Journal of Information Hiding and Multimedia Signal Processing ©2011 ISSN 2073-4212
Ubiquitous International Volume **2**, Number **1**, January **2011**

Boosting Approach for Score Level Fusion in Multimodal Biometrics Based on AUC Maximization

Mehdi Parviz and M. Shahram Moin

Multimedia Systems Research Group
Iran Telecommunication Research Center
End of North Kargar Street, Tehran, Iran 1439955471
mparviz@itrc.ac.ir; moin@itrc.ac.ir

Received April 2010; revised August 2010

Abstract. *We investigate AdaBoost and bipartite version of RankBoost abilities to minimize AUC and its application for score level fusion in multimodal biometric systems. To do this, we customize two methods of weak learner training. Empirical results show comparable AUC for AdaBoost and RankBoost.B which previously was addressed theoretically. We demonstrate exhaustive results among state of the art classifiers and techniques, e.g., SVM, GMM and SUM rule in this area. AdaBoost and RankBoost.B achieve significant performance improvement compared to GMM and SUM rule, and the performance comparable to SVM. Besides empirical results, we show that, instead of adding a constant weak learner in order to maximize AUC using AdaBoost, instances could be weighted initially in each class inversely proportional to the number of instances in the corresponding classes.*
Keywords: Multimodal, Biometrics, AUC, Boosting.

1. **Introduction.** Nowadays, as a result of advancement in computation power and storage capacity of computers, fusion of many source of information become more accessible. Also verification systems are influenced by this progression and tend to use several biometrics instead of one biometric.

Using multimodal arises the question that how to fuse biometrics information. Researches in recent years show that fusion can be done in different levels and the score level fusion is the best in sense of simplicity and amount of information which supposed to be combined. Generally, there are three approaches to score fusion: 1) transformation based score fusion, 2) density based score fusion, 3) classifier based score fusion. Transformation based methods usually are applied after score normalization step. Sum rule, Product rule, Min rule and Max rule belong to this category, amongst them, Sum rule shows the best experimental results [1]. Density based score fusion methods are based on score distribution estimation. Well-known density estimation models like Naïve Bayesian [2] and Gaussian Mixture Model (GMM) [3] have been used for fusion. Classifier based score fusion treat scores as features and try to find the best decision boundary like the case of binary classification binary classification problem [4, 5]. For instance, in [6] Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP) have been used for classifier based score fusion. Multivariate polynomials of hyperbolic functions [7] and its combination with GradientBoost [8] have been applied for score level fusion.

Besides this taxonomy, some algorithms are introduced which try to minimize ranking error and therefore improving Receiver Operating Characteristic (ROC) curve, which is based on maximizing Area under the Curve (AUC) of ROC. In [9], Toh et al. developed least square error based framework to do this, and examined their algorithm for score level fusion. Optimizing AUC in kernel based model was presented in [10], and Freund et al. introduced RankBoost [11] for this purpose. Continuing development of RankBoost, Rudin et al. [12] introduced margin based and coordinate descent of RankBoost. Also, they have proved that AdaBoost not only minimize classification error, but also under certain condition, can optimize AUC. Latest discoveries on AdaBoost capability is inspiring to exploit AdaBoost as a credible algorithm for score fusion. As a consequence, in this paper, we investigate boosting based method not only as a classifier, but also as an algorithm to optimize AUC in multimodal biometrics.

We organized the paper as follows: in Section 2 we bring a brief introduction to AUC criteria. In Section 3, we explain bipartite version of RankBoost and reformulate weak learners score level fusion. Also, we demonstrate the reasons for using boosting methods in score level fusion. In Section 4, we present experimental results over XM2VTS and NIST databases. Finally, Section 5, is dedicated to conclusion.

2. **AUC of ROC curve.** In many cases, in a binary classification problem cost of errors for two classes are not equal, thus, global measures like minimum total error rate does not reflect the real error. As a consequence, changing decision threshold to satisfy imposed cost is inevitable. ROC is a very comprehensive measure, to show how classifiers could deal with this problem. In addition, ROC curve helps to validate classifier performance for undetermined threshold and variable error cost situations [13]. ROC can be understood as a plot of True acceptance rate (TAR) versus false acceptance rate (FAR). In fact, for each possible value of decision threshold, it shows a pair of TAR and FAR values, thus, ROC curve can be determined completely by varying the decision threshold.

To reduce ROC curve to a single scaler value, AUC is used as a measure which can grant our request. AUC is a performance metric that is invariant to unequal error cost and unbalanced class sample size. For example, AUC of base classifier is 0.5 and that of ideal classifier is 1 independent of inequality in error cost and sample size between two classes. AUC can be calculated as follows:

$$AUC = \frac{\sum_{\mathrm{x}_0} \sum_{\mathrm{x}_1} \mathrm{I}(h(\mathrm{x}_1)) > h(\mathrm{x}_0))}{|X_0||X_1|} \tag{1}$$

where $|X_0|$ and $|X_1|$ denote the number of instances for each class in binary classification problem, and $\mathrm{I}(u)$ denotes the indicator function. According to the Equation 1, it can be inferred that misranking error is an affine transform of AUC [12] and therefore minimizing misranking error will increase AUC. Hence minimizing misranking error is the key concept of AUC optimization and until now, wide variety of methods have been developed to train classifiers in order to optimize AUC. In this paper we focus on boosting based algorithms which optimize AUC.

3. **Boosting based score fusion scheme.** As mentioned before, one approach to score level fusion is to exploit classifiers for finding the best decision boundary between genuine and imposter instances. Let $\mathrm{x}^{(n)} = [x_1^{(n)}, x_2^{(n)}, ..., x_N^{(n)}]$ denote the scores of N different biometric matchers for n'th instance, and $y^{(n)} \subseteq -1, 1$ denotes its corresponding label. Note that -1 and 1 refer to imposter and genuine classes, respectively. The goal is to find a decision function like $H : \mathbb{R}^N \to -1, 1$ to recognize labels of unseen instances. Through boosting based algorithms we seek a function that minimize classification error.

In this section, we investigate the idea of using boosting in multimodal biometrics. AdaBoost and bipartite version of RankBoost are those boosting based methods which here will be examined. Up to now AdaBoost as a powerful classifier has not been considered enough in score level fusion problems. There is only one paper in literature that uses AdaBoost to combine palm data with tokenised random numbers [14]. Also, AdaBoost capability in optimizing AUC, proved by Rudin et. al. [12], makes it an appealing method in such area. On the other hand, RankBoost and its bipartite version have been designed to minimize misranking error explicitly [11]. Here, we bring an overview on Bipartite RankBoost, then we explain condition under which AdaBoost optimizes AUC. Also, definitions which are necessary to describe this condition are presented.

3.1. **Bipartite RankBoost.** Freund et al. introduced RankBoost in detail and reported the experimental results of applying this algorithm for meta-searching and movie-recommendation problems [11]. Similar to AdaBoost, this algorithm attempts to find a combination of "weak rankers" to create highly accurate single ranker. Furthermore, they described efficient implementation version of RankBoost, i.e., RankBoost.B, for bipartite feedback. Broadly speaking, the term of bipartite feedback is used when there are two sets of instances and the problem is to rank all instances of one set over another set. Since minimizing ranking error implies optimizing AUC of ROC curve and due to similarity of bipartite feedback and binary classification problems, which are cost-sensitive and has unbalanced classes, this special case of Rankboost algorithm can be applied to such classification problems in which optimization should be done over wide range of decision thresholds.

The pseudocode for Rankboost.B is shown in Figure 1. As it can be inferred from this figure, despite similarity of AdaBoost and RankBoost.B, there are some discrepancies between them. Although final output in two cases is a linear combination of weak learners, training of weak learners and computing of α_t's are accomplished in different ways.

Algorithm **RankBoost.B**

Given: disjoint subset X_0 and X_1 of X.

Initialize:

$$v_1(\mathrm{x}) = \begin{cases} 1/|X_1| & \text{if} \quad \mathrm{x} \in X_1 \\ 1/|X_0| & \text{if} \quad \mathrm{x} \in X_0 \end{cases}$$

For t=1, ..., T:

- Train weak learner using distribution $D_t = v_t(\mathrm{x}_0)v_t(\mathrm{x}_1)$
- Get weak ranking $h_t : X \in \mathbb{R}$
- Choose $\alpha_t \in \mathbb{R}$
- Update:

$$v_{t+1}(\mathrm{x}) = \begin{cases} \frac{v_t(\mathrm{x})\exp(-\alpha_t(\mathrm{x})}{Z_t^1} & \text{if} \quad \mathrm{x} \in X_1 \\ \frac{v_t(\mathrm{x})\exp(\alpha_t(\mathrm{x})}{Z_t^0} & \text{if} \quad \mathrm{x} \in X_0 \end{cases}$$

Where Z_t^1 and Z_t^0 normalize v_t over X_1 and X_0:

$$Z_t^0 = \sum_{\mathrm{x} \in X_0} v_t(\mathrm{x})\exp(-\alpha_t h_t(\mathrm{x}))$$

$$Z_t^1 = \sum_{\mathrm{x} \in X_1} v_t(\mathrm{x})\exp(\alpha_t h_t(\mathrm{x}))$$

Output the final ranking: $H(\mathrm{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathrm{x})$.

Figure 1. Pseudocode of RankBoost for bipartite feedback.

3.2. **Computing α.** In our experiments we use a weak learner training algorithm which generate weak learner h_t with output $\{0,1\}$. Thus according to [11], there exist two methods to compute α_t which are applicable here. Before presenting these methods it should be noticed that an upper bound for ranking error can be obtained as follows:

$$rloss_{\mathrm{D}}(H) \leq \Pi_{t=1}^{T} Z_t$$

where:

$$Z_t = Z_t^0 Z_t^1$$

where Z_t^0 and Z_t^1 normalize instance weights over X_1 and X_0 respectively. Both of methods for computing α are based on minimizing Z_t or an approximation of it. In addition, these methods guarantee that $Z_t \leq 1$.

First method : in this case, $h(\mathrm{x})$ has the range of $0, 1$. For $b \in \{-1, 0, 1\}$, let:

$$W_b = \sum_{\mathrm{x}_0,\mathrm{x}_1} D_t(\mathrm{x}_0\mathrm{x}_1)\mathrm{I}(h_t(\mathrm{x}_0) - \ h_t(\mathrm{x}_1) = b) \tag{2}$$

Dropping subscript t and replacing W_{-1}, W_{+1} with W_- and W_+, respectively, we have:

$$Z = W_0 + W_- e^{-\alpha} + W_+ e^{\alpha} \tag{3}$$

by taking derivative of Z with respect to α and set it to zero, value of α can be computed as:

$$\alpha = \frac{1}{2}\ln(\frac{W_-}{W_+}) \tag{4}$$

which means that:

$$Z = W_0 + 2\sqrt{W_- W_+} \tag{5}$$

To determine the weak learner, we should minimize Z in Equation (5), and set the value of α using Equation (4).

Second method: unlike the first method, here, $h(x)$ takes continuous values in range [0,1]. Also, it should be considered that this method can be used when $h(x)$ takes one of the discrete values of 0 or 1. In [11], it has been shown that:

$$Z \leq \left(\frac{1-r}{2}\right)e^{\alpha} + \left(\frac{1+r}{2}\right)e^{-\alpha} \tag{6}$$

where:

$$r = \sum_{\mathrm{x}_0,\mathrm{x}_1} D(\mathrm{x}_0, 1)(h(\mathrm{x}_0 - h(\mathrm{x}_1)) \tag{7}$$

In bipartite version, r can be rewritten as follows:

$$r = \sum_{x} d(\mathrm{x})s(\mathrm{x})h(\mathrm{x}) \tag{8}$$

where:

$$s(\mathrm{x}) = \begin{cases} +1 & \text{if} \quad \mathrm{x} \in X_1 \\ -1 & \text{if} \quad \mathrm{x} \in X_0 \end{cases}$$

and:

$$d(\mathrm{x}) = v(\mathrm{x}) \sum_{x':s(\mathrm{x}) \neq (\mathrm{x}')} v(\mathrm{x}') \tag{9}$$

The upper bound for Z in Equation (6) is minimized when:

$$\alpha = \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right) \tag{10}$$

It can be revealed that calculating α according to Equation (10), results that $Z < \sqrt{1-r^2}$. Also, to minimize Z, it is adequate to maximize $|r|$ as defined in Equation (7) (Equation(8) in Bipartite case) and then set α according to Equation (10). It should be noticed that having $W_- < W_+$, in the first method, and having negative value of r, in the second method implies negative values for α. This yields that decision boundary of weak learner and final learner are negatively correlated. However, we prevent h_t with negative α's, since, as explained in [11], it will cause poor generalization performance of the final classifier.

3.3. **weak learner.** In this section, we describe weak learners and their training algorithms and reformulate weak learner algorithms in the case of scores level fusion. Similar to boosting based algorithm, Rankboost also includes training weak learner subroutine with slight difference. In RankBoost, weak learner gives weak ranking instead of weak classification. Also, in contrast with the previous ranking application,i.e.,meta-searching and movie-recommendation [11], in multimodal biometrics scores are meaningful, and weak learner always has value for each instance. Thus, $h(x)$ is defined as follows

$$h(\mathrm{x}) = \begin{cases} 1 & \text{if} \quad w.x_i - \theta > 0 \\ 0 & \text{if} \quad w.x_i - \theta \leq 0 \end{cases}$$

where $w \in \{-1, 1\}$. As have been seen, in RankBoost algorithm, how to compute α and what range $h(\mathrm{x})$ has, determines how to train weak learner. Thus, according to this definition of $h(x)$, we rewrite weak learner training algorithm. Figure 2 shows pseudocode of weak learner training algorithm based on defined $h(\mathrm{x})$ and second method of computing α. For abbreviation we neglect writing the pseudocode of the algorithm based on first method of computing α. In second method, calculation of r is very costly. Although in bipartite case computations are reduced, we find out that r can be calculated in simpler fashion with that of Equation(8). Since for $c \in \{0, 1\}$:

$$\sum_{\mathrm{x}_c} v(\mathrm{x}_c) = 1$$

then Equation (9) is reduced to:

$$d(\mathrm{x}) = v(\mathrm{x})$$

consequently we can rewrite r as follows:

$$\begin{aligned} r &= \sum_{x:w.x_i>\theta} h(\mathrm{x})v(\mathrm{x})s(\mathrm{x}) + \sum_{x:w.x_i\leq\theta} h(\mathrm{x})v(\mathrm{x})s(\mathrm{x}) \\ &= \sum_{x:w.x_i>\theta} v(\mathrm{x})s(\mathrm{x}) \end{aligned} \tag{11}$$

3.4. **AdaBoost optimization.** In [12], it has been shown that under specific conditions, AdaBoost will optimize AUC as Bipartite RankBoost does. Rudin et al. showed that if F-skew takes zero value, any sequence of α_t which minimize Adaboost's objective, also minimize Rankboost's objective, consequently, Adaboost and Rankboost achieve the same AUC. In a binary classification problem "Skew"is a quantity which measures the unbalance between positive and negative instances. "F-Skew" shows difference between contribution of positive instances and negative instances in AdaBoost's objective function. Zero value of F-Skew means equally contribution of both classes. Moreover, they proved that if the constant weak hypothesis $h_0(\mathrm{x}) = 1$ is included in the set of AdaBoost's weak classifiers, then $\lim_t \rightarrow \infty$ F-skew $= 0$.

In fact, this condition is equal to weighting instances in each class inversely proportional to the number of instances in the corresponding classes which has been used in unbalanced data classification problems. In the following, we show this equivalence.

We introduce $H_w(\mathrm{x})$ as the final classifier which is trained using weighted instances and $H_c(\mathrm{x})$ as the final classifier which has the constant weak hypothesis $h_0(\mathrm{x}) = 1$ in its set of weak classifiers. If we set the first weak learner as follows:

$$h_0(\mathrm{x}) = 1$$

Algorithm **WeakLearn.B**
Given: distribution over $X \times X$.
set of scores $\{x_i\}_{i=1}^N$
for each score x_i, set of candidate thresholds $\{\theta_j\}_{j=1}^J$ such that $\theta_1 \geq ... \geq \theta_J$
Initialize:
$r^* = 0$
$\pi(\mathbf{x}) = s(\mathbf{x})v(\mathbf{x})$
$w_1 = 1, w_2 = -1$
$For \quad i = 1, ..., N:$
1. $For \quad k = 1, 2:$
2. $L = 0$
3. $\theta_0 = \infty$
4. $For \quad j = 1, ..., J:$
5. $L = L + \sum_{\mathbf{x}:\theta_{j-1} \geq w_k x_i > \theta_j} \pi(\mathbf{x})$
6. $if \quad L > r^*$
7. $r^* = L$
8. $i^* = i$
9. $\theta^* = \theta_j$
10. $w^* = w_i$
Output weak ranking (x_{i^*}, θ^*, w^*)

FIGURE 2. The weak learner.

then the weighted training error of h_0 is:

$$\epsilon_0 = \frac{|X_0|}{|X_0| + |X_1|}$$

and α_0 can be computed as follows:

$$\alpha_0 = \frac{1}{2}\ln\left(\frac{1-\epsilon_0}{\epsilon_0}\right) = \frac{1}{2}\ln\left(\frac{|X_1|}{|X_0|}\right) \tag{12}$$

From AdaBoost algorithm we know that:

$$D_0(n) = \frac{1}{|X_0| + |X_1|} \tag{13}$$

$$D_1(n) = \frac{D_0(n)\exp(-\alpha_0 y^{(n)} h_0(\mathrm{x}^{(n)}))}{Z_0} \tag{14}$$

$$Z_0 = \sum_{\mathrm{x}} D_0(n)\exp(-\alpha_0 y^{(n)} h_0(\mathrm{x}^{(n)})) \tag{15}$$

plugging Equations (12), (13) and (15) into (14), we obtain D1(n) as follows:

$$D_1(n) = \begin{cases} 1/2|X_0| & \text{if} \quad \mathrm{x} \in X_0 \\ 1/2|X_1| & \text{if} \quad \mathrm{x} \in X_1 \end{cases}$$

It states that:

$$H_c(\mathrm{x}) = \alpha_0 + H_w(\mathrm{x})$$

It can be easily seen that these two classifiers have identical ROC curve. Although adding a constant value to the output of the classifier modifies decision threshold, it does not alter ROC curve.

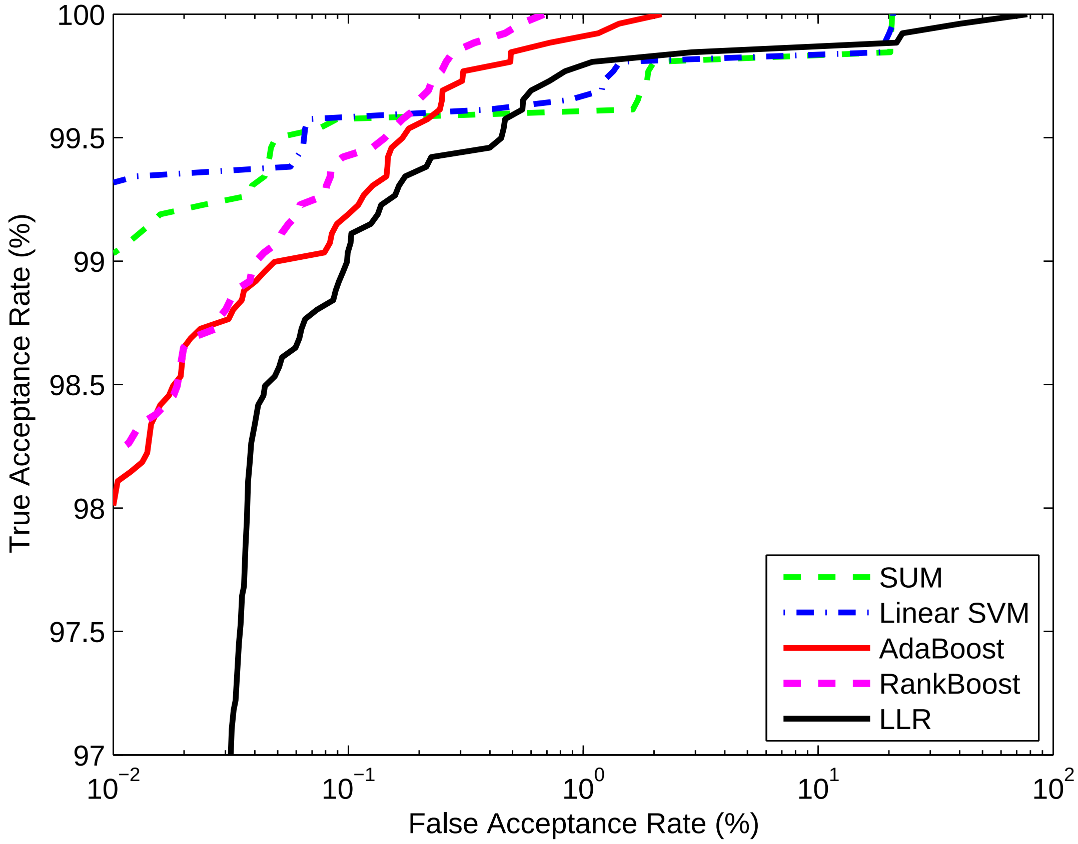

FIGURE 3. ROC curves of AdaBoost based fusion, LLR based fusion, SUM based fusion, and SVM with linear kernel over NIST database.

TABLE 1. HTER of different fusion techniques over XM2VTS and NIST databases.

	SUM	Linear SVM	AdaBoost	RankBoost	LLR
XM2VTS	1.6335	**1.1258**	1.5615	1.3252	2.2536
NIST	0.4646	0.4826	**0.4099**	0.4551	0.4580

4. **Experimental results.** In order to evaluate performance of boosting-based fusion methods, experimental tests over different databases have been run and different fusion techniques have been used. XM2VTS-Benchmark database [15] for score level fusion, and fing-face part of NIST-bssr1 [16] were used. In XM2VTS training and test set partitioning is already defined and there is no need to do partitioning randomly. In case of NIST partitioning randomly for ten times. Cross-validation algorithm with ten folds is run over 32 part of XM2VTS and each partition of NIST. According to mentioned taxonomy for score level fusion techniques, we selected benchmark methods from each category. From transformation based methods, SUM rule with min-max normalization, from density based methods, GMM, and from classifier based methods, SVM were selected. To compare AdaBoost with other classifier based methods, we evaluated SVM. We used linear SVM because in our experiments SVM with radial basis function kernel did not perform well as SVM with linear kernel. In this paper we used GMM fitting algorithm proposed in [17] and SVM light [18]. Also we used implemented version of AdaBoost from Statistical Pattern Recognition Toolbox (STPRTool) [19]. To satisfy AUC optimization condition we weighted instances inversely proportional to number of instances in corresponding classes.

The ROC curves of SUM rule, SVM with linear kernel, LLR, AdaBoost and RankBoost over NIST database are depicted in Figure 3. As it can be seen, AdaBoost and RankBoost

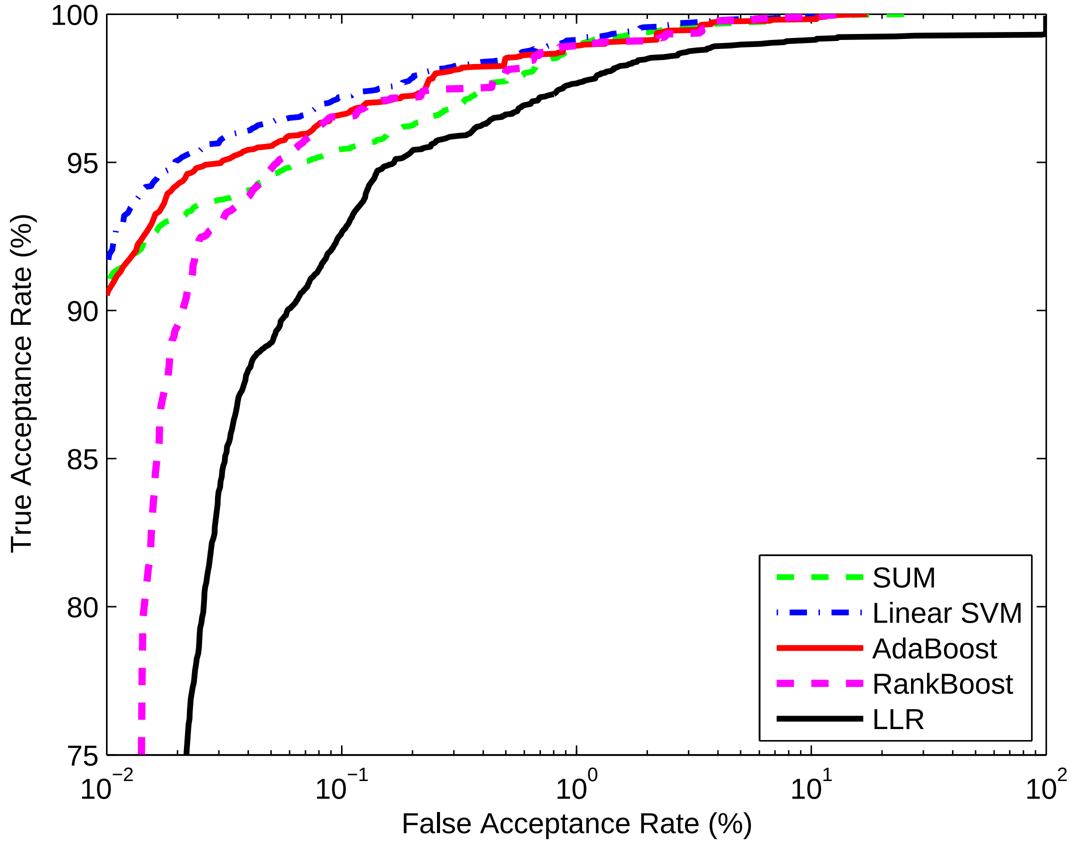

FIGURE 4. ROC curves of Boosting based fusion, LLR based fusion, SUM based fusion, and SVM with linear kernel over XM2VTS database.

reach higher performance compared to LLR. In comparison with SUM rule and Linear SVM in bigger portion of FAR range ROC of boosting approach are above that of SUM and Linear SVM. Furthermore, it can be seen, AdaBoost achieves performance comparable to that of RankBoost. Also results in Table 1 shows that HTER of boosting methods over NIST are lower than the other methods.

In Figure 4 ROC curves over XM2VTS have been depicted. It can be inferred that classifier based methods outperformed SUM and LLR. In this case, boosting approaches have higher performance than LLR and SUM, but they are not superior than Linear SVM. In addition, HTER in Table 1 proves these statements. After Linear SVM which achieves lowest HTER, RankBoost and AdaBoost have lower HTER respectively.

5. **Conclusion.** In this paper, we approved practically the theoretic equality of AdaBoost and Rank- Boost.B in sense of AUC optimization. We applied RankBoost.B, and AdaBoost in multimodal biometrics score level fusion with understanding its ability in AUC optimization. Also, we reformulated weak learner used in RankBoost.B for applications in which scores are meaningful. We explained that weighting instances inversely proportional to the number of instances in the corresponding classes is another statement of the condition under which AdaBoost will optimize AUC. According to experimental results, AdaBoost and RankBoost.B reached higher performance compared to SUM rule, LLR. Finally our experiment showed that boosting approach and SVM with linear kernel have comparable performance. Based on these results we conclude the paper as follows:

- Classifier approach outperforms compared to transformation based score fusion and density based score fusion.
- AdaBoost achieves the same level of performance compared to RankBoost.B

- Performance of SVM is comparable with boosting based algorithm which shows SVM ability as a classifier based method for score level fusion.
- Weighting instances inversely proportional to the number of instances in the corresponding classes is another statement of the condition under which AdaBoost will optimize AUC

Acknowledgment. This work is supported by Iran Telecommunication Research Center, under the project "*New Algorithms for Improving the Performance of Multimodal Biometric Systems.*"

REFERENCES

[1] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226-239, 1998.
[2] N. Poh and S. Bengio, Eer of fixed and trainable fusion classifiers: A theoretical study with application to biometric authentication tasks, *Proc. of Conference on Lecture Notes in Computer Science*, pp. 74-85, 2005.
[3] K. Nandakumar, Y. Chen, S. C. Dass, and A. K. Jain, Likelihood ratio-based biometric score fusion, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 342-347, 2008.
[4] J. Fíerrez-Aguilar, J. Ortega-Garcia, J. Gonzalez-Rodriguez, and J. Bigün, Discriminative multimodal biometric authentication based on quality measures, *Pattern Recognition*, vol. 38, no. 5, pp. 777-779, 2005.
[5] Y. Ma, B. Cukic, and H. Singh, A classification approach to multi-biometric score fusion. *Proc. of Conference on AVBPA* , pp. 484-493, 2005.
[6] S. Bengio, C. Marcel, S. Marcel, and J. Mariethoz, Confidence measures for multimodal identity verification, *Information Fusion*, vol. 3, no. 4, pp. 267-276, 2002.
[7] K. A. Toh, and W. Y. Yau, Combination of hyperbolic functions for multimodal biometrics data fusion, *IEEE Trans. Systems Man and Cybernetics*, vol. 34, no. 2, pp. 1196-1209, 2004.
[8] M. Parviz, and M. S. Moin, Multivariate polynomials estimation based on gradientboost in multimodal biometrics, *Proc. of Conference on ICIC*, vol. 15, no. 3, pp. 471-477, 2008.
[9] K. A. Toh, J. Kim, and S. Lee. Maximizing area under roc curve for biometric scores fusion, *Pattern Recognition*, vol. 41, no. 11, pp. 3373-3392, 2008.
[10] S. Gao and Q. Sun, Improving semantic concept detection through optimizing ranking function, *IEEE Trans. Multimedia*, vol. 9, no. 7, pp. 1430-1442, 2007.
[11] Y. Freund, R. D. Iyer, R. E. Schapire, and Y. Singer, An efficient boosting algorithm for combining preferences, *Journal of Machine Learning Research*, vol. 4, pp. 933-969, 2003.
[12] C. Rudin and R. E. Schapire, Margin-based ranking and an equivalence between adaboost and rankboost, *Accepted for Journal of Machine Learning Research*, 2008.
[13] T. Fawcett, An introduction to roc analysis, *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861-874, 2006.
[14] L. Nanni and A. Lumini, A multi-modal method based on the competitors of FVC2004 and on palm data combined with tokenised random numbers Pattern Recognition Letters, vol. 29, no. 9, pp. 1344-1350, 2008.
[15] N. Poh and S. Bengio, Database, protocols and tools for evaluating score-level fusion algorithms in biometric authentication. *Pattern Recognition*, vol. 39, no. 2, pp. 223-233, 2006.
[16] NIST, Biometric scores set, *http://www.itl.nist.gov/iad/894.03/biometricscores/*.
[17] M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 381-396, 2002.
[18] T. Joachims, Making large-scale support vector machine learning practical,*Advances in kernel methods*, MIT Press., MA, Cambridge, pp. 169-184, 1999.
[19] V. Franc and V. Hlaváč, Statistical pattern recognition toolbox for Matlab, *Research Report CTU-CMP-2004, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic*, 2004.