vision, data mining, machine learning, and artificial intelligence. His research findings are reported in over 80 peer-reviewed papers. Dr. Yuan is a recipient of Ralph E. Powe Junior Faculty Enhancement award in 2008 and the Air Force Summer Faculty Fellowship in 2011, 2012, and 2013. He also received two research awards and a teaching award from UNT in 2007, 2008, and 2012, respectively. He served in the editorial board of several international journals and as session chairs in many conferences..

Mohamed Abouelenien is currently a postdoc research fellow in electrical engineering and computer science department at University of Michigan, Ann Arbor. He received his BSc and MSc degrees in electronics and communication

engineering in 2005 and 2008 from the Arab Academy for Science and Technology in Egypt and his Ph.D. from department of computer science and engineering at University of North Texas in 2013. His research interest includes multimodal analysis, deception detection, machine learning, ensemble classification, image processing, pattern recognition, face recognition, computer vision, and dimensionality reduction.

1 Introduction

Face recognition (FR) is an active research problem and many methods have been developed to improve the robustness and accuracy of the automatic process. Despite the great improvement, the application of the methods to many real-world scenarios faces challenge of great variation in the number of training examples per human subject ([1, 2]).

In the training of a FR algorithm, the number of images for each human subject is usually assumed to be equal. This is only true in the controlled environment ([2]). However, there are many applications, in which the acquisition of face images is limited due to policy and economy considerations, and hence the number of training examples of each subject varies greatly. For example, images are taken from a terrorist in custody to provide an extensive reference for future recognition. On the other hand, a large number of people only have a couple of such face images captured on occasions such as application for a driver license or interview by an officer at customs and border protection. The training image set with abundant examples of some subjects, i.e., the majority classes, and much less number of examples of the others, i.e., the minority classes, exhibits the defining property of the imbalanced data set.

The problem of face recognition with imbalanced training data has drawn attention of researchers and new methods are developed. [1] incorporated a cost factor into the penalty function of Support Vector Machine (SVM). By assigning different costs to classes (i.e., subjects), the experiments demonstrated that the recognition of a person with less number of examples was improved. [3] proposed a doubly weighted non-negative matrix factorization method to account for pairwise similarity of face samples within a class and a discriminant score of image pixels. The between sample weight was claimed to be a significant factor to improve the performance given imbalanced training set. [4] proposed an imbalanced SVM to deal with skewed class boundary in face detection. Similar to the method presented in ([1]), a cost factor was used to penalize the misclassification of the minority examples, i.e., the examples from the minority classes [1].

Despite the efforts devoted to the algorithm development for learning from Imbalanced Data Set (IDS) problem in FR, it is desirable to understand in what circumstances IDS affects the FR learning outcomes, and, hence, proper algorithmic remedies can be devised. Robust methods are needed to maximize the information embedded in the training data set without relying on user introduced bias. In this article, we analyze the effects of IDS to the performance of a face recognition system and propose a multi-class boosting method that suppresses the face recognition errors by training an ensemble of classifiers with subsets of examples. By recovering the balance among classes in the subsets, the proposed method circumvents the class skewness and demonstrates improved performance.

The rest of this article is organized as follows: Section 2 reviews the related work in multi-class boosting methods and ensemble for learning from imbalanced data. Section 3 describes

our proposed boosting-based, multi-class classification approach that takes advantage of data sampling and weight adjustment. Section 4 presents our experimental results and discussion. Section 5 concludes the paper.

2 Related Work

Boosting methods were designed to solve binary classification problems. Directly applying a boosting method to a multi-class problem, e.g., face recognition, is not straightforward. Intuitive solutions include translating a multi-class problem into several binary classification problems using one-against-all or one-against-one strategies ([5]). Using one-against-all strategy, one model is constructed for each class; the one-against-one strategy constructs one model for each pair of classes ([6]). [7] introduced Adaboost.MH, which employed a hamming loss to represent the average error rate for the weak hypothesis over all the binary predictions. Using extra bits to encode class labels, the ensemble was able to tolerate mistakes made by a small number of classifiers ([8]). Guruswami and Sahai extended AdaBoost.OC ([9]) and proposed AdaBoost.ECC ([10]) which replaced pseudo loss with a common measurement to evaluate the training error. [11] proposed an extension to AdaBoost by including the number of classes in the classifier weight. The accuracy of each classifier only needs to be better than random guess (i.e., $1/K$). [12] proposed a generalization framework by employing an additive factor to the accuracy, which filled the theoretical gap of error relaxation for the base classifiers in multi-class boosting.

In many real-world applications training data are usually uneven among classes. To address the problems of learning from IDS, one thrust of efforts focuses on using cost matrix. [13] introduced AdaUBoost that modified the weight updating rule and loss function such that the minority examples were emphasized with higher weights. A similar strategy was used in ([14, 15]) to boost multiple base-classifiers with asymmetric misclassification costs. [14] described three variations of cost-sensitive boosting, each of which used a cost factor to modify examples' weights. [16] introduced "acceleration" to the weight updating rule. The weight of a costly example receives greater increment when it is misclassified, and decreases less otherwise. [17] treated the misclassified minority examples and majority examples differently and proposed a confusion matrix-based weight to account for various difficulties in classifying rare classes.

Among the boosting methods for learning from IDS, sampling strategies have been heavily explored to create balanced training data sets. [18] introduced SMOTEboost that generated synthetic minority examples using SMOTE strategy during training. [19] combined boosting and data generation and introduced the DataBoost-IM method, where hard-to-classify instances from both majority and minority classes were identified and used to generate synthetic examples. A similar idea of creating synthetic examples was also employed in E-Adsampling algorithms ([20]). Compared to DataBoost-IM, which led to the creation of a large number of synthetic minority examples, E-Adsampling faced possible loss of the originally misclassified examples. [21] proposed RAMOBoost that ranked minority examples during boosting iteration and created synthetic minority examples based on a distribution function. Seiffert et al. proposed the RUSboost ([22]) that extended the AdaBoost methods by using random under-sampling to select subsets of examples. It was demonstrated that the performance of RUSboost was comparable to SMOTEboost.

Both cost embedding and sampling strategies improve binary classification using imbalanced training data. Extending to multi-class boosting, however, is not forthright ([12]). In addition, in the application of face recognition, many state-of-the-art learning methods, e.g., LDA and Eigenface, produce stable classification results which abate the driving force of boosting strategy: diversity ([23]). Our proposed multi-class boosting method addresses the problems of learning from imbalanced data and enabling employment of stable learners in the ensemble.

3 Multi-class Boosting for Learning from Imbalanced Data

In a multi-class classification problem, let the number of classes be K. The labels can then be encoded with values 1 and $-\frac{1}{K-1}$. For example, for an instance $\mathbf{x}_i$ that belongs to class 2, its label is expressed as $\{-\frac{1}{K-1}, 1, -\frac{1}{K-1}, \ldots, -\frac{1}{K-1}\}^T$, where the value of the second component of the vector is 1 indicating that this example belongs to class 2 and the rest are $-\frac{1}{K-1}$.

Directly extending AdaBoost to address multi-class, imbalanced problems fails due to the stringent constraint on the performance of the weak learner ([24, 11]). Given a multi-class data set, it is reasonable to assume equal probability for a random guess to label an instance to one of the K classes. Hence, the expected error is $1 - \frac{1}{K}$. The empirical error ϵ can then be expressed as the average error over all classes:

$$\epsilon = \frac{1}{M}\sum_{j=1}^{K}(\sum_{i=1}^{M_j}\frac{K-1}{K}) \tag{1}$$

where M_j is the number of examples in class j and $M = \sum_j M_j$.

Provided with a multi-class, imbalanced data set, a classifier trained with an imbalanced data set could result in greater generalization error than a classifier trained with a balanced data set due to the dominating number of examples in the majority classes ([6]). Following the same error minimization strategy, the classifier yields into the region of the minority class. Clearly, the cause of the suboptimal classifier is the uneven number of examples in the class overlap. Ideally, if the balance is restored in this region, the bias will diminish.

Another issue arises from stable learners that are frequently used in face recognition applications. For instance, Eigenface method constructs a subspace from the training examples and a face recognized by finding the nearest neighbor in the projected subspace. Hence, when a data set S is used to train such a face recognizer $f(\mathbf{x})$, the evaluation error over set S is close to zero. Knowing that the weight update is driven by error, we can expect little, if not zero, changes in the weights for the next training round.

To address both issues of uneven data size induced bias and stable learner, we propose a multi-class boosting method (multiBoost.imb). Our method is presented in Algorithm 1. In multiBoost.imb, we introduce a perturbation strategy that selects a subset of examples from the majority classes according to the data distribution. The selected examples and the minority examples form a training set. Let $|S_I|$ denote the smallest class size. Following the data distribution w_i ($w_i : (\mathbf{x}_i, \mathbf{y}_i) \in S_A$), a subset of examples from each majority class S_A, denoted with S'_A, is randomly selected so that the size of this subset equals the size of the minority class, i.e., $|S'_A| = |S_I|$. The selected majority examples and the minority examples form a subset S' for training a weak learner:

$$S' = \{\cup S_{I_p}, \cup S'_{A_q}\} \text{ and } |S'_{A_q}| = |S_{I_p}|, \tag{2}$$

where p is the index of the minority classes and q is the index of the majority classes. S_{I_p} denotes the set of examples in the minority class p; S'_{A_q} denotes the subset of examples of the majority class q. The changing subset of training example ensures the construction of a group of diverse classifiers even with stable weak learners. In addition, the equal number of examples that represents all classes suppresses the influence of the IDS to the construction of a classifier f^t.

Unlike the training process, the entire data set S is used in the evaluation of each classifier f^t. This is necessary because not only the data distribution needs to be updated, but also the weight α^t to the classifier f^t has to be consistent to the overall performance of the learner. Without the knowledge of the underlying true data distribution and hence the overlapped regions among classes, empirical error is a reasonable metric.

The weight α^t determines how much a learner f^t contributes to the final decision as shown in Eq. (6). Given an IDS, the great empirical error of an unbiased learner results in a smaller weight assignment. In fact, as training continues, the examples within the overlapped regions are likely to have greater probabilities. To suppress possible over-weighting the biased classifier, an attenuation factor γ ($\gamma \geq 1$) is included in the weight calculation (see Eq. (4), which are more likely to happen in the later stage of the training. Large γ subsides the impact of empirical error ϵ^t. When $\gamma = 1$ the weight calculation reduced to AdaBoost.M1 ([7]); whereas when $\gamma = K - 1$ the weight becomes that of the SAMME algorithm in ([11]).

Assuming that classifiers are trained independently, the majority voting of an ensemble should lead to better results than using a single classifier ([24]). This suggests that the weight of classifiers that perform better than random guessing should be positive. Hence, the ratio $\frac{\gamma(1-\epsilon)}{\epsilon}$ has to be greater than one. Following this assumption, the maximum acceptable error rate for a weak learner is bounded by

$$\epsilon < \frac{\gamma}{\gamma + 1}. \tag{7}$$

In contrast to AdaBoost, the inclusion of γ improves the error tolerance. If we relax our requirement of the error rate of the weak learners to be equivalent to that of the random guess, i.e.,

$$\epsilon = \frac{K-1}{K}, \tag{8}$$

combining with Eq. (7) results in the upper bound for γ, i.e.,

$$\gamma = K - 1. \tag{9}$$

Hence, the choice of γ lies in the range of $[1, K-1]$.

The multiBoost.imb method updates data distribution following the exponential function as shown in Eq. (5). Given that the class label is encoded as a vector that consists of 1 and $\frac{-1}{K-1}$ ([11]), where the index of 1 indicates the class label, the dot product $\mathbf{y}_i f(\mathbf{x}_i)$ yields one of the following two values:

$$\mathbf{y}_i f(\mathbf{x}_i) = \begin{cases} \frac{K}{K-1} & \text{if } \mathbf{x}_i \text{ is correctly classified} \\ \frac{-K}{(K-1)^2} & \text{if } \mathbf{x}_i \text{ is misclassified} \end{cases} \tag{10}$$

Algorithm 1 MultiBoost.imb

Input: an imbalanced data set that consists of p minority classes and q majority classes: $S = \{\cup S_{I_p}, \cup S_{A_q}\}$.
Initialize the weight w_i for each $(\mathbf{x}_i, \mathbf{y}_i)$ with $\frac{1}{M}$.
for $t = 1, 2, \ldots, T$ **do**
  Construct a training set S' following Eq. (2).
  Train a classifier f^t using S' such that error is minimized.
  Compute error of f^t using the entire data set S:

$$\epsilon^t = \sum_{i=1}^{M} w_i^t [\![f^t(\mathbf{x}_i) \neq \mathbf{y}_i]\!] \tag{3}$$

  where $[\![\cdot]\!]$ is the indicator function that returns 1 if the argument is true.
  if $\epsilon^t \geq \frac{\gamma}{1+\gamma}$, then stop and set $T = t - 1$
  Compute the weight α^t for f^t:

$$\alpha^t = \log \frac{\gamma(1 - \epsilon^t)}{\epsilon^t} \tag{4}$$

  Update and normalize data distribution

$$w_i^{t+1} = \frac{w_i^t e^{-\frac{1}{2}\alpha^t \mathbf{y}_i f^t(\mathbf{x}_i)}}{W^t}, \tag{5}$$

  where $W^t = \sum_i w_i^t$.
end for
The ensemble $F(\mathbf{x})$ aggregates f^t by maximizing the weighted sum:

$$F(x) = \arg\max_k (\sum_{t=1}^{T} \alpha^t f^t(\mathbf{x})) \tag{6}$$

where $k \in [1, \ldots, K]$.

It is clear that the update to w_i of a misclassified instance is smaller than that of a correctly classified instance. The gradually increased w_i of a misclassified instance is consistent with the relaxed constraint on the error rate. Hence, it prevents over emphasizing the large number of misclassified majority instances.

Given that the normalized data distribution w_i sums to one, we can express the sum of the data distribution as follows:

$$\sum_i w_i^{t+1} = \sum_i w^t \frac{e^{-\alpha^t \mathbf{y}_i f^t(\mathbf{x}_i)}}{W^t} = \frac{1}{ZM} \sum_{i=1}^{M} \prod_{s=1}^{t} e^{-\alpha^s \mathbf{y}_i f^s(\mathbf{x}_i)}$$
$$= \frac{1}{ZM} \sum_{i=1}^{M} e^{-\mathbf{y}_i \sum_{s=1}^{t} \alpha^s f^s(\mathbf{x}_i)} = \frac{1}{ZM} \sum_{i=1}^{M} e^{-\mathbf{y}_i f^*(\mathbf{x}_i)} = 1.$$

where W^t is a normalization factor and $Z = \prod_{s=1}^{t} W^s$. Hence, the product of the normalization factor equals to the normalized sum of weight updates following AdaBoost:

$$Z = \frac{1}{M} \sum_{i=1}^{M} e^{-\mathbf{y}_i f^*(\mathbf{x}_i)} \tag{11}$$

where $f^*(\mathbf{x}_i) = \sum_{s=1}^{t} \alpha^s f^s(\mathbf{x}_i)$ is an intermediate ensemble.

When an instance is misclassified, i.e., $[\![f^t(\mathbf{x}_i) \neq \mathbf{y}_i]\!] = 1$, the function $e^{-\mathbf{y}_i f^*(\mathbf{x}_i)} > 1$. Together with Eq. (3), we have the upper bound of the error as the product of the normalization factors $\epsilon \leq \prod_s W^s$.

To find appropriate α, we minimize this error bound $\prod_s W^s$. Following the definition of W^t, we have

$$\prod_s W^s = \prod_s (\sum_i w_i^t e^{-\alpha^t \mathbf{y}_i f^t(\mathbf{x}_i)}) \tag{12}$$

Notice that $\mathbf{y}_i f^t(\mathbf{x}_i)$ results in two values as shown in Eq. (10), which is equivalent to $\frac{1}{2} f^t(\mathbf{x}_i) f^t(\mathbf{x}_i) \lambda (h^*(\mathbf{x}_i) - q)$, where $f^t(\mathbf{x}_i) f^t(\mathbf{x}_i) = \frac{K}{K-1}$ and q is a threshold. That is, the product of the true label and classification result is expressed as function $\frac{\lambda}{2}(q - h^*(\mathbf{x}_i))$ that gives the following results:

$$\frac{\lambda}{2}(h^*(\mathbf{x}_i) - q) = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ is correctly classified} \\ \frac{-1}{(K-1)} & \text{if } \mathbf{x}_i \text{ is misclassified} \end{cases}$$

where h^* outputs 1 or -1 when $\mathbf{x}_i$ is classified correctly or incorrectly; λ is the inverse of the uninformative error rate, i.e., $\lambda = \frac{K}{K-1}$; q sets the threshold for deciding weight changes. Hence, combining with Eq. (8), we have

$$\mathbf{y}_i \mathbf{y}_i \lambda = f^t(\mathbf{x}_i) f^t(\mathbf{x}_i) \lambda = 1/\epsilon^2.$$

Based on the convexity of the exponential functions, the error upper bound in Eq. (12) is expressed as follows:

$$\prod_s w_i^t e^{-\alpha^t f^t(\mathbf{x}_i) f^t(\mathbf{x}_i) \frac{\lambda}{2}(h^*(\mathbf{x}_i) - q} = \prod_s w_i^t e^{-\alpha^t \frac{1}{\epsilon^2}(h^*(\mathbf{x}_i) - q)}$$
$$\leq \prod_s \sum_i w_i^t (h^*(\mathbf{x}_i) e^{-\alpha^t \frac{1}{\epsilon^2}(1-q)} + (1 - h^*(\mathbf{x}_i)) e^{\alpha^t \frac{1}{\epsilon^2} q}) \quad .$$

Taking the first derivative with respect to α^t and setting it to zero, we have the expression of α^t:

$$\alpha^t = \ln \frac{\gamma(1-\epsilon)}{\epsilon} \tag{13}$$

where $\gamma = \frac{1-q}{q}$.

Note that, in training a classifier, only a portion of examples from the majority classes are used. Hence, the error minimization is in the context of a subset of balanced training examples. However, the α given in Eq. (13) is subject to the entire training data set, which accounts for the entire data set.

4 Experiments and Discussion

4.1 Experiment Settings

We employed the Eigenface ([25]) and Fisherface ([26]) for face recognition and use four public face databases. The AT&T data set consists of 40 subjects with 10 images for each. The AR face database consists of 126 subjects (among which we used 50 to be consistent with the other two face data sets) and 11 images were cropped for each subject. The Yale database consists of 38 subjects and 65 images for each. LFW has more than 13,000 face images of over 5,000 subjects. The majority of the subjects have less than three images. In our experiments, we used the LFW images aligned with deep funneling method ([27]) and randomly selected 40 subjects, each of which has at least 20 images such that we can form different imbalance ratios and perform cross validation.

To simulate imbalanced training data, half of the classes (or subjects) were used as the majority classes, and the other half were treated as the minority classes. By re-sampling the data sets, we created training data with various imbalance ratios. The average performance of the leave-one-out cross validation serves as the baseline in our studies.

Cross-validation was used. Depending on the data set size, the number of folds varies. For example, AT&T database consists of 40 subjects. In the experiments of learning from imbalanced data set with imbalance ratio $\beta = 2$, five examples of each subject from 1 through 20 were randomly selected, and the other five were used as testing examples. Subjects 21 through 40 were treated as the majority classes and, based on the imbalance ratio, ten examples were used for each subject in the training. The majority and the minority classes were switched in another experiment. Experiments were designed to reveal the effects of IDS with respect to the imbalance ratio and the difficulty of the problems. We focused on the evaluation of classifying the minority classes since that is the origin of most errors.

State-of-the methods were used in our comparison study. Extension of SMOTEboost for multi-class problem was developed based on AdaBoost.M2. In our initial study that follows this extension ([18]), it took more than 24 hours to complete the training of one SMOTEboost ensemble of 10 base learners and the performance is no better than SMOTEboost using AdaBoost.M1 framework. Hence, the results reported in this comparison study for SMOTEboost is based on AdaBoost.M1. RUSboost, on the other hand, was developed for binary-class classification. For the comparison purpose, we extend it again following the spirit of AdaBoost.M1. We also limit our ensembles to 10 base learners due to great time expense for cross-validation.

Table 1 The average error rate and standard deviation of multiBoost.imb with imbalanced face data sets.

Data Sets	S_i (β)	Average Error Rate (%)							
		Base	AdaBoost .M1	SAMME	multiBoost .imb	RUS boost	SMOTE boost	SAMME +RUS	SAMME +SMOTE
Eigenface									
AT&T	5 (2)	2.5	10.0	10	**8.8**	9.5	**8.8**	9.5	**8.8**
		(6.3)	(16.8)	(16.8)	(14.5)	(15.2)	(14.2)	(15.2)	(14.2)
	2 (5)		22.9	22.9	**18.1**	18.5	20.9	19	20.9
			(18.3)	(18.3)	(15.9)	(15.6)	(16.6)	(16.3)	(16.7)
AR	5 (2)	19.5	27.4	27.4	25.1	25.6	**25**	25.6	**25**
		(15.0)	(29.5)	(29.5)	(29.1)	(28.7)	(29.7)	(29.3)	(29.7)
	2 (5)		49.4	49.6	**40.8**	41.5	47.6	41.9	47.5
			(20.2)	(20.2)	(19.7)	(21.5)	(20.6)	(19.6)	(20.5)
Yale	32 (2)	28.5	76.9	76.8	**75.1**	75.7	76.2	75.2	76.3
		(10.2)	(7.2)	(7.2)	(6.4)	(6.9)	(6.9)	(6.8)	(7.2)
	8 (8)		88.9	88.9	**82.4**	84.1	88.1	84.9	88.2
			(3.5)	(3.5)	(3.5)	(3.6)	(3.3)	(3.6)	(3.5)
LFW	5 (2)	83	90.5	90.5	88.6	89	81.8	88.4	**81.1**
		(12.2)	(10.2)	(11.6)	(12.7)	(11.9)	(13.6)	(12.4)	(13.2)
	2 (5)		95	95	**87.5**	97.5	94.9	89.4	94.9
			(4.8)	(4.8)	(7)	(15.8)	(4.8)	(6.2)	(4.8)
Fisherface									
AT&T	5 (2)	2	14.3	14.3	10.8	10.8	13.8	**9**	13.8
		(4.1)	(20.6)	(20.6)	(18.2)	(17.6)	(22.4)	(15)	(22.4)
	2 (5)		40.7	40.7	**16.3**	**16.3**	35.9	18.4	36.2
			(22.9)	(22.9)	(14)	(13.8)	(22.6)	(14.7)	(22.2)
AR	5 (2)	2.8	30.8	30.8	18.6	**17.2**	30.2	18.6	30.2
		(5.7)	(32.5)	(32.5)	(25.1)	(25.9)	(29.2)	(26.3)	(29.2)
	2 (5)		46.3	46.3	**17.6**	17.8	40.3	20.3	40
			(22.1)	(22.1)	(16.7)	(16.9)	(21.1)	(17.1)	(22)
Yale	32 (2)	4.6	31.4	31.4	**28.1**	29.9	29.6	35.5	29.7
		(2.5)	(13.1)	(13.1)	(12)	(11.9)	(12.3)	(12.5)	(12.9)
	8 (8)		58.7	58.7	59.2	60.1	**50.7**	63.5	50.8
			(8.4)	(8.4)	(7.3)	(7.3)	(8.5)	(6.3)	(7.9)
LFW	5 (2)	66	84.5	84.5	**69.1**	69.5	86.1	71.4	84.5
		(18.9)	(16.4)	(16.4)	(21.3)	(21.2)	(16.8)	(20.2)	(16.4)
	2 (5)		97.5	97.5	**84.1**	96.3	96.7	84.7	96
			(5.6)	(5.6)	(11.9)	(15.8)	(6)	(11.5)	(5.7)

4.2 Performance Analysis

In our performance analysis, we focus on classification of the minority classes under different imbalance ratios since the minority classes are usually the important ones in a FR problem. Table 1 summaries the average error rate (across all classes in each data set) for the testing scenarios. The standard deviation is included in the parenthesis. Since the baseline is the best result using leave-one-out cross-validation, there is only one baseline error for each data set. The bold face font highlights the best average performance in each case. Among all scenarios, multiBoost.imb achieved 4 best performance out of 8 low imbalance ratio cases (four data sets with two different base learners) and 7 best performances out of 8 higher imbalance ratio cases. The maximum improvement as compared to the second best performance among all other methods is 12.8% in the high imbalance ratio cases and 8% in the low imbalance ratio cases. It is worth of noting that with low imbalance ratio, i.e., $\beta = 2$,

multiBoost.imb achieved highly competitive results against the baseline performance using both base learners.

Clearly, imbalance affects base learners differently. It is interesting to note that AT&T, AR, and Yale are fairly easy cases when Fisherface is applied to the balanced data sets. The average error rates of the baseline performance as reported in Table 1 are 2%, 2.8%, and 4.6% for AT&T, AR, and Yale, respectively. However, if the training data is imbalanced, ensembles using Fisherface as base learner degrade significantly in their performance.

It is evident that our proposed method multiBoost.imb achieves better results when the imbalance ratio is higher. In contrast to the low imbalance ratio, multiBoost.imb also achieved better performance with a couple of classes compared to the baseline. It is worth noting that RUSboost failed to create an effective classifier ensemble in the higher imbalance ratio with LFW data set. RUSboost mostly ignored all subjects except one in the case of LFW data set with $\beta = 5$. However, when we combined the random undersampling with SAMME, the performance improved greatly.

Among all other methods, the combination of SAMME with SMOTE sampling method yielded competitive performance in low imbalance ratio cases (4 out of 8 cases). It is interesting to note that SAMME with SMOTE exhibited even lower average error rate compared to the baseline with slightly greater STD. Although RUSboost and SMOTEboost have very close average error rate in high imbalance ratio with LFW data set, the performance of SMOTEboost is better than RUSboost.

4.3 The Attenuation Factor

One key parameter in our proposed method is the attenuation factor γ. Besides the benefit it introduces to relax the error upper bound for each learner and, hence, avoids early termination, it contributes a constant addition to the learner weight, which diversifies the ensemble. However, with larger γ, the ensemble becomes less likely to terminate due to the relaxed error upper bound. So it is important to find the appropriate values for γ.

We studied our method by varying γ value from 1 to $2K$, and the experiment using the combination of each data set and a γ value is repeated 6 times. Both KNN and decision trees are used as the base learner and the imbalance ratio include low ratio value, i.e., $\beta = 2$ and high ratio values ($\beta = 5$ for AT&T and AR and $\beta = 8$ for Yale). The average error rate of the ensemble with different base learner varies fairly greatly. Depending on the cases such difference can be up to 40%. Also, the imbalance ratios result in performance margin in the range of 10%. Hence we use the average error rate improvement as an indicator to reveal the trend of γ. In calculate the improvement rate, we use the error rate with $\gamma = 1$ as the base.

Fig. 1 depicts the average error rate improvement. Because the number of classes in each data set differs, the range of each curve varies accordingly. The results with $\gamma = K$ is marked with an enlarged double-line symbol in red. When $\gamma = 1$, multiBoost.imb degrades to AdaBoost.M1, and when $\gamma = K - 1$, multiBoost.imb becomes SAMME. It is clear that when $\gamma = 1$ all base learners and ensembles result in the greatest error rate. As γ increases, the error rate reduces. This trend continues even beyond $K - 1$. At $\gamma = 2K$, the error rate remains relatively small with little fluctuation from $\gamma = K - 1$. However, it is clear that the error rate gives suboptimal results when $\gamma = K - 1$. The best performance is mostly achieved when the attenuation factor is in the range of (1, K-1). Clearly, an attenuation factor that is far greater than K-1, however, does not result in significantly degraded performance. This is in part because the weight (or the contribution ratio) of the base learner becomes

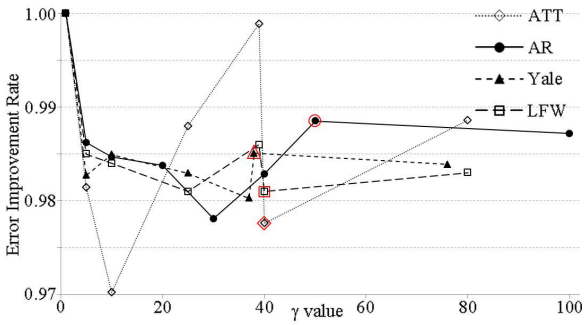

Figure 1 The average error rate with respect to the attenuation factor. The double-line symbols highlight the average error rate with $\gamma = K$.

less dependent on the learner performance but the attenuation factor. That is, the ensemble becomes a collection of equally weighted weak learners trained with subsets of examples.

Table 2 The average training time in seconds.

Base Learner	Data sets	β	AdaBoost .M1	SAMME	multiBoost .imb	RUSboost	SAMME +RUS
Eigenface	AT&T	2	27.59	26.04	25.93	25.92	17.14
		5	19.66	19.73	11.88	11.81	7.42
	AR	2	48.07	43.95	48.48	48.66	31.61
		5	36.87	32.66	19.04	19.27	10.39
	YALE	2	38.34	36.05	38.24	37.34	31.31
		8	30.17	32.34	13.96	13.75	8.83
	LFW	2	68.71	68.9	50.57	52.18	37.87
		5	50.22	47.96	24.28	22.59	13.99
Fisherface	AT&T	2	37.21	34.07	32.66	32.51	23.34
		5	27.57	27.56	18.43	18.45	13.06
	AR	2	56.38	56.33	55.19	55.08	43.52
		5	44.05	45.08	27.49	27.36	18.78
	YALE	2	56.67	55.62	52.91	55.08	46.99
		8	59.63	47.66	20.99	21.95	15.57
	LFW	2	91.66	82.69	62.3	61.66	48.14
		5	56.74	59.33	31.55	28.92	20.67

4.4 Efficiency Analysis

Table 2 and 3 present the average training time. For SMOTEboost and SAMME with SMOTE sampling, we recorded both the sampling and learning times. Between two base learners, there is no significant difference in efficiency although Fisherface based methods took slightly longer time in comparison to Eigenface based methods.

Because less number of examples used to train each base learner, undersampling based methods took much shorter time to complete model creation. Data resampling takes time, which is trivial compared to the learning time, and we report the sampling and training times together. The maximum undersampling time is less than 5 seconds. Compared to AdaBoost.M1 and SAMME, multiBoost.imb, RUSboost, and SAMME with random undersampling used almost equivalent amount of time in the low imbalance ratio cases and about 50% less amount of time in the high imbalance ratio cases. The slight advantage

Table 3 The average training time in seconds.

Base Learner	Data sets	β	SMOTEboost		SAMME+SMOTE	
			learning	sampling	learning	sampling
Eigenface	AT&T	2	35.14	1.62	33.85	3.5
		5	39.09	628.08	39	630.07
	AR	2	65.61	2.94	70.3	3.02
		5	83.51	1026.04	64.69	1004.94
	YALE	2	51.92	2.90	46.10	2.97
		8	44.64	372.58	43.35	368.02
	LFW	2	91.07	2.93	96.29	3.31
		5	121.71	126.24	117.92	126.41
Fisherface	AT&T	2	46.16	1.65	47.71	1.72
		5	51.13	629.73	58.46	641.88
	AR	2	82.85	2.96	82.51	3.01
		5	82.07	1002.49	79.85	1005.34
	YALE	2	62.52	3.02	58.36	3.01
		8	53.12	365.97	60.28	372.58
	LFW	2	110.61	2.97	117.54	3.09
		5	140.91	126.01	126.05	126.17

in efficiency of undersampling based method comes from its simplicity. That is, random undersampling during training iterations is independent and no training history is consulted.

On the other hand, oversampling based methods, e.g., SMOTEboost and SAMME with SMOTE, clearly require a significant amount of time, especially when there is a high imbalance ratio, to resample the training set. The sampling time can take as high as 94% of the overall training time. Even without considering the sampling time used in SMOTEboost and SAMME with SMOTE, the learning time was much greater due to larger number of training examples.

Comparing time used for all face data sets, we can see that the increment of time in high imbalance ratio case is less dramatic. This is because each face example consists of more than ten thousands features. The high dimensionality requires significantly more time in training. When the imbalance ratio is low, this time for SMOTEboost is relatively small (in the order of 3 seconds). However, in the case of large imbalance ratio, the time cost to create new artificial examples is substantial for SMOTEboost. The exponential time increment makes SMOTEboost a less attractive method to handle highly imbalanced, high-dimensional problems.

5 Conclusion

In this article, we propose multiBoost.imb method that greatly improves the performance to learn from imbalanced data without relying on user introduced bias. Experimental results demonstrated that the error rate of multiBoost.imb is consistently lower than that of AdaBoost.M1. The advantage becomes more apparent when the imbalance ratio enlarges. In some cases our method achieves even lower error rates beyond that of the baseline model, which is trained with all available examples. With our downsampling strategy, stable classifiers such as Eigenface can be employed as the base learner in an ensemble. Our studies of the attenuation factor show that the best performance is mostly achieved when the attenuation factor is within the range of (1, K-1). An attenuation factor that is far greater than K-1, however, does not result in significantly degraded performance.

The comparison study was conducted with respect to the state-of-the-art sampling-based ensemble methods for imbalanced data sets. When learning from balanced data sets or ones with low imbalance ratio, the performance of the compared methods is similar. However, the improvement is substantial when imbalance ratio is high. In contrast to underampling based methods, multiBoost.imb takes much less number of training iterations to achieve a performance that takes RUSboost many times more iterations to attain. Efficiency analysis shows that multiBoost.imb and random undersampling based methods demonstrated similar efficiency given the same number of base learners while oversampling based methods exhibited a poor efficiency due to the excessive amount of time used to create and train with the additional synthetic examples. In the cases of large imbalance ratio, the extra time it takes SMOTEboost to create the training time increases exponentially, which makes it a less attractive method to handle highly imbalanced, high-dimensional problems.

References

[1] Y.-H. Liu and Y.-T. Chen. Face recognition using total margin-based adaptive fuzzy support vector machines. *IEEE Transactions on Neural Networks*, 18(1), 2007.

[2] Y. Zhang and Z.-H. Zhou. Cost-sensitive face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10), 2010.

[3] J. Lu and Y-P. Tan. A doubly weighted approach for appearance-based subspace learning methods. *IEEE Transactions on Information Forensics and Security*, 5(1):71–78, March 2010.

[4] Y.-H. Liu, Y.-T. Chen, and Shey-Shin Lu. Face detection using kernel pca and imbalanced svm. In *Lecture Notes in Computer Science , International Conference on Natural Computation 2006*, volume 4221, page 351 ¨C 360, 2006.

[5] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

[6] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *journal of computer and system sciences*, 55(119–139), 1997.

[7] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. In *Machine Learning*, pages 80–91, 1999.

[8] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[9] Robert E. Schapire. Using output codes to boost multi-class learning problems. In *Proceedings of the 14th International Conference on Machine Learning*, pages 313–321, 1997.

[10] Venkatesan Guruswami and Amit Sahai. Multiclass learning, boosting, and error-correcting codes. In *Proceedings of the 12th Annual Conference on Computational Learning Theory*, pages 145–155, 1999.

[11] Ji Zhu, Hui Zou, Saharon Rosset, and Trevor Hastie. Multi-class adaboost. *Statistics and Its Interface*, 2:349–360, 2009.

[12] Indraneel Mukherjee and Robert E. Schapire. A theory of multiclass boosting. *Proceedings of Twenty-Fourth Annual Conference on Neural Information Processing Systems*, 2010.

[13] Grigoris Karakoulas and John Shawe-Taylor. Optimizing classifiers for imbalanced training sets. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 253–259, Cambridge, MA, USA, 1999. MIT Press.

[14] Yanmin Sun, Mohamed S. Kamel, Andrew K.C. Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40:3358 – 3378, 2007.

[15] Benjamin X. Wang and Nathalie Japkowicz. Boosting support vector machines for imbalanced data sets. *In Foundations of Intelligent Systems*, pages 38–47, 2008.

[16] Wei Fan, Salvatore J. Stolfo, Junxin Zhang, and Philip K. Chan. Adacost: Misclassification cost-sensitive boosting. In *16th International Conference on Machine Learning*, 1999.

[17] Mahesh V. Joshi, Vipin Kumar, and Ramesh C. Agarwal. Evaluating boosting algorithms to classify rare classes: Comparison and improvements. In *First IEEE International Conference on Data Mining*, pages 257–264, 2001.

[18] Nitesh V. Chawla, Aleksandar Lazarevic, Lawrence O. Hall, and Kevin W. Bowyer. Smoteboost: Improving prediction of the minority. In *Seventh European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 107–119, 2003.

[19] Hongyu Guo and Herna L Viktor. Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *SIGKDD Explorations*, 6(1):30–39, June 2004.

[20] Ordonez Jon Geiler, Li Hong, and Guo Yue-jian. An adaptive sampling ensemble classifier for learning from imbalanced data sets. In *International MultiConference of Engineers and Computer Scientists*, volume 1, March 2010.

[21] S. Chen, H. He, and E. A. Garcia. RAMOBoost: Ranked minority oversampling in boosting. *IEEE Transactions on Neural Networks*, 21(10):1624–1642, 2010.

[22] Chris Seiffert, Taghi M. Khoshgoftaar, Jason Van Hulse, , and Amri Napolitano. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transaction on systems, man, and Cybernetics, Part A: Systems and Humans*, 40(1), January 2010.

[23] J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, and S. Z. Li. Ensemble-based discriminant learning with boosting for face recognition. *IEEE Transactions on Neural Networks*, 17(1):166–178, 2006.

[24] Gunther Eibl and Karl-Peter Pheiffer. Multiclass boosting for weak classifiers. *Journal of Machine Learning Research*, 6:189–210, 2005.

[25] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[26] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[27] G. B. Huang, M. Mattar, H. Lee, and E. Learned-Miller. Learning to align from scratch. In *Advances in Neural Information Processing Systems (NIPS)*, Lake Tahoe, Nevada, United States, December 3–6 2012.