JBC: Joint Boost Clustering method for synthesis aperture radar images

Article *in* Proceedings of SPIE - The International Society for Optical Engineering · November 2007

DOI: 10.1117/12.749062

5 authors, including:

Chu He
Wuhan University

Xin Xu
Wuhan University

Hong Sun
Wuhan University

JBC: Joint Boost Clustering method for synthesis aperture radar images

Mengling Liu, Chu He, Gui-Song Xia, Xin Xu, Hong Sun
Signal Processing Lab., Electronic Information School, Wuhan University,
Wuhan 430079, P.R.China.

ABSTRACT

A clustering method based on Joint Boost for Synthesis Aperture Radar images is proposed. In this method, we follow the steps of Joint Boost, but substitute weak learns with basic clustering algorithm. We compute the sharing features between samples in order to reduce clustering times. The proposed clustering method, JBC constructs a new training set by random sampling from the original dataset, then selects the best feature and the best clusters for sharing, and calculates a distribution over the training samples using current shared feature and clusters, and finally a basic clustering algorithm (e.g. K-mean) is applied to partition the new training set. The final clustering solution is produced by aggregating the obtained partitions. The clustering results for SAR images show that the proposed method has a good performance.

Keywords: clustering, Joint Boost, Synthesis Aperture Radar(SAR), k-means, multi-classifier

1. INTRODUCTION

Clustering, sometimes called unsupervised classification, is a crucial and interesting topic in data analysis, which classifies dataset using the similarity between clusters without the knowledge of labels of samples. For image data, the partition based clustering methods are usually used, such as k-mean and its variations. However, as dealing with noisy images with low signal noise ratio (SNR), e.g. Synthesis Aperture Radar (SAR) images, the traditional partition based clustering methods can not provide useful and satisfying results effectively. These may attribute to two reasons: one is that the features involved in the clustering processing sometimes are not for noisy data and have much redundancy; the other is that just using one clustering method is not able to give a high clustering precision.

Boosting is a fashionable approach in learning. The most important principle of boosting is combining a set of weak classifiers to form a high-performance prediction rule. Joint Boost is a new boosting approach proposed by Torralba, which not only gives rights to designers to add new weak classifiers endlessly until reaching a low error rate, but also allows designers to remove the redundancy between features by sharing them.

However, unlike classification problems, there are no established methods to combine multiple clusterings. This problem is more difficult than designing a multi-classifier method because in the clustering case there is lack of knowledge concerning the label of the cluster to which a training point actually belongs. Frossyniotis and his collaborators have proposed a boosting based clustering method, which substitutes weak classifiers to simple clustering algorithms (i.e. k-means) and adds weighted clustering result to get final clusters. But it is not effective and most of time can't work on SAR images, since the features of noisy data are not considered validly and are redundancy.

In this paper, A Joint Boost Clustering (JBC) method is proposed for clustering noisy images with low SNR, e.g. SAR Images. The JBC iteratively recycles the training examples providing multiple clusterings and resulting in a common partition. At each iteration, firstly, the JBC selects the best feature and the best clusters for sharing, then calculates a distribution over the training samples using current shared feature and clusters, and constructs a new training set by random sampling from the original dataset, and finally a basic clustering algorithm (e.g. K-mean) is applied to partition the new training set. The final clustering solution is produced by aggregating the obtained partitions.

The JBC method has been designed in following steps. We assume a given image X on a lattice S, a basic clustering algorithm (weak learner) and the required number of clusters C. The maximum number of iterations T of JBC will be considered fixed, although this parameter is meaningless considering stopping criteria. The algorithm is summarized below. (1) Choosing samples from original dataset randomly, initialize sample weights, confirm first distribution of samples. (2) Applying a simple clustering algorithm to calculate cluster centers, computer the distances between samples and centers. (3)Computing error according to these distributions, and selecting the best sharing feature and according clusters in the whole clustering space. (4) Renewing sample weights according to distributions and updating the strong

MIPPR 2007: Pattern Recognition and Computer Vision, edited by S. J. Maybank, Mingyue Ding, F. Wahl, Yaoting Zhu
Proc. of SPIE Vol. 6788, 678814, (2007) · 0277-786X/07/$18 · doi: 10.1117/12.749062

classification. Make sure that the weights of wrong classifications are added and the weights of right classifications are reduced. So we can pay more attention to those samples which have wrong classifications in next iteration. (5) Resamping training set according to the new sample weights randomly and turning to the first step to iterate. Finally, the clustering solution is produced by aggregating the obtained partitions.

The features we used here include Local Binary Pattern (LBP), Scale Invariant Feature Transform (SIFT), GLCM, Gabor filter bank, and other image primitives. We also establish a SAR image database containing more than 1000 images for image clustering, ranging from many kinds of different scenes and objects.

2. BOOSTING, CLUSTERING ON BOOSTING AND JOINT BOOST

A new multiple clustering method is proposed, which we will call Joint Boost-clustering. At each iteration of JBC algorithm, a distribution over the training samples is computed and a new training set is sampled randomly from the original dataset. We compute and select the best features and the best clusters for sharing. Then a basic clustering algorithm is applied to partition the new training set. The final clustering solution is produced by aggregating the obtained partitions.

Firstly, we introduce a clustering method based on boosting. (D.Frossyniotis,2004)

2.1 A clustering method based on boosting

Boosting is a general method for improving the accuracy of any given learning algorithm[1]. Boosting finds many rough rules of thumb. It can be a lot easier than finding a single, highly accurate prediction rule. Boosting algorithm calls these rough rules "weak learners". The final classification result is given by combining these weak rules into a single classification rule which will be more precise than any one of the "weak learners".

A clustering method based on boosting, which was called *boost-clustering*[2]. At each iteration of this algorithm, a distribution over the training samples is calculated and a new training set is set up using random sampling from the original dataset. Then a basic clustering algorithm is called to classify the new training set. The final clustering solution is obtained by aggregating the clustering results using weighted voting, where the weight of each classification is a measurement of its quality.

According to the idea of *boost-clustering*, we introduce a new clustering method similar with *boost-clustering*, but Joint boost-clustering follows the steps of the Joint boost algorithm for classification. JBC selects features for sharing to reduce the total number of clustering.

2.2 The Joint Boost algorithm

Joint Boost was proposed to share weak-learners across classes. For instance, if we have 3 classes, we might define the following classifiers:

$$H(v,1) = D^{1,2,3}(v) + D^{1,2}(v) + D^{1,3}(v) + D^{1}(v)$$
$$H(v,2) = D^{1,2,3}(v) + D^{1,2}(v) + D^{2,3}(v) + D^{2}(v)$$
$$H(v,3) = D^{1,2,3}(v) + D^{1,3}(v) + D^{2,3}(v) + D^{3}(v)$$

(1)

where each $D^{S(n)}(v)$ is itself an additive model of the form $D^{S(n)}(v) = \sum_{m=1}^{Mn} h_m^n(v)$ .The n refers to a node in the "sharing graph" (see Figure 1), which specifies which functions can be shared between classifiers.

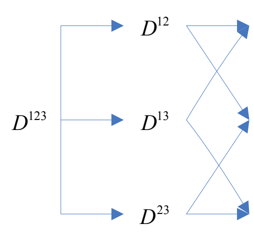


Fig.1. All possible ways to share features amongst 3 classifiers.

The idea of Joint Boost algorithm is that at each boosting round, we examine various subsets of classes, and considering fitting a weak classifier to distinguish that subset from the background. We pick sharing features between samples can help to reduce the number of weak learners. Joint Boost chooses sharing features by computing an error:

$$J_{wse}(n)=\sum_{c=1}^{C}\sum_{i=1}^{N}\omega_i^c(z_i^c-h_m(v_i,c))^2 \qquad (2)$$

We choose the first class which has the least error. Then we select the second class which has the least error jointly with the first class. We repeat until adding all the classes, then pick the sharing that has the largest error reduction.

The flow cart of Joint Boost algorithm is summarized below.

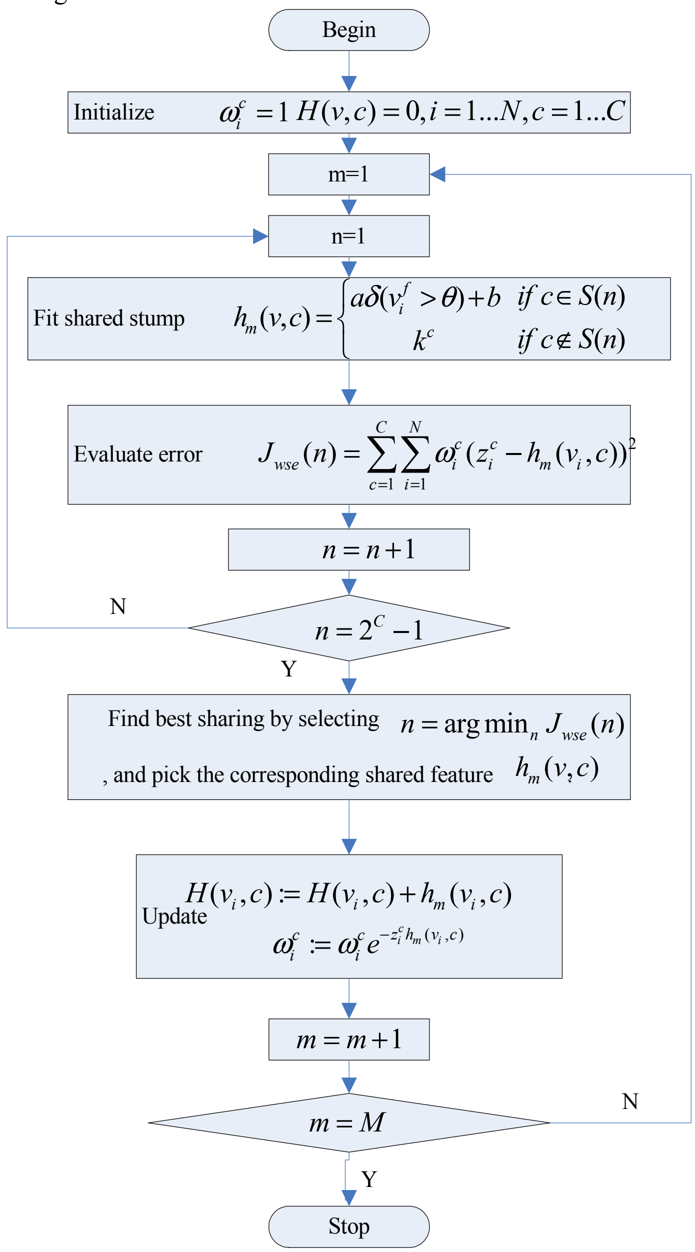


Fig.2. The flow cart of Joint Boost with regression stumps. v_i^f is the f'th feature of the i'th training example, $z_i^c\in\{-1,+1\}$ are the labels for class c, and ω_i^c are example weights. N is the number of training examples, and M is the number of rounds of Joint Boost.

3. THE JOINT BOOST-CLUSTERING ALGORITHM

3.1 Joint Boost-clustering Algorithm.

A new multiple clustering method is proposed, which we will call Joint Boost-clustering. At each iteration of JBC algorithm, a distribution over the training samples is computed and a new training set is sampled randomly from the original dataset. We compute and select the best features and the best clusters for sharing. Then a basic clustering algorithm is applied to partition the new training set. The final clustering solution is produced by aggregating the obtained partitions. The Joint Boost-clustering Algorithm is summarized below.

Given: Input sequence of N instances $(x_1, ..., x_N)$, $x_i \in R^d$, $i = 1, ..., N$,a *basic clustering algorithm*, the number C of clusters to partition the data set and the maximum number of iterations T.

1). Initialize $w_i^c = 1$ for $i = 1, ..., N$ .Set $H(v_i, c) = 0, i = 1, ..., N, c = 1, ..., C$.

2). Repeat while $t \le T$

a) Repeat while $n \le 2^C - 1$

i)Call the basic clustering algorithm, to get the clustering centers.

ii)Get the sharing stump

$$h_t(v, c) = \frac{1}{\sum_{k=1}^{C} \frac{d(x_i, \mu_j)}{d(x_i, \mu_k)}} , \tag{3}$$

where $\mu_j \in R^d$ corresponds to a cluster center.

iii)Calculate the error:

$$J_{wse}(n) = \sum_{c=1}^{C} \sum_{i=1}^{N} \omega_i^c (z_i^c - h_t(v_i, c))^2 \tag{4}$$

b) pick the best sharing by selecting $n = \arg\min_n J_{wse}(n)$,choose the corresponding shared feature $h_t(v, c)$

c) Update distribution W :

$$\omega_i^c = \omega_i^c e^{-z_i^c h_t(v_i, c)} \tag{5}$$

d) Update strong classification: $H(v_i, c) := H(v_i, c) + h_t(v_i, c)$ (6)

3). Output: the final clustering hypothesis H .

3.2 The details of JointBoost clustering algorithm

It is clearly that the method has been developed following the steps of the JointBoost algorithm for classification. We assume a given set X of N dimensional instances x_i , a basic clustering algorithm(weak learner) and the required number of cluster C. The maximum number of iterations T of JointBoost-clustering will be considered fixed, although this parameter is meaningless considering the stopping criteria. Choosing sharing features can reduce the total number of weak learners and meanwhile speed up computational rate.

At each iteration $t = 1, ..., T$, first a dataset X^t is sampled from X using weight distribution $W^t = \{\omega_i^t\}$, then we have to decide which classes are going to share a feature. We begin with computing all the features for the leaves (single samples). We select first the class that has the best reduction of the error. Then we select the second class that has the best error reduction jointly with the previously selected class. We repeat until we have added all the classes. We select the sharing that has the largest error reduction. Then a basic clustering algorithm is called to get clustering centers for all sample x_i , $i = 1, ..., N$. Following the steps, we continue to compute $h_t(v, c)$ which denote the membership between samples and centers. Then calculate the error and choose the shared feature. Stopping at this step, we successfully obtain the sharing feature. Update the weight for next round and renew the strong classification. The final hypothesis is get by adding all the sharing feature $h_t(v, c)$ which have the largest error reduction.

Using this method, we reduce the weight of a well-clustered data point and put more attention on badly clustered data points. Another advantage of JBC is the reduction of the clustering times so that we can have a more rapid clustering program and a comparative clustering result.

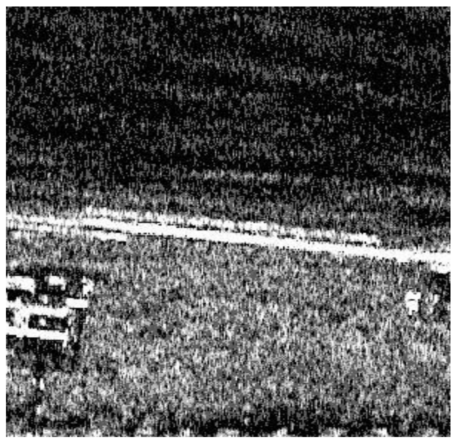
4. EXPERIMENTAL RESULTS

In studying the JBC method, we considered k-means clustering algorithm. In our implementation, the membership degree $h_{i,j}$ for every sample x_i to cluster j is produced based on Euclidean distance d

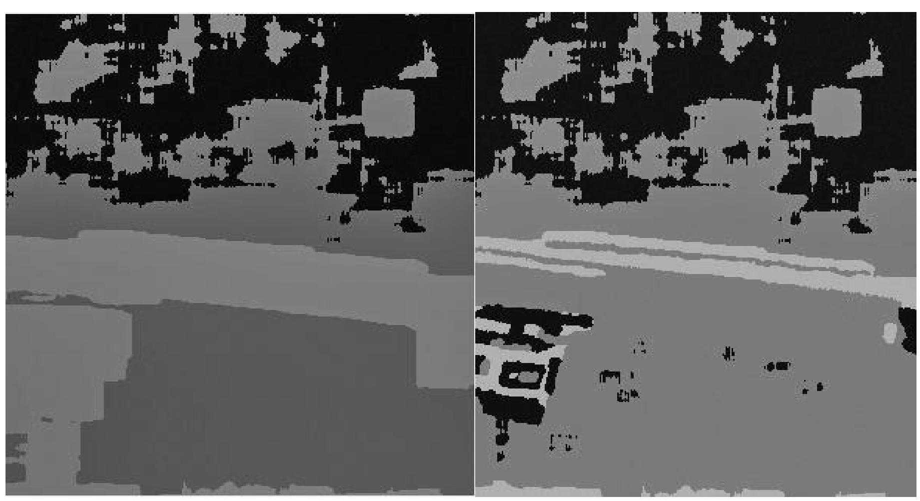
$$h_t(v,c) = \frac{1}{\sum_{k=1}^{C} \frac{d(x_i,\mu_j)}{d(x_i,\mu_k)}} \quad \text{, where } \mu_j \in R^d \text{ corresponds to a cluster center.} \tag{7}$$

In order to demonstrate the performance of the JBC algorithm, the features we used here include Local Binary Pattern (LBP), Scale Invariant Feature Transform (SIFT), GLCM, Gabor filter bank, and other image primitives. We also establish a SAR image database containing more than 1000 images for image clustering, ranging from many kinds of different scenes and objects.

Because of the length of article, we only show some results of clustering for two SAR images. Clustering results using k-means method are also displayed to be compared. The first SAR image used here has 300*350 pixels. We want to classify it into 3 classifications without any label information. The second SAR image used here has 1000*350 pixels. We want to classify it into 5 classifications. Each sample has 143 features including LBP, SIFT, GLCM and Gabor filter bank. The sample set is a X*143 matrix. The clustering results of JBC method and k-means method are showing below.



(a)



(b) (c)

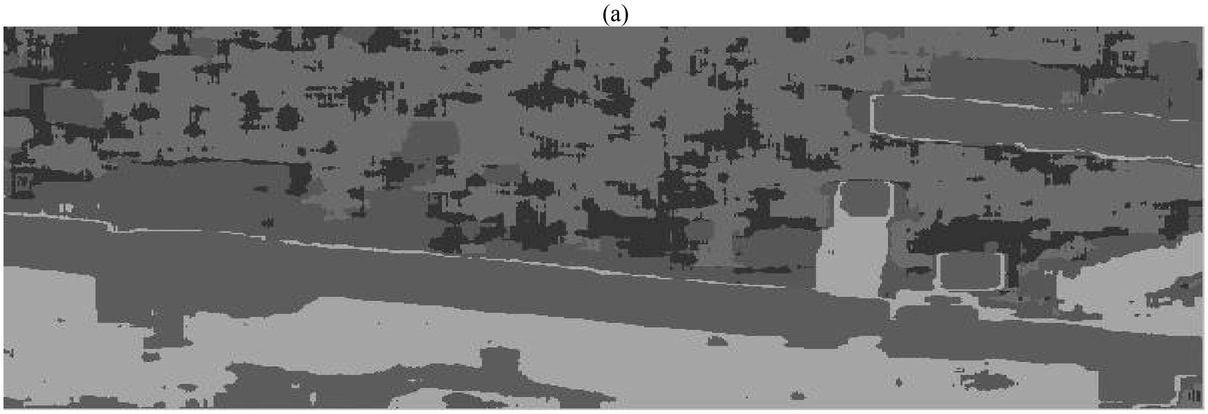
Fig3. The clutering results comparion between JBC and K-means.

(a) original figure

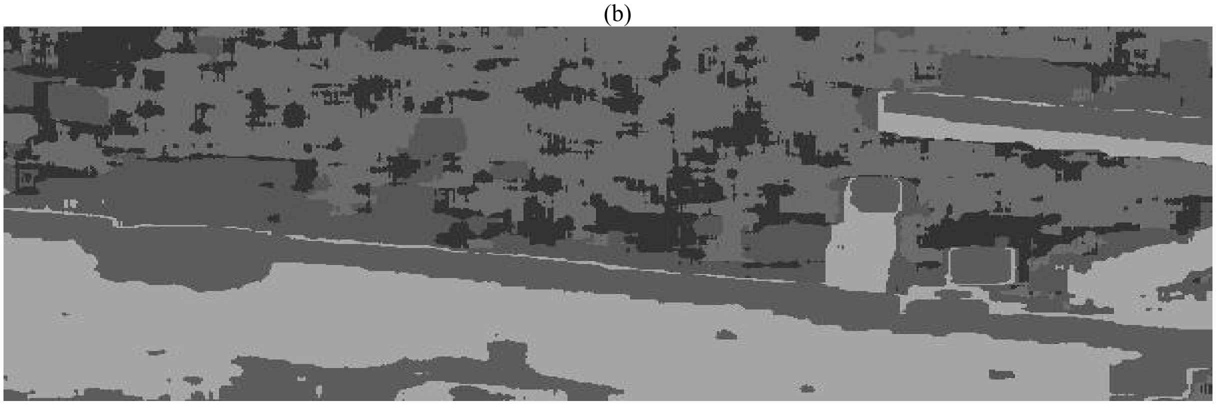
(b) k-means used on LBP, SIFT, GLCM and Gabor filter bank with 3 class

(c) JBC used on LBP, SIFT, GLCM and Gabor filter bank with 3 class

(a)



(b)



(c)

Fig4. Another clutering results comparion between JBC and K-means.
(a) original figure
(b) k-means used on LBP, SIFT, GLCM and Gabor filter bank with 5 classes
(c) JBC used on LBP, SIFT, GLCM and Gabor filter bank with 5 classes

From the above results of JBC and K-means methods which are used on same features (see Fig3 and Fig4), we can see JBC method performances well than k-means on same features of the same scene. Also, It demonstrates that JBC method can deal with SAR images which have low SNR. But the result of JBC method is not precise enough, further study of improving the JBC algorithm is still needed in the future.

5. CONCLUSIONS

In this work a new clustering methodology has been introduced based on the principle of JointBoost. The proposed methodology is a multiple clustering method based on the iterative application of a basic clustering algorithm. At each iteration, firstly, the JBC t selects the best feature and the best clusters for sharing, then calculates a distribution over the training samples using current shared feature and clusters, and constructs a new training set by random sampling from the original dataset, and finally a basic clustering algorithm (e.g. K-mean) is applied to partition the new training set. The final clustering solution is produced by aggregating the obtained partitions with sharing features.

Considering the iterative number T is meaningless, further study of effective stop criterions is needed in the future.

Acknowledgment

The work is supported by the National Nature Science Foundation of China under projects No.60372057 and No.4037605. The authors are also grateful to "Open Research Fund of State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing 2007".

References

1. Robert E. Schapire. The boosting approach to machine learning: An overview. In D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, B. Yu, editors, *Nonlinear Estimation and Classification*. Springer, 2003.
2. D.Frossyniotis. A clustering method based on boosting. Pattern Recognition Letters, 2003.
3. Antonio Torralba, Kevin p. Murphy and William T.Freeman. Sharing Visual Features for Multiclass and Multiview Object Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 5, pp. 854-869, May, 2007.
4. Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms:Bagging, boosting, and variants. *Machine Learning*, 36(1/2):105–139, 1999.
5. Robert E. Schapire. A brief introduction to boosting. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence,1999.
6. A. Torralba, K. P. Murphy and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection.Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp 762-769, 2004.
7. Ron Meir and Gunnar Rätsch. An introduction to boosting and leveraging. In Advanced Lectures on Machine Learning (LNAI2600), 2003.
8. Yoav Freund and Robert E. Schapire. A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, September, 1999.
9. Robert E. Schapire. Theoretical views of boosting. In Computational Learning Theory: Fourth European *Conference,* EuroCOLT'99, pages 1-10, 1999.
10. Robert E. Schapire. Theoretical views of boosting and applications. In Tenth International Conference on Algorithmic Learning Theory, 1999.
11. Lev Reyzin and Robert E. Schapire. How boosting the margin can also boost classifier complexity. In Proceedings of the 23rd International Conference on Machine Learning, 2006.
12. Aurélie C. Lozano, Sanjeev R. Kulkarni and Robert E. Schapire. Convergence and consistency of regularized boosting algorithms with stationary beta-mixing observations. In Advances in Neural Information Processing Systems 18, 2006.
13. Cynthia Rudin, Corinna Cortes, Mehryar Mohri and Robert E. Schapire. Margin-based ranking meets boosting in the middle. In 18th Annual Conference on Computational Learning Theory, 2005.
14. Robert E. Schapire, Marie Rochery, Mazin Rahim and Narendra Gupta. Boosting with prior knowledge for call classification. IEEE Transactions on Speech and Audio Processing, 13(2), March, 2005.
15. Robert E. Schapire. Advances in boosting. In Uncertainty in Artificial Intelligence: Proceedings of the Eighteenth Conference, 2002.

115