The Application of Support Vector Machines for Classification of Audio Signals

Nataša Reljin[1], Dragoljub Pokrajac[2]

Abstract — **Identifying and classifying audio content is of great importance nowadays. There exists very large pool of audio signals, which need to be automatically classified. In this paper, we used wavelet descriptors for characterizing short music sequences, and performed classification based on linear support vector machines (SVM). Performed experiments provided good results with classification accuracy of more than 76%.**

Keywords — **Audio signals, wavelets, support vector machines, classification, confusion matrix.**

I. INTRODUCTION

The quantity of multimedia information (audio, video, pictures,…) stored in digital form increases on daily basis. Faced with such large pool of information, a human user encounters several challenges: how to find specific multimedia information based on content description; how to recognize and retrieve desired information fast; how to determine category to which novel content belongs; to which data it is similar? Indeed, fast recognition, retrieval and classification of multimedia content are current research topics [1-9]. Among all different types of multimedia information, in this paper we concentrate on audio signals and their classification.

To perform efficient classification, it is important to identify relevant features that describe audio signals and help distinguish signals belonging to different categories of interest. Ideally, we are interested in detecting a small number of features bearing a bulk of information necessary for classification. The features of interest should have similar values for objects in the same category, and significantly different values for objects in distinct classes.

Humans perform classification of sounds based on subjective criteria such as whether melodies sound alike, and use similarity in music categories such as rhythm, tonality, etc. In general, every musical sound has a specific timbre that results in highly complex signals [2]. Signal analysis and processing apparatus is necessary to extract descriptors suitable for automatic characterization of music sounds. Fourier analysis is widely used technique for describing similarities between long sequences of stationary signals composed of sine waves. Although it is used as a description tool in MPEG-7 compression standard [2], this method is not well suited to describe very specific features like those in mainly non-stationary polyphone musical fragments. For instance, when analyzing a signal by using a large window, the frequencies cannot be sufficiently resolved in time. In contrast, when using a small window, a fine time resolution is possible, but, low frequency components can no longer be identified. This means that for retrieving audio signals, which are time-varying, highly irregular and non-stationary, Fourier analysis is not capable of providing information about all frequencies contained in some sequence. In addition, this analysis does not provide any temporal information whatsoever. Furthermore, practical reasons dictate the use of short subsequences for retrieving audio material. When applying the Fourier analysis on these sequences, high resolution is not possible. For all these reasons, the use of wavelet transform (WT) is proposed [10]. WT provides multi-resolution analysis in both time and frequency domains. This way, details or global trends that cannot be identified in one resolution, could be detected in another. Wavelet transform is capable of distinguishing very small and delicate differences between signals, even from short fragments. Consequently, the wavelet transform is recognized as a powerful tool for identifying and describing audio content [2]. In our related research, we determined that four parameters from nine WT decomposition levels— maximum, minimum and their positions with respect to the beginning of the piece (a total of 36 features) — are sufficient for describing audio sequence [9]. In this paper, we use the same WT parameters to form feature vectors, which are associated to each music piece from the database. Each piece is assigned to one of the two classes according to the title of the song. After the dataset is formed, we train the classification model – a linear support vector machine. To evaluate the accuracy of the proposed classifier, we perform *K*-fold cross validation and use confusion matrix as a measure of classification performance [11-12].

The paper is organized as follows. Section II describes feature extraction using wavelets. In Section III we present classification methodology in more details, followed by experimental results in Section IV.

II. FEATURE EXTRACTION USING WAVELETS

In this section we describe extraction of relevant features for classifying musical sequences by means of wavelet coefficients. Wavelets are mathematical functions that split data into different time-frequency components, and then analyze each component based on a resolution matched to it [10]. Moreover, individual wavelet functions are localized in

[1]Nataša Reljin is with Delaware State University, AMTP Dept., 1200 N DuPont Hwy, Dover, DE, 19901, USA, E-mail: natasa.reljin@gmail.com

[2]Dragoljub Pokrajac is with Delaware State University, CIS Dept., AMTP Dept. and CREOSA, 1200 N DuPont Hwy, Dover, DE, 19901, USA, E-mail: dpokrajac@desu.edu

time, which is quite different from Fourier transform. Such a property of wavelets makes time-frequency analysis possible.

The continuous wavelet transform (CWT) transforms a continuous, square-integrable function $f(t)$, into a function $W_\psi(s,\tau)$ of two continuous real variables: translation, τ, and positive scale s, defined as:

$$W_\psi(s,\tau) = \int_{-\infty}^{\infty} f(t)\psi_{s,\tau}(t)dt\,. \qquad (1)$$

Here, the function $\psi_{s,\tau}(t)$, known as a *basis* (or *mother*) *wavelet*, is defined as:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right). \qquad (2)$$

As we can see, a basis wavelet is translated and scaled *wavelet function* $\psi(t)$. A wavelet function $\psi(t)$ is a zero-mean waveform confined in time (i.e., with limited duration). Scale s allows the compression or expansion of function: the larger scale factor the more substantial expansion in time. This way, analysis of observed signal on different frequency scales may be performed. By changing the translation coefficient, τ, wavelet moves along the temporal axis, permitting the analysis of entire signal $f(t)$ in time domain.

In practical applications, scale and translation changes are discrete. Typically, a *binary* (or *dyadic*) scaling system is used, where scale and translation are related through an integer k: $s = 2^k$, and $\tau = 2^k l$. Hence, the wavelet function can be expressed as:

$$\psi_{k,l}(t) = 2^{-k/2}\psi(2^{-k}t - l);\; k,l \in Z^+. \qquad (3)$$

Discrete wavelet transform (DWT) is obtained by discretization of time such that $t = m\cdot\Delta t, m \in Z$, and by replacing integral in Eq. (1) with infinite sum:

$$W_\psi(k,l) = \sum_{m=-\infty}^{\infty} f(m\Delta t)\psi_{k,l}(m\Delta t)\,, \qquad (4)$$

where discretized wavelet functions are defined by Eq. (3).

When performing a wavelet decomposition, the mother wavelet and its s-scaled replicas are shifted along the entire signal $f(t)$. For each scale, a wavelet transform is calculated along the whole shift τ. For one-dimensional signals, e.g., audio sequences, the wavelet decomposition results in approximation and details components. Signal $f(t)$ is convolved with low-pass filter followed by downsampling for obtaining approximation, and with the high-pass filter followed by downsampling for detail component. In the next step, approximation component splits into new approximation and detail components by using the same procedure, and so on. Coefficient k determines number of details: for example, k=2 means that in first step approximation (a_1) and detail (d_1) components are obtained; while in the second (last) step, new approximation (a_2) and new detail (d_2) components are obtained from approximation component a_1.

To describe music sequences, several different wavelets and wavelet families were considered in [2], as well as a variety of WT parameters. Here, we use a Daubechies wavelet of order 4 (db4) [2], [9]. Note that this wavelet does not have an explicit mathematical representation, but can be obtained from the roots of corresponding generating polynomial. To describe content of a music sequence, feature vectors are formed using the following four descriptors per each WT detail: maximum and minimum values, and their positions with respect to the beginning of the transformed sequence. One feature vector corresponds to one five-second subsequence of a musical sequence. To select relevant decomposition details, the characteristics of human aural system and properties of decomposition details are used. By listening different audio sequences we determined that a particular fragment may be recognized when high frequency content (corresponding to the first two details) is filtered out. On the other hand, high-order WT details (above the 11-th detail) consist of almost constant signals. Therefore, we decide to use WT details k=3 to k=11, i.e., total of 9 details and 4x9=36 features per example.

Feature vectors are stored in a *feature matrix*, $\mathbf{X}=\{x(i,j)\}=\{\mathbf{x}_i(j)\}$, where rows i=1,2,…,N correspond to examples—music pieces (subsequences), and columns j=1,2,…,d, to features (d=36 in our case). Thus, $\mathbf{x}_i$ denotes a d-element feature vector corresponding to the i-th subsequence.

III. Support Vector Machine Classifier

Classification is a mapping of feature vectors into a set of discrete class labels using a classifier—a suitably chosen parametric model [12]. To perform classification, model parameters are determined through *learning procedure* using labeled training dataset. Subsequently, a classifier is evaluated by performing classification on a test set, consisting of examples unseen during the learning procedure. During testing, a class label prediction is obtained from the model output and classification accuracy is determined by its comparison with a known class label.

Linear support vector machines (SVMs) belong to a group of generalized linear classifiers [12]. The main idea of support vector machines is to construct a hyperplane (e.g., a line in 2-dimensional space), which separates points that belong to two classes, such that the minimal distance between points and the separation hyperplane is maximized. The distance between points closest to the separation hyperplane and the hyperplane is referred to as *margin*. Points which are at the minimal distance from the separation hyperplane are referred to as *support vectors*. Support vector machines use structural risk minimization principle [13] and strive to achieve zero training error while minimizing the complexity of the model by minimizing its VC dimension (VC dimension is inversely proportional to the decision margin which SVMs maximize). If the linear separation is not possible, SVMs minimize the number of misclassified examples on the training set by introduction of slack variables and regularization.

Formally, SVM learning can be stated as the following quadratic programming problem [12]:

$$\min_{\mathbf{w},\xi_i,d_0}\left(\frac{1}{2}\mathbf{w}^T\mathbf{w}+C\sum_{i=1}^{N}\xi_i\right) s.t.$$
$$\left(\mathbf{w}^T\mathbf{x}_i+d_0\right)c_i \geq 1-\xi_i, i=1,...,N \quad (5)$$
$$\xi_i \geq 0, i=1,...,N.$$

Here, **w** is vector orthogonal to the separation plane, d_0 is the intercept of the separation hyperplane, $c_i \in \{-1,1\}$ is a class label of *i*-th example, ξ_i are slack variables and *C* is preset regularization constant.

Using Karush-Kuhn-Tucker (KKT) theorem [12], SVM learning can be performed as the optimization in dual space of Lagrangian multipliers λ_i. The learning phase reduces to the following optimization problem:

$$\max_{\lambda}\left(\sum_{i=1}^{N}\lambda_i-\sum_{l=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j c_i c_j \mathbf{x}_i{}^T\mathbf{x}_j\right) \text{ where}$$
$$\sum_{i=1}^{N}\lambda_i c_i = 0 \quad (6)$$
$$\lambda_i \geq 0, i=1,...,N$$
$$\lambda_i \leq C, i=1,...,N.$$

Classification of a new example $\mathbf{x}_{new}$ is performed as:

$$c_{new} = sign\left(\mathbf{w}^T\mathbf{x}_{new}+d_0\right), \quad (7)$$

which can be expressed using the Lagrangian multipliers as:

$$c_{new} = sign\left(\sum_{i:\lambda_i>0}\lambda_i c_i \mathbf{x}_i{}^T\mathbf{x}_{new}+\frac{1}{N_s}\left(\frac{1}{c_i}-\sum_{j:\lambda_j>0}\lambda_j c_j \mathbf{x}_i{}^T\mathbf{x}_j\right)\right), \quad (8)$$

where N_s denotes number of support vectors (i.e., number of non-zero Lagrangian multipliers).

Classifier performance can be evaluated by performing *K*-fold cross-validation [12] and using appropriate performance metrics. In *K*-fold cross-validation, available dataset is randomly split into *K* disjoint subsets, the following procedure is repeated *K* times: different *K*-1 disjoint subsets are used for learning model parameters, and the remaining subset is used for model evaluation. This guarantees optimal use of available data to train the model and fair assessment of its classification performance.

TABLE I:
CONFUSION MATRIX

	True Class 1	*True Class 2*
Detected Class 1	True Negative (TN)	False Positive (FP)
Detected Class 2	False Negative (FN)	True Positive (TP)

Performance can be measured using a confusion matrix, defined in Table I for a two-class problem. The confusion matrix provides summary for assignment of examples from each class to the predicted classes, using results from all *K* experiments in the cross-validation process. Based on the confusion matrix, the following performance measures are derived [11]:

$$Precision = TP/(TP+FP) \quad (9)$$
$$Recall = TP/(TP+FN) \quad (10)$$
$$F-value = \frac{2\cdot Recall\cdot Precision}{Recall+Precision}. \quad (11)$$

Note that *Precision* is the fraction of test set examples correctly labeled as belonging to the class 2 divided by the total number of elements labeled to the same class by a model. *Recall* is the partial accuracy when classifying examples from the class 2, and *F-value* is the harmonic means of these two ratios.

IV. EXPERIMENTS

We created and used audio database consisting of examples from four musical sequences. The sequences correspond to two different songs, and each song was performed by two different performers. Songs are sampled at 44.1 kHz and 16 bps. Each song is divided into pieces of 5 seconds, thus containing about 220,000 samples. One example in the database corresponds to each such music piece. In order to extract features, each piece was converted to mono and amplitudes are normalized to 1 [3].

We used the following songs: *Take a Chance on Me*, performed by ABBA[1] and by Erasure[2] (total of 49 pieces); and *Something Stupid*, performed by Frank Sinatra[3] and by Robbie Williams[4] (total 59 pieces). As an illustration, Fig. 1 shows a five-seconds music piece, which corresponds to a subsequence from the 10th to the 15th second of the ABBA's *Take a Chance on Me*, and its first 11 WT details (from top to down).

We assigned examples from *Take a Chance on Me* a class label -1 (referred to as class 1 in Table I), while the other song examples are assigned class label +1 (corresponding to the class 2). To balance classes, we use technique of under-sampling [14], where random 10 samples from a majority class were discarded.

For each piece from the dataset, we performed one-dimensional db4 wavelet transform, used details from 3 to 11 and selected wavelet descriptors (max, min and their positions) as features, thus constructing feature matrix as described in Section II.

We varied regularization parameter *C*, and for each parameter value we trained linear support vector machine. We applied *K*-fold cross-validation with *K*=10, and used *precision, recall and F-value* metrics as defined in Section III to evaluate the classification performance. The summary of results is given on Table II.

Regularization parameter *C* specifies trade-off between minimizing training-errors and model complexity. By changing the value of *C*, we could control to which extent misclassified points have influence on training. With *C*=1e6, 1e9 we achieved 100% percent classification accuracy on class 1 (*precision* is 1). The higher *C* initially led to improvement of *recall* (classification accuracy of class 2) which reached 76.5% for *C*=1e9. Subsequent increase of *C* worsened generalization performance, as reflected by decrease of all three observed performance measures.

Obtained classification accuracy compares well with results

[1] ABBA, *The Album*, Sweden, 1977.
[2] Erasure, *Erasure Pop!: The First 20 Hits*, 1992.
[3] Frank Sinatra, *The World we Knew*, Warner Music Group, 1967.
[4] Robbie Williams, *Swing when you're Winning*, EMI Int'l, 2001.

reported in [9]. There, using a different dataset, we demonstrated identification and retrieval of a music piece with accuracy of 60%, by using neural networks. In contrast, by using linear SVM classifiers, here we report partial classification accuracies of 76.5% and 100%. For fair comparison, however, the techniques to be compared should be evaluated on the same databases.

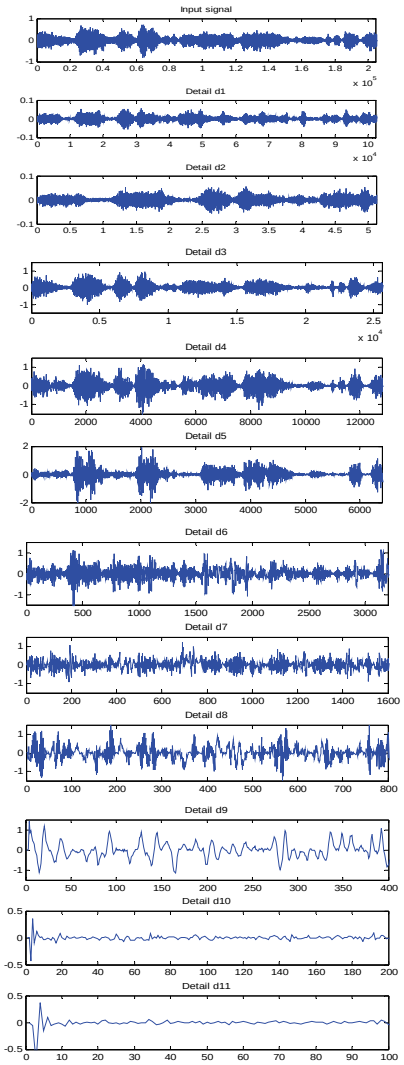

Fig. 1. Music sequence from ABBA's song *Take a chance on Me*, and its first 11 details

TABLE II
PERFORMANCE MEASURES

C	*Recall*	*Precision*	*F-value*
1e6	0.73469	1	0.84706
1e9	0.76531	1	0.86705
1e12	0.69388	0.96333	0.8067

V. CONCLUSION

For describing short audio sequences, wavelet coefficients are very useful descriptors, since they can capture very small and delicate differences between time-varying signals. By using linear support vector machines, an accurate classification of audio sequences can be performed (partial accuracies of 76.5% and 100% for a two-class problem). Our work in progress involves expanded datasets and using SVM classifier in transformed space. Also, we will experiment on multi-class classification using generalized SVM paradigm.

ACKNOWLEDGEMENT

D. Pokrajac and N. Reljin have been partially supported by NIH (grant #2 P20 RR016472-04), DoD/DoA (award 45395-MA-ISP) and NSF (awards # 0320991, #HRD-0310163, #HRD-0630388).

REFERENCES

[1] G. Lu, "Techniques and data structures for efficient multimedia retrieval based on similarity", *IEEE Trans. Multimedia*, vol. 4, no. 3, pp. 372-384, Sept. 2002.

[2] S. Rein, M. Reisslein, "Identifying the classical music composition of an unknown performance with wavelet dispersion vector and neural nets", *Information Sciences*, vol. 176, no. 12, pp. 1629-1655, June 2006.

[3] T. Pohle, E. Pampalk, G. Widmer, "Evaluation of frequently used audio features for classification of music into perceptual categories", Conference CD, *Conf. CBMI 2005*, Riga, Latvia, 21-23 June 2005.

[4] M. Clausen, F. Kurth, "A unified approach to content-based and fault-tolerant music recognition", *IEEE Trans. Multimedia*, vol. 6, no. 5, pp. 717-731, Oct. 2004.

[5] P. Muneesawang, L. Guan, "An interactive approach for CBIR using a network of radial basis functions", *IEEE Trans. Multimedia*, vol. 6, no. 5, pp. 703-716, Oct. 2004.

[6] R. Brunelli, O. Mich, "Image retrieval by examples", *IEEE Trans. Multimedia*, vol. 2, no. 3, pp. 164-171, Sept. 2000.

[7] K.-M. Lee, W. N. Street, "Cluster-driven refinement for content-based digital image retrieval", *IEEE Trans. Multimedia*, vol. 6, no. 6, pp. 817-827, Dec. 2004.

[8] S. Čabarkapa, N. Kojić, V. Radosavljević, G. Zajić, B. Reljin, "Adaptive content-based image retrieval with relevance feedback", *Proc. EUROCON 2005 Conf.*, vol. 1, pp. 147-150, Belgrade, Serbia, 21-24 Nov. 2005.

[9] V. Đorđević, N. Reljin, I. Reljin, "Identifying and retrieving of audio sequences by using wavelet descriptors and neural network with user's assistance", *Proc. EUROCON 2005 Conf.*, vol. 1, pp. 167-170, Belgrade, Serbia, 21-24 Nov. 2005.

[10] M. Vetterli, J. Kovačević, *Wavelets and Subband Coding*, Prentice Hall, Signal Processing Series, Englewood Cliffs, NJ, 1995.

[11] P.-N. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, Addison-Wesley, 2005.

[12] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[13] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 2000.

[14] A. Lazarević, D. Pokrajac, J. Nikolić, "Applications of neural networks in network intrusion detection", *Proc. NEUREL 2006 Conf.*, pp. 59-64, Belgrade, Serbia, 25-27 Sept. 2006.