Content-Based Audio Classification and Retrieval by Support Vector Machines

Guodong Guo and Stan Z. Li

***Abstract*—Support vector machines (SVMs) have been recently proposed as a new learning algorithm for pattern recognition. In this paper, the SVMs with a binary tree recognition strategy are used to tackle the audio classification problem. We illustrate the potential of SVMs on a common audio database, which consists of 409 sounds of 16 classes. We compare the SVMs based classification with other popular approaches. For audio retrieval, we propose a new metric, called distance-from-boundary (DFB). When a query audio is given, the system first finds a boundary inside which the query pattern is located. Then, all the audio patterns in the database are sorted by their distances to this boundary. All boundaries are learned by the SVMs and stored together with the audio database. Experimental comparisons for audio retrieval are presented to show the superiority of this novel metric to other similarity measures.**

***Index Terms*—Audio classification, binary tree, content-based retrieval, distance-from-boundary (DFB), pattern recognition, support vector machines (SVMs).**

I. 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 [4], [5], [11], [23]. For these reasons, commercial products of audio retrieval are emerging, e.g., [23] (http://www.musclefish.com) and [17] (http://www.comparisonics.com).

While research in speech recognition, a closely related area, has a long history, research on content-based classification and retrieval of audio sounds is relatively new. An important recent work is done by Wold *et al.* [23], represented by their system called "Muscle Fish." The work distinguishes itself from the previous audio retrieval work [6]–[8] in its content-based capability. In the Muscle Fish system, various perceptual features, such as loudness, brightness, pitch, timbre 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 another work by Liu *et al.* [11], similar features plus subband energy ratios are used; the separability of different classes is evaluated in terms of the intra- and interclass scatters to identify highly correlated features; and a classification is performed by using a neural network. Foote [4] choose to use 12 mel-frequency cepstral coefficients (MFCCs) plus energy as the audio features. A tree-structured vector quantizer is used to partition the feature vector space into a discrete number of regions or "bins." Euclidean or cosine distances between histograms of sounds are compared and the classification is done by using the NN rule. In [14], a filter bank consisting of 256 phase-compensated gamma phone filters proposed by Cook [2] is used to extract audio features.

In this paper, we present new techniques for content-based audio classification and retrieval. In feature selection, perceptual features, mel-cepstral features and their combinations are considered for the task. While perceptual features like brightness, bandwidth and subband energies capture the spectral characteristics of the sounds, some of characteristic features of sounds are lost. The cepstral coefficients capture the shape of the frequency spectrum of a sound, from which most of the original sound signal can be reconstructed, and hence, provide a complement to the perceptual features.

For classification, a statistical learning algorithm called support vector machine (SVM) is used. The SVMs are proposed recently by Vapnik [3], [22], and have been used to solve some practical problems, such as face detection [13], three-dimensional (3-D) objects recognition [15], and so on [22]. Here, we focus on the use of SVMs to solve the audio classification problem, and propose to construct a binary tree structure for the SVMs in a multiclass recognition scenario.

In audio retrieval, conventional similarity measure based on Euclidean distance of the audio patterns to the query, suffers from three problems:

1) The retrieval results corresponding to different query patterns within the same class may be much different, which can be illustrated in Fig. 1(a). When point "a" (or "b," both belong to class 1) is used as a query, more samples from class 3 (or 2) can be retrieved in the top match (e.g., top 10 or 20), because these examples are closer to the query "a" (or "b") than other samples within class 1, when it is measured by the Euclidean or even Mahalanobis distance. But, most similar patterns can be retrieved in the top matches if point "c" is used as a query, because it is located close to the distribution center of the samples in class 1. However, the user may hope to get the same retrieval results no matter what queries are given.
2) The retrieval performance is sensitive to the sample topology: a compact distribution of the samples belonging to the same class like those in class 5 can make easy the retrieval task. However, the more scatter or arbitrary the distribution, the more deteriorate the retrieval performance, which is the case mostly in practice. The traditional Euclidean distance measure can not solve this problem.

Manuscript received March 23, 2000; revised January 2, 2001 and January 30, 2002.

G. Guo is with the Computer Sciences Department, University of Wisconsin, Madison, WI 53706 USA (e-mail: gdguo@cs.wisc.edu).

S. Z. Li is with Microsoft Research China, Beijing 100080, P.R. China.

Digital Object Identifier 10.1109/TNN.2002.806626

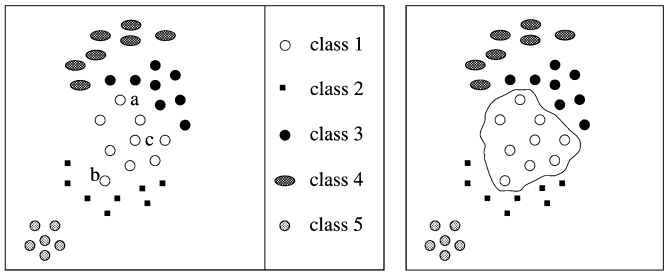

Fig. 1. (a) Examples of 2-D features belonging to five different classes. The three points a, b, and c are from class 1, however, the retrieval results in terms of these three queries may be much different using NN based similarity measure, and lots of patterns without any perceptual similarity to the queries can be retrieved on the top list. (b) A nonlinear boundary separates the samples of class 1 from that of classes 2–5. The same retrieval result can be obtained for these three query patterns.

3) The average retrieval accuracy is low.

In this paper, we take a new approach that provides a solution to above problems. A new metric called distance-from-boundary (DFB) is proposed to measure audio pattern similarities. The basic idea is that a (nonlinear) boundary separates the samples belonging to one class with the remaining. This nonlinear boundary encloses the similar patterns inside no matter what the distribution is. In Fig. 1(b), a nonlinear boundary separates all samples in class 1 with others belonging to classes 2–5. The signed distances to this nonlinear boundary can be used to rank these samples. Note that the boundary is not sensitive to the sample distributions.

The boundaries can be learned from training examples. We choose to use the SVMs to learn the boundaries because of their good generalization property. In addition, the boundaries learned by SVMs can be represented easily—several support vectors and their combination coefficients. Furthermore, the distance of the patterns in the database to this boundary can be simply calculated, just through computing the weighted sum of several dot products.

Our DFB-based similarity measure has three advantages: 1) the retrieval performance is relatively insensitive to the sample distribution; 2) the same retrieval results can be obtained with respect to different query patterns within the same class; 3) the boundary distance metric can give better retrieval performance than the traditional Euclidean distance based approach.

The paper is organized as follows. In Section II, we introduce the basic theory of the SVM for two-class classification problem, and our binary tree strategy to solve the multiclass problem. In Section III, we present our DFB metric for retrieval. Section IV describes the methods for feature extraction and combination. Then, Section V shows the extensive experiments for audio classification and retrieval. Finally, Section VI gives the conclusions and discussions.

II. SVMs

A. Basic Theory of SVMs

Given a set of training vectors belonging to two separate classes, $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$, where $\mathbf{x}_i \in R^n$ and $y_i \in \{-1, +1\}$, one wants to find a hyperplane $\mathbf{w}\mathbf{x} + b = 0$ to separate the data. In Fig. 2(a), there are many possible

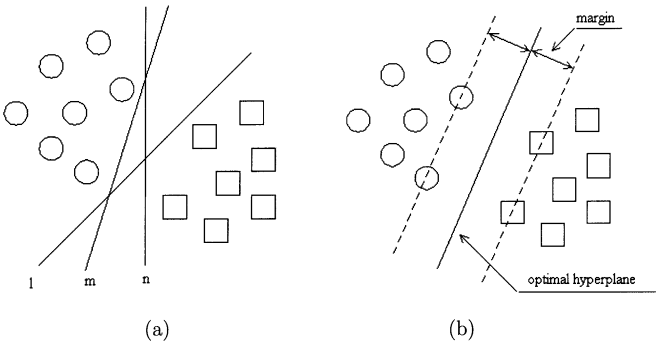

Fig. 2. Classification between two classes using hyperplanes. (a) Arbitrary hyperplanes l, m and n. (b) The optimal separating hyperplane with the largest margin identified by the dashed lines, passing the support vectors.

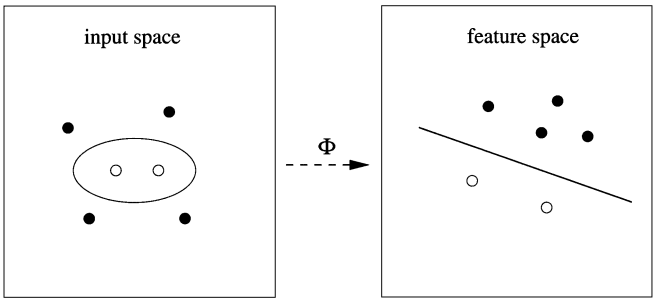

Fig. 3. Feature space is related to input space via a nonlinear map $\mathbf{\Phi}$, causing the decision surface to be nonlinear in the input space. By using a nonlinear kernel function, there is no need to do the mapping explicitly.

hyperplanes, but there is only one [shown in Fig. 2(b)] that maximizes the margin (the distance between the hyperplane and the nearest data point of each class). This linear classifier is termed the optimal separating hyperplane (OSH).

The solution to the optimization problem of SVM is given by the saddle point of the Lagrange functional

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i \{y_i[(\mathbf{w} \cdot \mathbf{x}_i) + b] - 1\} \quad (1)$$

where α_i are the Lagrange multipliers. Classical Lagrangian duality enables the *primal* problem (1) to be transformed to its *dual* problem, which is easier to solve. The solution is given by

$$\bar{\mathbf{w}} = \sum_{i=1}^{l} \bar{\alpha}_i y_i \mathbf{x}_i, \quad \bar{b} = -\frac{1}{2}\bar{\mathbf{w}} \cdot [\mathbf{x}_r + \mathbf{x}_s] \quad (2)$$

where $\mathbf{x}_r$ and $\mathbf{x}_s$ are any two support vectors with $\bar{\alpha}_r, \bar{\alpha}_s > 0, y_r = 1, y_s = -1$.

To solve the nonseparable problem, Cortes and Vapnik [3] introduced slack variables $\xi_i \geq 0$ and a penalty function, $F(\xi) = \sum_{i=1}^{l} \xi_i$, where the ξ are a measure of the misclassification error. The solution is identical to the separable case except for a modification of the Lagrange multipliers as $0 \leq \alpha_i \leq C, i = 1, \ldots, l$. The choice of C is not strict in practice, and we set $C = 200$ in all our experiments. We refer to [22] for more details on the nonseparable case.

The SVM can realize nonlinear discrimination by kernel mapping [22]. In Fig. 3, the samples in the input space can not be separated by any linear hyperplane, but can be linearly

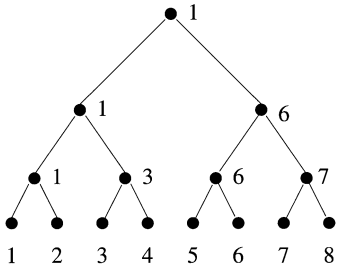

Fig. 4. Binary tree structure for eight classes audio classification. For a coming test audio, it is compared with each pair, and the winner will be tested in an upper level until the top of the tree. The numbers 1–8 encode the classes. By bottom-up comparison of each pair, a unique class number will finally appear on the top of the tree.

separated in the nonlinear mapped feature space. Note that here the feature space of the SVMs is different from the audio feature space.

There are three typical kernel functions for the nonlinear mapping [10], [22]: 1) Polynomial $K(\mathbf{x},\mathbf{y}) = ((\mathbf{x}\cdot\mathbf{y}+1))^d$, where parameter d is the degree of the polynomial; 2) Gaussian radial basis function $K(\mathbf{x},\mathbf{y}) = \exp(-((\mathbf{x}-\mathbf{y})^2/2\sigma^2))$, where parameter σ is the width of the Gaussian function; 3) Multilayer perception $K(\mathbf{x},\mathbf{y}) = \tanh(\text{scale}.(\mathbf{x}\cdot\mathbf{y}) - \text{offset})$, where *scale* and *offset* are two given parameters. However, another kernel function, called exponential radial basis function (ERBF) [10]

$$K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{|\mathbf{x}-\mathbf{y}|}{2\sigma^2}\right) \tag{3}$$

performs better than above three in our experimental comparisons. So we only evaluate the SVMs with ERBF kernel function in all our experiments.

For a given kernel function, the classifier is given by

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{l} \bar{\alpha}_i y_i K(\mathbf{x}_i,\mathbf{x}) + \bar{b}\right). \tag{4}$$

B. Multiple-Classes Classification

Previous sections describe the basic theory of SVM for two-class classification. A multiclass pattern recognition system can be obtained from two-class SVMs. Usually, there are two schemes for this purpose. One is the one-against-all strategy to classify between each class and all the remaining; the other is the one-against-one strategy to classify between each pair. For the latter, the problem is how to combine the binary classification results to obtain the final decision. A classical method is to use the voting strategy, however, the comparison will be $c(c-1)/2$ times, which results in heavy computation when the number of classes c is large.

We propose to construct a bottom-up binary tree for the multiclass classification. Suppose there are eight classes in the data set, the decision tree is shown in Fig. 4, where the numbers 1–8 encode the class labels. Note that the numbers encoding the classes are arbitrary without the meaning of ordering. By comparison between each pair, one class number is chosen representing the "winner" of the current two classes. The selected classes (from the lowest level of the binary tree) will come to the upper level for another round of tests. Finally, a unique class label will appear on the top of the tree. The underlying mechanism is similar to a sports game, in which each pair of persons or teams (class labels) does a competition, and only the winner can go to next round. The champion appears finally.

The advantage of the binary tree structure is to reduce the number of comparisons in the test stage. It just does comparisons for $(c-1)$ times, instead of the $c(c-1)/2$ times of comparisons in the traditional voting mechanism. This benefit is especially useful when the number of classes is very large. If a test pattern can be classified correctly, the final output of the binary tree is the same no matter how it is arranged. When a test pattern is classified incorrectly, the final output of the tree may be different, which depends on the arrangement of the classes in the leaves.

Denote the number of classes as c, the SVMs learn $c(c-1)/2$ discrimination functions in the training stage, and carry out comparisons of $c-1$ times under the binary tree structure. Note that although we just do $c-1$ comparisons in testing, we still need to train $c(c-1)/2$ classifiers. Because the nodes in the middle levels are not determined in advance, instead, they depend on the test example. If c is not the power of 2, we can decompose c as: $c = 2^{n_1} + 2^{n_2} + \cdots + 2^{n_I}$, where $n_1 \geq n_2 \geq \cdots \geq n_I$. Because any natural number (even or odd) can be decomposed into finite positive integers which are the power of 2. If c is an odd number, $n_I = 0$; otherwise, $n_I > 0$. Note that the decomposition is not unique, but the number of comparisons in the test stage is always $c-1$.

III. Distance From Boundary and Ranking

Recall that the pair $(\mathbf{w},b)$ defines a *separating hyperplane* or boundary of equation $\mathbf{w}\cdot\mathbf{x}+b=0$. The signed distance $D(\mathbf{x};\mathbf{w},b)$ from point $\mathbf{x}$ to the boundary $(\mathbf{w},b)$ is defined by $D(\mathbf{x};\mathbf{w},b) = (\mathbf{w}\cdot\mathbf{x}+b)/\|\mathbf{w}\|$. In nonlinear mapping with a kernel function, the boundary $(\mathbf{w},b,K)$ equation is

$$\sum_{j=1}^{m} \beta_j^* K(\mathbf{x}_j^*,\mathbf{x}) + b^* = 0 \tag{5}$$

where $\mathbf{x}_j^*(j=1,\ldots,m;m\leq l)$ are support vectors, β_j^* are the combination coefficients or weights, b^* is a constant, and $K(\cdot,\cdot)$ is the kernel function to perform the nonlinear mapping. Then, the signed distance $D(\mathbf{x};\beta^*,\mathbf{x}^*,b^*,K)$ from point $\mathbf{x}$ to the boundary $(\beta^*,\mathbf{x}^*,b^*,K)$ with kernel function $K(\cdot,\cdot)$ is defined as

$$D(\mathbf{x};\beta^*,\mathbf{x}^*,b^*,K) = \frac{\sum_{j=1}^{m}\beta_j^* K\left(\mathbf{x}_j^*,\mathbf{x}\right) + b^*}{\|\sum_{j=1}^{m}\beta_j^*\mathbf{x}_j^*\|}. \tag{6}$$

In the case of c classes, we have c boundaries. The kth boundary separates the examples of class k from others.

Definition 1 (signed distance to the kth boundary): If the boundary separating class k from others is $(\beta_k^*, \mathbf{x}_k^*, b_k^*, K)$, the signed distance of pattern $\mathbf{x}$ to the kth boundary is computed by

$$D(\mathbf{x}; \beta_k^*, \mathbf{x}_k^*, b_k^*, K) = \frac{\sum_{j=1}^{k_m} \beta_{kj}^* K\left(\mathbf{x}_{kj}^*, \mathbf{x}\right) + b_k^*}{\| \sum_{j=1}^{k_m} \beta_{kj}^* \mathbf{x}_{kj}^* \|} \tag{7}$$

where $\mathbf{x}_{kj}^* (j = 1, \ldots, k_m)$ are the support vectors to construct the kth boundary, and β_{kj}^* are the optimal coefficients, and b_k^* are some constants, $k = 1, \ldots, c$.

In our boundary distance measure, the patterns within the same class have positive distances to their enclosing boundary, while other patterns have negative distances to this boundary.

In retrieval, when a query pattern $\mathbf{q}$ is given, a boundary index k^* related to the query pattern is first found by

$$k^* = \arg \max_{1 \le k \le c} D(\mathbf{q}; \beta_k^*, \mathbf{x}_k^*, b_k^*, K). \tag{8}$$

Then, (7) is used to calculate the signed distances of all patterns $\mathbf{x}_i, i = 1, \ldots, N$ in the audio database to the k^*th boundary $D(\mathbf{x}_i; \beta_k^*, \mathbf{x}_k^*, b_k^*, K)$.

Definition 2 (distance-from-boundary for ranking): The signed distance to the kth boundary $D(\mathbf{x}_i; \beta_k^*, \mathbf{x}_k^*, b_k^*, K)$ are calculated and sorted in descending order, thus to rank the patterns $\mathbf{x}_i, i = 1, \ldots, N$ in the database.

In DFB similarity measure, the audio patterns located inside a boundary are considered similar. The ranking of the patterns are based on their distances to the boundary. Sometimes, we need to take into consideration the degree of similarity. That is, how similar these patterns to the query. In other words, there are still differences among the patterns located inside a boundary. To solve this problem we further rank the patterns located inside a boundary (returned in Definition 2), and call it boundary constrained similarity measure (BCSM).

Definition 3 (boundary constrained similarity measure): The patterns with positive distances to the kth boundary $D(\mathbf{x}_i; \beta_k^*, \mathbf{x}_k^*, b_k^*, K)$ are sorted again based on their Euclidean distances to the query (point to point), the smaller the distances, the similar the patterns to the query. Thus, to rank the patterns $\mathbf{x}_i, i = 1, \ldots, N_p$ located inside the boundary $D(\mathbf{x}_i; \beta_k^*, \mathbf{x}_k^*, b_k^*, K)$. While the ranking of patterns outside the boundary remains the same as that in Definition 2.

We assume that the closer the patterns to the query in the Euclidean space, the more similar they are to the query in BCSM. Note that the metrics of DFB and BCSM have no difference in the calculation of the retrieval accuracy. The difference is only the positions of the similar patterns in the top matches. Hence, we just show the experimental results based on DFB. Further research is to consider the positions of the similar patterns in the evaluation methodology.

IV. Audio Feature Selection

Before classification and retrieval, the audio features are extracted first. An audio signal (8-bit ISDN μ-law encoding) is pre-emphasized with parameter 0.96 and then divided into frames. Given the sampling frequency of 8000 Hz, the frames are of 256 samples (32 ms) each, with 25% (64 samples or 8 ms) 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 pre-emphasized signal magnitude at i and 400^2 is an empirical threshold. Then audio features are extracted from each nonsilent frame.

A. Definition of Audio Features

Two types of features are computed from each frame: 1) perceptual features, composed of total power, subband powers, brightness, bandwidth, and pitch and 2) MFCCs. Their definitions are given in the following, where the FFT coefficients $F(\omega)$ are computed from the frame.

- *Total Spectrum Power*. Its logarithm is used: $P = \log(\int_0^{\omega_0} |F(\omega)|^2 d\omega)$, where $|F(\omega)|^2$ is the power at the frequency ω and $\omega_0 = 4000$ Hz is the half sampling frequency.
- *Subband Powers.* The frequency spectrum is divided into four subbands with intervals $[0, (\omega_0/8)], [(\omega_0/8), (\omega_0/4)], [(\omega_0/4), (\omega_0/2)]$, and $[(\omega_0/2), \omega_0]$. The logarithmic subband power is used, $P_j = \log(\int_{L_j}^{H_j} |F(\omega)|^2 d\omega)$, where L_j and H_j are lower and upper bound of subband j.
- *Brightness.* The brightness is the frequency centroid $\omega_C = (\int_0^{\omega_0} \omega |F(\omega)|^2 d\omega / \int_0^{\omega_0} |F(\omega)|^2 d\omega)$.
- *Bandwidth.* Bandwidth B is the square root of the power-weighted average of the squared difference between the spectral components and the frequency centroid, $B = \sqrt{(\int_0^{\omega_0} (\omega - \omega_C)^2 |F(\omega)|^2 d\omega / \int_0^{\omega_0} |F(\omega)|^2 d\omega)}$.
- *Pitch Frequency*. A simple pitch detection algorithm, based on detecting the peak of the normalized autocorrelation function, is used. The pitch frequency is returned if the peak value is above a threshold ($T = 0.65$, chosen empirically), or the frame is labeled as nonpitched otherwise.
- *Mel-Frequency Cepstral Coefficients*. These are computed from the FFT power coefficients ([16, p. 189]). The power coefficients are filtered by a triangular bandpass filter bank. The filter bank consists of $K = 19$ triangular filters. They have a constant mel-frequency interval, and covers the frequency range of 0 Hz–4000 Hz. Denoting the output of the filter bank by $S_k (k = 1, 2, \ldots, K)$, the MFCCs are calculated as, $c_n = \sqrt{(2/K)} \sum_{k=1}^{K} (\log S_k) \cos[n(k - 0.5)\pi/K]\ n = 1, 2, \ldots, L$, where L is the order of the cepstrum.

B. Formation of Feature Sets

The means and standard deviations of the above eight original perceptual features are computed over the nonsilent frames, thus to form a 16-dimensional feature vector. Adding the silence ratio (number of silent frames/total number of frames) and the pitched ratio (number of pitched frames/total number of frames) to this vector gives an augmented 18-dimensional perceptual feature vector, named "perc." Each x_i of the 18 components in

the perc set is normalized according to $x_i' = (x_i - \mu_i)/\sigma_i$ (correlations between different features are ignored) where the mean μ_i and standard deviation σ_i are calculated over all the training set. This gives the final perceptual feature set, named "Perc."

The means and standard deviations of the L MFCCs are also calculated over the nonsilent frames, giving a $2L$-dimensional cepstral feature vector, named "Ceps$L$." The Perc and Ceps$L$ feature sets are weighted and then concatenated into still another feature set, named "PercCeps$L$," of dimension $18 + 2L$. The weighting is done as follows: There are 18 perceptual components in the Perc and $2L$ cepstral components in the Ceps$L$. Each of the 18 components has the unit standard deviation (std) after the normalization, and the total std of the 18 components is $s_1 = 18 \times 1$. The $2L$ components of the Ceps$L$ set are not normalized and have the total std of $s_2 = \sum_{i=1}^{2L} \sigma_i$ where σ_i is the std of the ith component. To account for the relative reliability of the two sets, the two sets are weighted by $(1/s_1)$ and $(1/s_2)$, are concatenated into PercCeps$L = (\mathrm{Perc}/s_1) \oplus (\mathrm{Ceps}L/s_2)$ where $\oplus$ stands for the concatenation operation.

V. Experiments on Audio Classification and Retrieval

The following experiments are aimed to evaluate 1) several classification and retrieval methods, that is, the SVM based approach with respect to NN (nearest neighbor), k-NN and NC (nearest center) and 2) three types of feature sets, namely Perc, Ceps, PercCeps. The results will also be compared with that of Muscle Fish [23] obtained from its web site.

The k-NN is a decision rule for classification [9] while the NC can be used for both classification and retrieval. In NC, a class is represented by the center of the prototypes belonging to that class, and the distance between the query and a class is that between the query and the class center.

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 [19] and [23]. 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 misclassified 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 [12]. It is defined as the average percentage number of patterns belonging to the same class as the query in the top n matches.

A. Evaluation With Disjoint Training and Test Sets

In the first evaluation, the 409 sounds are partitioned into a training set of 211 sounds and a test set of 198 sounds. 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.

TABLE I
Error Rates (and Number of Errors) Obtained by Using Disjoint Training and Test Sets. The Ticked Items (Along Columns) Indicate the Best Result of Each Algorithm With Respect to the Feature Set of CepsL and PercCepsL for $L = 5, 8, 10, 15, 20, 40, 60, 80, 100,$ and 120

Feature Set	SVM	NN	5-NN	NC
Perc	11.11% (22)	16.67% (33)	23.23% (46)	35.35% (70)
Ceps10	14.65% (29)√	21.21% (42)	30.30% (60)	55.05% (109)
Ceps40	16.67% (33)	19.70% (39)√	24.24% (48)√	42.42% (84)
Ceps80	17.17% (34)	20.71% (41)	25.25% (50)	40.40% (80)√
PercCeps5	12.63% (25)	17.68% (35)	21.21% (42)	43.43% (86)
PercCeps8	8.08% (16)√	13.13% (26)√	22.22% (44)	38.89% (77)
PercCeps60	10.61% (21)	16.16% (32)	21.72% (43)	32.32% (64)√
PercCeps80	10.10% (20)	15.66% (31)	20.71% (41) √	32.83% (65)

TABLE II
Error Rates (and Number of Errors) Obtained by Using Leave-One-Out Test. The Ticked Items (Along Columns) Indicate the Best Result Obtained by Each Algorithm With Respect to the Feature Set of CepsL and PercCepsL for $L = 5, 8, 10, 15, 20, 40, 60, 80, 100,$ and 120

Feature Set	SVM	NN	5-NN	NC
Perc	13.69% (56)	13.94% (57)	24.45% (100)	34.96% (143)
Ceps10	15.65% (64)√	24.94% (102)	33.25% (136)	54.77% (224)
Ceps40	16.38% (67)	22.98% (94)√	28.36% (116)√	42.05% (172)
Ceps60	16.38% (67)	23.96% (98)	30.07% (123)	41.56% (170)√
PercCeps5	12.71% (52)	15.16% (62)	19.80% (81)√	41.81% (171)
PercCeps8	11.00% (45)√	13.94% (57)√	20.78% (85)	37.65% (154)
PercCeps20	11.74% (48)	21.03% (86)	24.94% (102)	32.03% (131)√
PercCeps60	11.25% (46)	14.91% (61)	22.74% (93)	33.99% (139)

Table I shows the error rates (and the numbers of errors in brackets) of the four classification methods and the three types of feature sets. Because of the space limits, we only show partial results for the CepsL and PercCepsL, although we have tried for $L = 5, 8, 10, 15, 20, 40, 60, 80, 100,$ and 120. In addition, other results not shown here are not better than those in the table. Fig. 7 shows the retrieval performance of SVM, NN and NC measured in the average retrieval accuracy as a function of the number of top retrieved sounds, for the Perc, Ceps10, and PercCeps8 feature sets.

B. Evaluation by Leave-One-Out Tests

Secondly, each of the 409 sounds in the database is used as the query in turn. When a sound is used as the query, it is *not* used as a prototype, so the prototype set consists of the entire database minus the query. This is so called the "leave-one-out" test. All samples excluding the query q are used for training the SVMs, and a binary tree strategy is used to classify q.

Table II shows the error rates (and the numbers of errors) of the four classification methods and the three types of feature sets. Fig. 8 shows the retrieval performance of SVM, NN, and NC measured in the average retrieval accuracy as a function of the number of top retrieved sounds, for the Perc, Ceps10, and PercCeps8 feature sets.

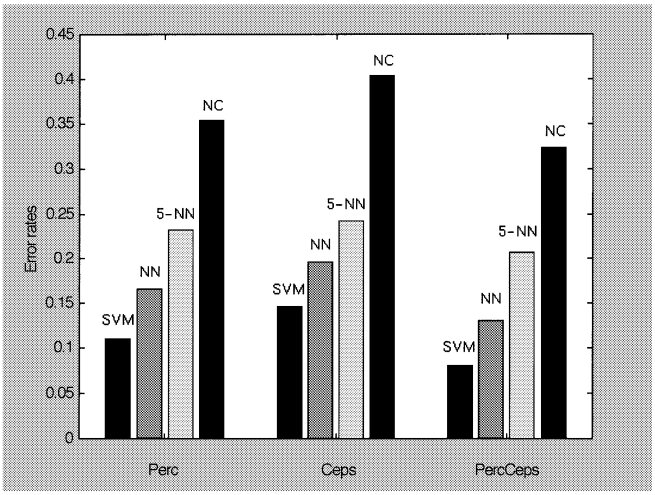

Fig. 5. Lowest error rates of four algorithms: SVM, NN, 5-NN, and NC, with respect to the three feature sets: Perc, Ceps, and PercCeps. The best results of each algorithm correspond to different feature dimensions as shown in Table I. The tests are on the disjoint training and test data.

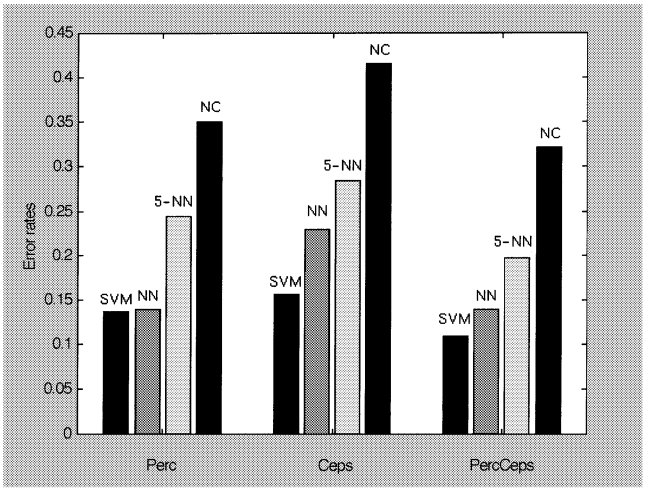

Fig. 6. Lowest error rates of these four algorithms with respect to the three feature sets: Perc, Ceps, and PercCeps. Note that the best results of each algorithm correspond to different feature dimensions as shown in Table II, evaluated by the leave-one-out test strategy.

C. Summary

For the SVMs, the kernel parameter σ can be set between 4 and 6 by trial and error. The values beyond this range can make worse the performance of the SVMs. We also find that $\sigma = 4$ is a little better for the Perc and Ceps feature sets, while $\sigma = 6$ is a little better for the PercCeps feature sets. This kind of parameter setting is used in both evaluations.

Among all the Ceps feature sets for the SVMs, Ceps10 is preferred over the others in terms of the error rate. The concatenation of the Perc and Ceps$L$ into PercCeps$L$ leads to improvements for most L values. The PercCeps8 feature set yields lower error rates than any perceptual or cepstral feature set alone. Overall, SVM + PercCeps8 yields the lowest error rate of 11.00% when evaluated by the leave-one-out test, while 8.08% by the disjoint training and test, of all combinations of methods and feature sets. The best performance of each classification algorithm with respect to the three feature sets are shown in Fig. 5 for the disjoint training and test data, and in Fig. 6 for the leave-one-out test.

For audio retrieval, the DFB based similarity measure (with the boundaries learned by the SVMs) has consistently higher accuracy than the other methods, which can be seen in the average retrieval accuracy curves in Figs 7 and 8, respectively. The results are consistent for both evaluations. Therefore, some conclusions can be drawn. 1) The SVM based approaches yield consistently lower error rates and higher retrieval accuracy than the other methods for all the feature sets. 2) The concatenation of the two types of feature sets into PercCeps$L$ leads to improvements for most L values. 3) The combination of SVM + PercCeps8 gives the overall best results among all methods and feature sets.

D. Comparison With Existing Systems

The Muscle Fish [23] is a famous content-based audio classification and retrieval system. Various perceptual features such as loudness, brightness, pitch, timbre are used to represent a sound, and the NN rule are used for classification and retrieval. Its classification error rate is 19.07% (78 errors out of 409 queries), as obtained from the Muscle Fish web interface http://www.musclefish.com/cbrdemo.html, which is roughly equivalent to the leave-one-out test with a modified version of NN. In comparison, the error rates of the leave-one-out test 13.94% (57 errors) for the NN+Perc method and 11.00% (45 errors) for the SVM + PercCeps8 method, respectively. The lowest error rate of 11.00%, obtained with SVM+PercCeps8, is significantly lower than that of Muscle Fish.

VI. Discussion and Conclusion

We have developed a multiclass classification strategy for the use of SVMs to solve the audio classification problem. The new recognition strategy extends the capability of a traditional bipartite framework to the solving of multiclass problems. We have presented the audio classification ERBF SVMs with the new strategy. Experimental results show that SVMs can be effectively trained for audio classification and can achieve lower error rate.

We have also presented a new metric called DFB for content-based audio retrieval. The boundaries can be learned effectively by the SVMs, and the signed distances are computed simply and efficiently by the weighted sum of dot products. The retrieval performance is significantly better than the traditional Euclidean distance based approach.

For the similarity measure based on DFB, we choose to use the SVMs to learn the boundaries. In case the number of support vectors is large, the computation cost will be heavy. To solve this problem, the reduced set method [1], [20] can be used to get a small number of vectors to represent the boundaries. Another way is to use the relevance vector machine [21] to learn each boundary, which can deliver just a small number of vectors to represent a boundary. Our focus is to propose a new metric called DFB or BCSM for similarity measure in audio retrieval, and select the SVMs to learn the boundaries. Simpler and more efficient method to learn the boundary can be expected in the near future.

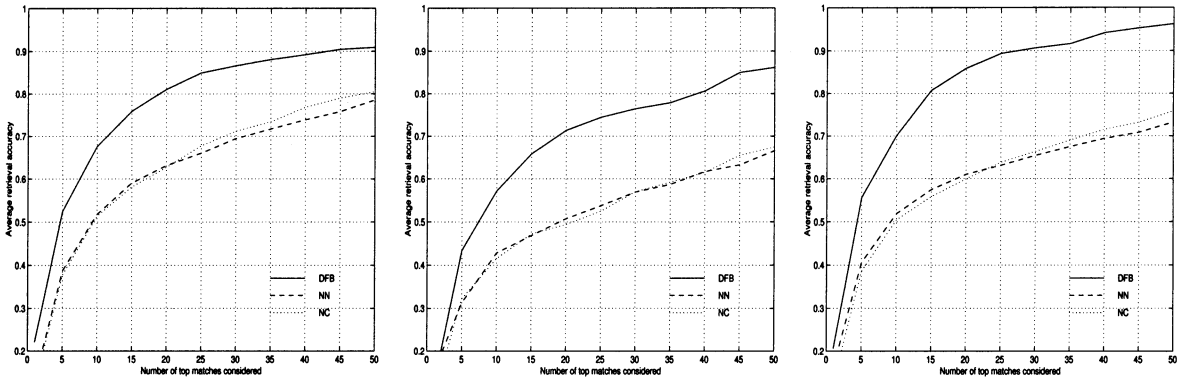

Fig. 7. Retrieval performance comparison among different similarity measures: NN, NC, and the distance-from-boundary (DFB) metric respectively, for the Perc (left), Ceps10 (middle), and PercCeps8 (right) feature sets, using disjoint training and test audio data. The DFB similarity measure gives the best average retrieval accuracy consistently.

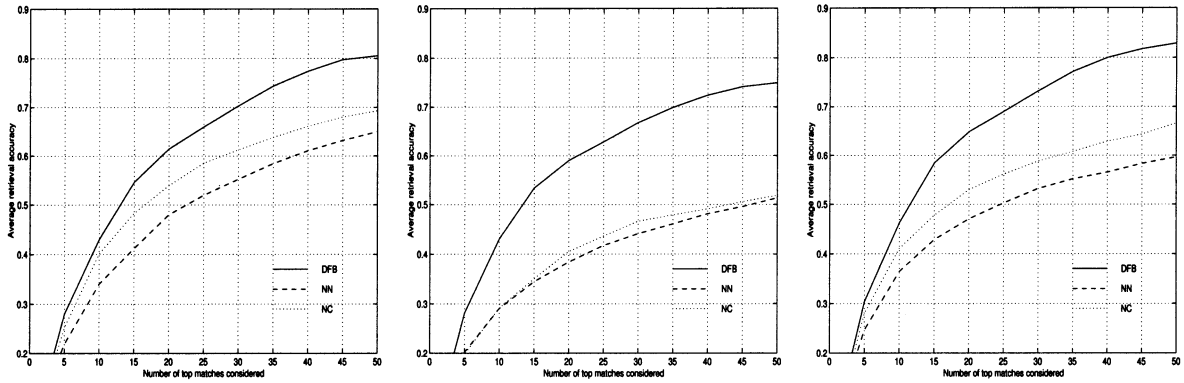

Fig. 8. Retrieval performance comparison among different similarity measures: NN, NC, and the distance-from-boundary (DFB) metric respectively, for the Perc (left), Ceps10 (middle), and PercCeps8 (right) feature sets, by the leave-one-out test strategy. The DFB similarity measure gives the best average retrieval accuracy consistently.

ACKNOWLEDGMENT

The authors are very grateful to the anonymous reviewers and the editors who give us many valuable comments and advice.

REFERENCES

[1] C. J. C. Burges, "Simplified support vector decision rules," in *Proc. 13th Int. Conf. Machine Learning*, L. Saitta, Ed., 1996, pp. 71–77.

[2] M. P. Cook, *Modeling Auditory Processing and Organization*. Cambridge, U.K.: Cambridge Univ. Press, 1993.

[3] C. Cortes and V. Vapnik, "Support vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, 1995.

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

[5] ——, "An overview of audio information retrieval," *ACM-Springer Multimedia Systems*, vol. 7, no. 1, pp. 2–11, Jan. 1999.

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

[7] B. Feiten and T. Ungvary, "Organizing sounds with neural nets," presented at the Proc. 1991 Int. Comput. Music Conf., San Francisco, CA, 1991.

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

[9] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Boston, MA: Academic, 1990.

[10] S. Gunn, "Support Vector Machines for Classification and Regression ISIS Tech. Rep.,", May 1998.

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

[12] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 837–842, Aug. 1996.

[13] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. CVPR*, 1997.

[14] S. Pfeiffer, S. Fischer, and W. E. Elsberg, "Automatic Audio Content Analysis," Univ. Mannheim, Mannheim, Germany, Tech. Rep. 96-008, Apr. 1996.

[15] M. Pontil and A. Verri, "Support vector machines for 3-D object recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 637–646, 1998.

[16] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[17] S. V. Rice, Audio and Video Retrieval Based on Audio Content, USA, Apr. 1998. White paper, Comparisonics, Grass Valley, CA.

[18] B. Scholkopf, S. Mika, C. J. C. Burges, P. Knirsch, K. Muller, G. Ratsch, and A. J. Smola, "Input space versus feature space in kernel-based methods," *IEEE Trans. Neural Networks*, vol. 10, pp. 1000–1017, Sept. 1999.

[19] S. Z. Li, "Content-based classification and retrieval of audio using the nearest feature line method," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 619–625, Sept. 2000.

[20] B. Scholkopf, S. Mika, C. J. C. Burges, P. Knirsch, K. Muller, G. Ratsch, and A. J. Smola, "Input space versus feature space in kernel-based methods," *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 1000–1017, 1999.

[21] M. E. Tipping, "The relevance vector machine," in *Advances in Neural Information Processing Systems 12*. Cambridge, MA: MIT Press, 2000.

[22] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[23] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search and retrieval of audio," *IEEE Multimedia Mag.*, vol. 3, pp. 27–36, July 1996.