BOOSTING FOR CONTENT-BASED AUDIO CLASSIFICATION AND RETRIEVAL: AN EVALUATION

Guodong Guo, Hong-Jiang Zhang, and Stan Z. Li

Microsoft Research China
3F, Beijing Sigma Center, No. 49, Zhichun Road, Haidian District
Beijing 100080, P.R.China

ABSTRACT

In this paper, we evaluate a recently proposed algorithm in machine learning called AdaBoost for content-based audio classification and retrieval. AdaBoost is a kind of large margin classifiers and is efficient for on-line learning. Our focus is to evaluate its classification and retrieval accuracy as compared with other methods. The Muscle Fish audio database of 409 sounds is used for the evaluation with perceptual and cepstral features.

1. INTRODUCTION

Audio data is an integral part of many modern computer and multimedia applications. Numerous audio recordings are dealt with in audio and multimedia applications. The effectiveness of their deployment is greatly dependent on the ability to classify and retrieve the audio files in terms of their sound properties or content. Rapid increase in the amount of audio data demands for a computerized method which allows efficient and automated content-based classification and retrieval of audio database.

Wold *etal* [17] have developed a system called "Muscle Fish". That work distinguishes itself from earlier work [4, 5, 6] in its content-based capability. There, various perceptual features are used to represent a sound. A normalized Euclidean (Mahalanobis) distance and the *nearest neighbor* (NN) rule are used to classify the query sound into one of the sound classes in the database. In Liu *etal* [10], separability of different classes is evaluated in terms of the intra- and inter-class scatters to identify highly correlated features. Foote [2] choose to use 12 mel-frequency cepstral coefficients (MFCCs) as the audio features. Histograms of sounds are compared and the classification is done by using the NN rule. In Pfeiffer *etal* [12], audio features are extracted by using gammaphone filters. Li [8] used the nearest feature line (NFL) method for content-based audio classification and retrieval. Li and Guo [9] proposed to use the support vector machines (SVMs) to classify and retrieve audio patterns. The SVM minimizes the structural risk, that is, the probability of misclassifying yet-to-be-seen patterns for a fixed but unknown probability distribution of the data. This is in contrast to traditional pattern recognition techniques of minimizing the empirical risk, that is, optimizing the performance on the training data. This minimum structural risk principle is equivalent to minimizing an upper bound on the generalization error.

Boosting [7] also tries to maximize the margin between positive and negative examples, in which ensemble of classifiers are trained sequentially. In each subsequent problem, examples are reweighted in order to emphasize the incorrectly classification by previous weak classifier. The final decision is a weighted combination of the weak classifiers. We use the simple nearest center (NC) classifier as the weak learner. The classification and retrieval performance is compared with the SVM based approach on the Muscle Fish database.

2. AUDIO FEATURE EXTRACTION

Before feature extraction, an audio signal (8-bit ISDN μ-law encoding) is preemphasized with parameter 0.96 and then divided into frames. Given the sampling frequency of 8000 Hz, the frames are of 256 samples (32ms) each, with 25% (64 samples or 8ms) overlap in each of the two adjacent frames. A frame is hamming-windowed by $w_i = 0.54 - 0.46 * \cos(2\pi i/256)$. It is marked as a silent frame if $\sum_{i=1}^{256}(w_i s_i)^2 < 400^2$ where s_i is the preemphasized signal magnitude at i and 400^2 is an empirical threshold.

Two types of features are computed from each frame: (i) perceptual features, composed of total power, sub-band powers, brightness, bandwidth and pitch; and (ii) mel-frequency cepstral coefficients (MFCCs). Then audio features are extracted from each non-silent frame. The means and standard deviations of the feature trajectories over all the non-silent frames are computed, and these statistics are considered as feature sets for the audio sound.

A 18-dimensional perceptual feature vector named "perc" is extracted, and normalized to form the final feature set named "Perc". The means and standard deviations of the

L MFCCs are calculated over the non-silent frames, giving a $2L$-dimensional cepstral feature vector, named "Ceps$L$". In the experiments, Ceps$L$ with L values in the range between 5 and 120, with the corresponding feature sets named Ceps5, $\cdots$, Ceps120, are evaluated. Further more, the Perc and Ceps$L$ feature sets are weighted and then concatenated into still another feature set, named "PercCeps$L$", of dimension $18 + 2L$, giving PercCeps5, $\cdots$, PercCeps120. See [8] for detailed definitions of these features.

3. ADABOOST

Boosting is a method to combine a collection of weak classification functions (weak learner) to form a stronger classifier. AdaBoost is an adaptive algorithm to boost a sequence of classifiers, in that the weights are updated dynamically according to the errors in previous learning [7]. AdaBoost is a kind of large margin classifiers.

Tieu and Viola [15] adapted the AdaBoost algorithm for natural image retrieval. They made the weak learner work in a single feature each time. So after T rounds of boosting, T features are selected together with the T weak classifiers.

We evaluate the AdaBoost algorithm of Tieu and Viola's version for content-based audio classification and retrieval, which can simultaneous select a small number of relevant features in the learning process. For each pair of audio classes, the AdaBoost is used to run for T rounds. When the distances to the mean values are used in each dimension [15], the weak learner is simple, *i.e.,* x is classified to class 1 if $|x - \mu_1| < |x - \mu_2|$.

AdaBoost Algorithm

Input: 1) n training examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ with $y_i = 1$ or 0; 2) the number of iterations T.

Initialize weights $w_{1,i} = \frac{1}{2l}$ or $\frac{1}{2m}$ for $y_i = 1$ or 0, with $l + m = n$.

Do for $t = 1, \ldots, T$:

1. Train one hypothesis h_j for each feature j with w_t, and error $\epsilon_j = Pr_i^{w_t}[h_j(x_i) \neq y_i]$.

2. Choose $h_t(\cdot) = h_k(\cdot)$ such that $\forall j \neq k, \epsilon_k < \epsilon_j$. Let $\epsilon_t = \epsilon_k$.

3. Update: $w_{t+1,i} = w_{t,i}\beta_t^{e_i}$, where $e_i = 1$ or 0 for example x_i classified correctly or incorrectly respectively, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

4. Normalize the weights so that they are a distribution, $w_{t+1,i} \longleftarrow \frac{w_{t+1,i}}{\sum_{j=1}^{n} w_{t+1,j}}$.

Output the final hypothesis,

$$h_f(x) = \begin{cases} 1 & if \ \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases} \quad (1)$$

where $\alpha_t = \log \frac{1}{\beta_t}$.

Above AdaBoost algorithm is only used for two-class classification. In a multi-class scenario, we use a majority voting strategy to combine all pair-wise classification results to get the final decision. Because the learning and classification process of AdaBoost is very fast, it does not cost much time for the combination of 120 pairs of classification (for 16 classes).

For retrieval, the samples of one class are considered as positive, while other examples in the training set are taken as negative, thus to learn 16 decision boundaries. In testing, when a query audio is given, one of the decision boundaries is found firstly, which is based on the largest distance to the 16 decision boundaries. Then all other audio patterns are ranked with respect to the selected boundary based on their signed distances to it.

4. SUPPORT VECTOR MACHINE

Support vector machine (SVM) learns an optimal separating hyperplane (OSH) given a set of positive and negative examples. Kernel functions are used for SVM to learn a non-linear boundary if necessary. See Vapnik [16] for a detailed introduction of SVM. Li and Guo [9] tried to use the SVM for audio classification and retrieval. In classification, a binary tree is used to tack the multi-class classification problem. Because the testing process of SVM is a little time consuming, the binary tree strategy can reduce the number of pairwise comparisons. Only $(n-1)$ comparisons for each query, where n is the number of classes.

For retrieval, the SVMs learn n decision boundaries in the training stage. In testing, one of the decision boundaries is found firstly to the query, and is used to rank other audio patterns in the database. See [9] for a detailed description. Here the SVM based methods are used in comparison with the AdaBoost approach for classification and retrieval.

5. EXPERIMENTS

An audio database of 409 sounds from Muscle Fish is used for the experiments, which is classified into 16 classes by Muscle Fish. The database can be obtained from **http://www.musclefish.com/cbrdemo.html**, and has been used in [17] [8] [9]. The names of the audio classes are altotrombone (13), animals (9), bells (7), cellobowed (47), crowds (4), female (35), laughter (7), machines (11), male (17), oboe (32), percussion (99), telephone (17), tubularbells (19), violinbowed (45), violinpizz (40), water (7). The numbers indicate how many samples in each class. The samples are of different length, ranging from one second to about ten seconds. To evaluate the classification performance, we calculate the *error rate*, which is defined as the ratio between the number of mis-classified examples and

the total number of testing examples. For retrieval performance evaluation, we compute the *average retrieval accuracy*, which has been used as a performance measure for texture image retrieval [11]. It is defined as the average percentage number of patterns belonging to the same class as the query in the top n matches.

The 409 sounds are partitioned into a training set of 211 sounds and a test set of 198 sounds, as that in [8]. The partition is done in the following way: (1) sort the sounds in each class in the alphabetical order of the file names, and then (2) construct the two sets by including sounds 1, 3, $\cdots$ in the prototype set and sounds 2, 4, $\cdots$ in the test set.

Three feature sets, Perc, CepsL and PercCepsL are used for the evaluation of the AdaBoost algorithm for audio classification. To demonstrate the boosting behavior, we run the AdaBoost step by step with $T = 1, 2, 3, \ldots, 18$ for the Perc feature, and show the results in the top graph of Fig. 1. It is interesting that only 4 steps (and hence 4 features used), the boosting process is ready to be steady. The lowest error rate is 25.76% corresponding to 18 rounds of boosting. For CepsL feature sets, we set L=40 for the AdaBoost, because Ceps40 is relatively better than other L values based on our previous experiments [8] [9]. The boosting process is shown in the middle graph of Fig. 1, which demonstrates the fast convergence of boosting. Finally, we use PercCeps120 (with dimension 258) feature for AdaBoost to see its behavior. From the bottom graph of Fig. 1, we can find that the boosting process is relatively smooth, while 20 rounds of boosting gives the lowest error rate 21.72%. After that, the error rates become a little higher. For comparison, we list the classification results of boosting together with NC and SVM in Table 1. It is obvious that by boosting, the error rates drop explicitly no matter what feature sets are used. However, the error rates of AdaBoost is still higher than those of SVM.

To test AdaBoost for audio retrieval, we take a similar strategy as that used in SVM based approach [9]. 16 decision boundaries are trained firstly using the training data. When a query audio is given, it selects one of the boundaries it should located in, and uses that boundary to rank other audio patterns in the test set based on their signed distances to the boundary. We call it distance-from-boundary (DFB) similarity measure, and used the SVM to learn the boundaries [9]. In Fig. 2, we compare the retrieval performance of NC, Boost and SVM measured by the average retrieval accuracy with respect to the number of top retrieved sounds, for the Perc, Ceps40 and PercCeps8 feature sets. The retrieval accuracy of Boosting is much lower than that based on SVM, and even a little lower than the NC based metric. The reason is probably that the AdaBoost method is not proper for the one-against-the-other classification strategy, in which only a small number of positive examples, e.g. 2-50, is presented, while the number of negative examples is about 160-200. Another reason may be that the NC classifier is not a good weak learner for AdaBoost on audio features. From the experiment, we also find that in retrieval, $T = 6, 10, 10$ are a little better than other T values for Perc, Ceps40, and PercCeps8 respectively, different from that in classification, where $T = 18, 40, 20$ give the lowest error rates for Perc, Ceps40, and PercCeps8 respectively.

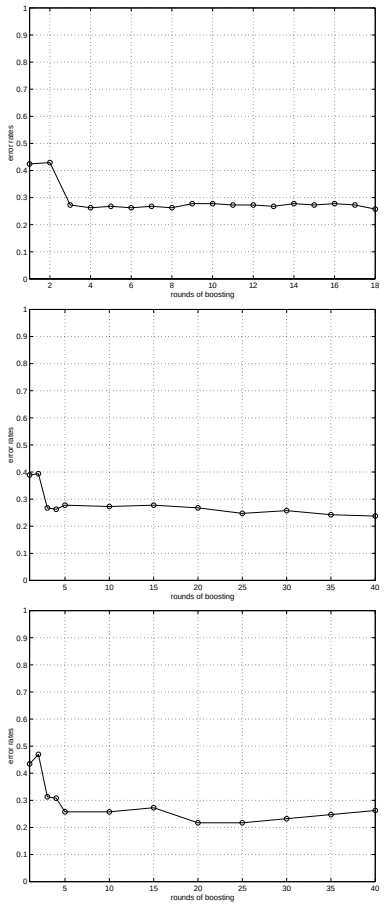

Figure 1: Boost performance with respect to rounds of boosting for the Perc (dimension 18), Ceps40 (dimension 80), and PercCeps120 (dimension 258) feature sets. It is observed that only a small number of boosting is enough for the AdaBoost on audio features. The problem of overfitting is not serious for audio features.

6. DISCUSSIONS AND CONCLUSIONS

We evaluated the performance of a most recently proposed algorithm AdaBoost for content-based audio classification and retrieval, in comparison with the SVM based approach and also the simple NC method. Boosting can improve the classification accuracy over the NC method, but still lower than the SVM. In retrieval, boosting can not improve the

Feature Set	NC	Boost	SVM
Perc	35.35%	25.76%	11.11%
Ceps40	42.42%	23.74%	16.67%
PercCeps8	38.89%	21.72%	8.08%

Table 1: Error rates obtained by using disjoint training and test sets. Partial results (only L=40, 8 for CepsL and PercCepsL respectively) are shown, which are relatively better than other L values for these classifiers.

performance of NC, and the accuracy is much lower than SVM based technique. The advantage of AdaBoost is its simplicity to implement and use. Our evaluation should activate more research on the boosting like algorithms in the context of content-based audio retrieval.

7. REFERENCES

[1] C. Cortes and V. Vapnik, "Support vector networks", *Machine Learning*, 20, 273-297, 1995.

[2] J. Foote, "Content-based retrieval of music and audio", in *Multimedia Storage and Archiving Systems II, Proc. of SPIE*, C. C. J. Kuo et al., Eds., 1997, vol. 3229, pp. 138–147.

[3] J. Foote, "An overview of audio information retrieval", *ACM-Springer Multimedia Systems*, 1998, In press.

[4] S. Foster, W. Schloss, and A. J. Rockmore, "Towards an intelligent editor of digital audio: Signal processing methods", *Computer Music Jounal*, vol. 6, no. 1, pp. 42–51, 1982.

[5] B. Feiten and T. Ungvary, "Organizing sounds with neural nets", in *Proceedings 1991 International Computer Music Conference*, San Francisco, 1991.

[6] B. Feiten and S. Günzel, "Automatic indexing of a sound database using self-organizing neural nets", *Computer Music Journal*, vol. 18, no. 3, pp. 53–65, 1994.

[7] Y. Freund and R. E. Schapire, A decision-theoretic generalization of online learning and an application to boosting. *J. Comp. & Sys. Sci*, 55(1):119-139, 1997.

[8] S. Z. Li, "Content-based classification and retrieval of audio using the nearest feature line method", *IEEE Trans. on Speech and Audio Processing*, Sep., 2000.

[9] S. Z. Li and G. Guo, Content-based audio classification and retrieval using svm learning, (invited talk), PCM, 2000.

[10] Z. Liu, J. Huang, Y. Wang, and T. Chen, "Audio feature extraction and analysis for scene classification", in *IEEE Signal Processing Society Workshop on Multimedia Signal Processing*, 1997.

[11] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data". *IEEE PAMI*, vol. 18, No, 8, 837-842, 1996.

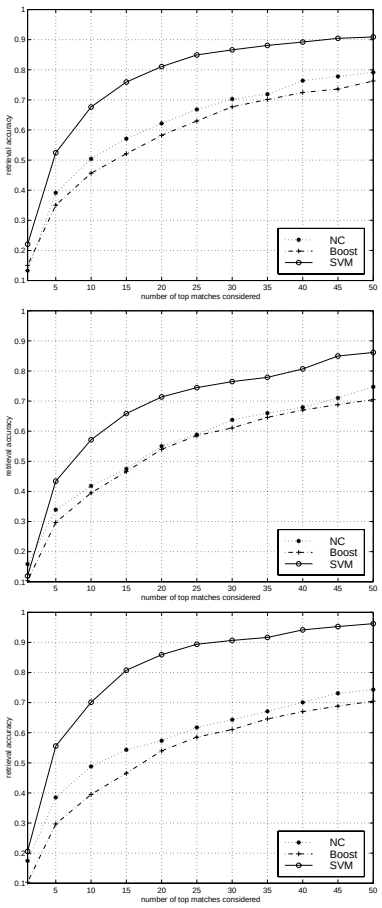

Figure 2: The retrieval performance comparison among different similarity measure: nearest center (NC), boost, and SVM respectively, for the Perc (top), Ceps40 (middle) and PercCeps8 (bottom) feature sets, using the disjoint training and test data.

[12] S. Pfeiffer, S. Fischer, and elsberg W. E, "Automatic audio content analysis", Tech. Rep. No. 96-008, University of Mannheim, Germany, April 1996.

[13] Stephen V. Rice, "Audio and video retrieval based on audio content", White pape, Comparisonics, P.O. Box 1960, Grass Valley, CA 95945, USA, April 1998.

[14] R. R. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann. Stat.*, 26(5):1651-1686, 1998.

[15] K. Tieu and P. Viola, Boosting image retrieval, in *Proc. of Computer Vision and Pattern Recognition*, v. 1, 228-235, 2000.

[16] V. N. Vapnik, *Statistical learning theory*, John Wiley & Sons, New York, 1998.

[17] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search and retrieval of audio", *IEEE Multimedia Magazine*, vol. 3, no. 3, pp. 27–36, 1996.