Acoustic events detection with Support Vector Machines

Jozef VAVREK, Matúš PLEVA, Jozef JUHÁR

Dept. of Electronics and Multimedia Communications, FEI TU of Košice,
Slovak Republic

vavrekjozef@centrum.sk, matus.pleva@tuke.sk, jozef.juhar@tuke.sk

***Abstract* - This paper deals with detection and classification of acoustic events which could indicate potentially dangerous situation. The main objective of the paper is to determine the classification accuracy of gun shots taken in noisy environment. The detection and classification method is based on Support Vector Machines. The training and testing data are gunshots which were recorded in an open space and consequently degraded with background street noise.**

***Keywords* – acoustic event, gunshot, detection, classification, Support Vector Machine, MFC Coefficients, optimal separating hyperplane**

I. Introduction

In general, the purpose of sound (event) recognition is to understand whether a particular sound belongs to a certain class or to distinguish one class of data from the rest. This is a sound recognition problem, similar to voice, speaker, or speech recognition. Sound recognition systems can be partitioned into two main modules. First, a sound detection stage isolates relevant sound segments from the background by detecting abrupt changes in the audio stream. Then, a classifier tries to assign the detected sound to a category. Generally, the conventional event detection methods are based on the energy calculation and sound classifiers are often based on statistical models.

There exist some types of data classifiers, for example Gaussian mixture models (GMMs) [1], hidden Markov models (HMMs) [1], neural networks (NNs) [2], and Support Vector Machines (SVM) [2], [3]. All these classification techniques are based on training models that were used for learning processes.

One of the most widely used methods for classification of sound events is the SVM. This paper describes the basic features of the SVM classification method. SVM is applied in two experiments, which deals with the processing of recordings of the gun shots.

II. Feature extractions

Feature extraction is also called a parameterization method. We have used the mel-cepstrum model as a main parameterization method in this work. This method deals with mel-frequency-cepstral coefficients (MFCC) [4], [5].

Mel-frequency cepstral coefficients (MFCC) have been a popular signal representation method used in many audio classification tasks. The basis for the MFCC mel-frequency scale is derived from the human perceptual hearing system. The calculation of the MFCC parameters begins with the segmentation of a signal into overlapping frames. The power spectrum of each frame is transformed into the logarithmic mel-frequency spectrum, using a filter-bank of 24 triangular filters.

In this work we used a filter-bank of 24 triangular filters, 24 and 12 MFCC per frame (in most cases the first 12 MFCC). The audio signal was divided into frames of length 25ms with overlapping 15ms.

III. Support vector machines

Support vector machines have become a popular tool in many kinds of machine learning tasks. SVMs are based on *statistical learning theory* and *structural risk minimization* [2]. SVM is the relatively new promising method for learning separating functions in pattern recognition (classification) tasks and represents novel learning techniques that have been introduced in the theory of *VC bounds*. The *VC* (Vapnik-Chervonenkis) *dimension h* [2] is a property of a set of approximating functions of the learning machine that is used in all important results in the statistical learning theory. VC dimension increases as the number of weights vector parameters increases. In other words, a learning machine with many parameters will have a high VC dimension, whereas a machine with few parameters will have a low VC dimension.

In the simplest pattern recognition tasks, support vector machines use a linear separating hyperplane to create a *classifier with a maximal margin*. In order to do that, the learning problem is cast as a *constrained nonlinear optimization problem*. In cases when given classes can not be linearly separated in the original input space, the SVM firstly nonlinearly transforms the original input space into a higher-dimensional *feature space*. This transformation could be achieved by using various nonlinear mappings: polynomial, sigmoid as in multilayer perceptrons, RBF mappings having as basic functions the radially symmetric functions such as Gaussian, different spline functions. After this nonlinear transformation step, the task of an SVM in finding the linear *optimal separating hyperplane* in this *feature space* is relatively trivial. The resulting hyperplane in feature space will be optimal in the sense of being a maximal margin classifier with respect to training data. Training data are represented in the form:

$$(x_1, y_1), ..., (x_m, y_m) \in X \times \{\pm 1\} \ , \tag{3.1}$$

where X is some nonempty set from which the patterns x_i (sometimes called *cases, inputs, instances*, or *observations*) are taken, usually referred to as the domain; the y_i are called *labels, targets*, or *outputs*. There are only two classes of patterns and they are labeled by +1 and -1. This is a particularly simple situation, referred to as (binary) pattern recognition or (binary) classification. In learning, we want to be able to generalize to unseen data points. In the case of pattern recognition, this means that giving some new pattern $x \in$ X, we want to predict the corresponding $y \in \{\pm 1\}$ [2].

A. Linear maximal margin classifier for linearly separable data

Consider the problem of binary classification. Training data are given as

$$(x_1, y_1), (x_2, y_2), ..., (x_l, y_l), x \in R^n, y \in \{+1, -1\}. \tag{3.2}$$

Data are linearly separable, and there are many different hyperplanes that can perform separation as we can see on Fig. 1.

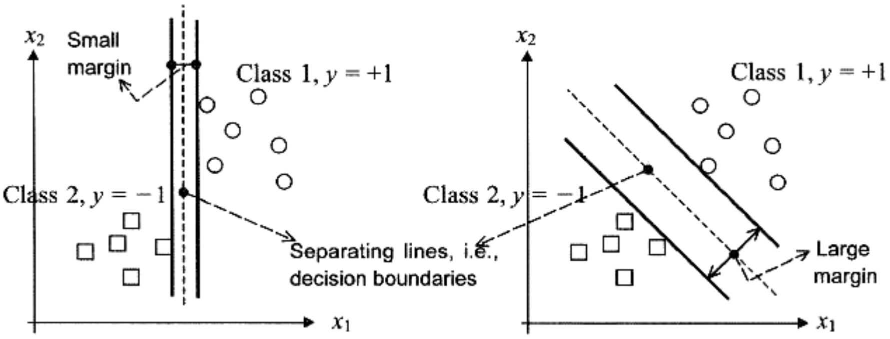

Fig. 1. Two out of many separating lines: right, good one with a large margin, and left, a less acceptable one with a small margin

Only sparse training data are available. In the case of classification of linearly separable data, this idea is transformed into the following approach: among all the hyperplanes that minimize the training error, we should find the one with the largest margin.

Using the given training examples during the learning stage, the machine finds parameters $\mathbf{w} = [w_1 \, w_2 \, ... \, w_n]^T$ and b of a discriminant of the decision function:

$$d(\mathbf{x}, \mathbf{w}, b) = \mathbf{w}^T \mathbf{x} + b = \sum_{i=1}^{n} w_i x_i + b, \tag{3.2}$$

where **x, w**, $\in R^n$, and the scalar b is called *bias*. After the successful training stage, using the weights obtained, the learning machine, given a previously unseen pattern **x**, produces output *o* according to an indicator function given as

$$if = o = sign(d(\mathbf{x}, \mathbf{w}, b)), \tag{3.3}$$

where *o* is the standard notation for the output from a learning machine. In other words, the decision rule is:

if $d(\mathbf{x}_\mathrm{p}, \mathbf{w}, \mathrm{b}) > 0$,
then pattern $\boldsymbol{x}_p$ *belongs to a class 1 (i.e.* $o = y_1 = +1$*)*, and

if $d(\mathbf{x}_\mathrm{p}, \mathbf{w}, \mathrm{b}) < 0$,
then pattern $\boldsymbol{x}_p$ *belongs to a class 2 (i.e.* $o = y_2 = -1$*).*

Therefore, there is a need to define an *optimal canonical hyperplane* as a canonical hyperplane having *maximal margin*. This search for a separating, maximal margin, canonical hyperplane is the ultimate learning goal in statistical learning theory underlying SVMs. The notion of distance between a point and hyperplane is very useful and important.

In R^n let there be a given point $\mathbf{P}\left(x_{1p}, x_{2p}, ..., x_{np}\right)$ and a hyperplane $d(\boldsymbol{x}, \boldsymbol{w}, b) = 0$ defined by $w_1x_1 + w_2x_2 + \ldots + w_nx_n \pm b = 0$. The distance *D* from point ***P*** to hyprplane *d* is given as

$$D = \frac{\left|\left(\mathbf{w}\mathbf{x}_p\right) \pm b\right|}{\|\mathbf{w}\|} = \frac{\left|w_{1p}x_{1p} + w_{2p}x_{2p} + ... + w_nx_{np} = \pm b\right|}{\sqrt{w_1^2 + w_2^2 + ... + w_n^2}}. \tag{3.4}$$

At this point, we can consider an optimal canonical hyperplane, that is, a canonical hyperplane having a maximal margin. Then we can optimally separate a training data. Thus, in order to find the optimal separating hyperplane having a maximal margin, a learning machine should minimize $\|\mathbf{w}\|^2$ subject to inequality constraints:

$$y_i[\mathbf{w}^T\mathbf{x}_i + b] \geq 1, i = 1, \ldots, l \tag{3.5}$$

This is a conventional *nonlinear optimization problem with inequality constraints*. Such an optimization problem is solved by the saddle point of the *Lagrange* function:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{l} \alpha_i \left\{y_i\left[\mathbf{w}^T\mathbf{x}_i + b\right] - 1\right\}, \tag{3.6}$$

where $\alpha_i \geq 0$ are *Lagrange* multipliers. This function requires that the partial derivatives of **w** and *b* be zero. Partial derivatives propagate to constraints $\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i x_i$ and $\sum_{i=1}^{l} \alpha_i y_i = 0$.
Substituting **w** into (3.6) gives the dual form

$$L_d(\mathbf{w}, b, \alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i\alpha_j y_i y_j \left\langle x_i \cdot x_j \right\rangle, \tag{3.7}$$

which is not anymore an explicit function of **w** or *b*. The optimal hyperplane can be found by maximizing (3.7) subject to $\sum_{i=1}^{l} \alpha_i y_i = 0$; $\alpha_i \geq 0$.

B. The nonlinear classifier

For n-dimensional input patterns (described by n-dimensional vectors), instead of nonlinear curves, an SVM must be able to create nonlinear separating hypersurface. One basic idea in designing nonlinear SVMs is to map input vectors $\mathbf{x} \in \mathrm{R}^n$ into vectors **z** of higher – dimensional feature space *F* (***z*** = $\boldsymbol{\varphi}(x)$***,*** *where* $\boldsymbol{\varphi}$ *represents a mapping* $R^n \rightarrow R^f$). Kernel function K ($\mathbf{x}_\mathrm{i}$, $\mathbf{x}_\mathrm{j}$) is used for mapping into higher-dimensional space. More about nonlinear classification, kernel functions, multi-class classification and SVM algorithms you can find in [2], [3].

IV. Experiments

The LIBSVM software (Library for Support Vector Machines) [6] was used to realize all the experiments. The main task of these experiments was to understand the principles of classification based on LIBSVM Support Vector Machines. LIBSVM is a MATLAB toolbox for classification and regression training and testing data by SVM. It supports multi-class classification. The functions: C-support vector classification (C-SVC) [5], ν-support vector classification (ν-SVC) [7], distribution estimation (one-class-SVM) [8] were used for the classification. In our case we used the function of C – support vector classification, because by this function there was reached the highest classification accuracy.

In each experiment, we used RBF kernel function for mapping into higher-dimensional space with parameters *C* (penalty parameter) and *gamma* (RBF kernel parameter). In most cases, the C = 8 and *gamma* = 2. We had identified these two parameters by *5-fold cross-validation* function. All about these functions and parameters which includes software LIBSVM, and much more about these software you can find in [6], [9].

A. Collecting data

Firstly, we tried to find recordings of gunshots with good quality on the Internet. We found a server with sound recordings, which included shots of the various types of weapons, breaking glass, screams and others [11]. Poor quality of audio recordings was the main cause of obtaining our own gunshots recordings.

Recording the gun shots was carried out in the open space, in slightly windy weather. Individual shots were realized by gas gun, using 9 mm alarm ammunition. A total of 48 shots were fired, individual shots from distance 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 25 and 30 meters from the recorder. Recording device *Olympus LS-10* was used for recording various shots. The 4 shots were fired for each distance, the first shot separately and then three shots in a row (thus it was fired the first shot, then some-second pause and then three shots in a row). Recorder was turned on during the whole experiment, so the recording captures the weapons charge, walking on snow and blowing wind (background). All the shots are characterized by low reverberation time and high-intensity sound pressure level. The shots were realized at different distances, their intensity depending on the distance from the recorder (the larger distance from the device, the lower intensity of shot). Recording was performed with a maximum sample rate and number of bits per sample that this device provide: f_{vz}*=96 kHz*, N_{bit}*=32 bit/sample*.

B. Training and testing data

For training there were used the first 24 shots and for testing the remaining 24 shots (approx. 5 min. recording is divided approximately in half).

C. Experiment A

The task of the experiment was to determine the accuracy of prediction (classification) testing data based on using training model generated by training data on the same number of bits per sample, sound pressure level, at the same background and a different sampling frequency. In this experiment was used sampling rate: f_{vz2}*=48 kHz*, f_{vz3}*=32 kHz*, f_{vz4}*=16 kHz*, f_{vz5}*=8 kHz* and bits per sample N_{bit}*=32bits/sample* for training and testing data.

The aim of the experiment was to find the lowest sampling rate, that we can to use, i.e. where is the highest success rate of the classification of test data. The change of the sampling frequency was carried out using the software *Cool Edit Pro version 2.00*. For sampling frequency 8 kHz was used 256 points FFT, 16 kHz – 512 points FTT, 32 kHz – 1024 points FFT and for frequency 48 kHz – 2048 points FFT (in MFCC). The results of the experiment A are summarized in I.

The experiment A shows that the lowest possible sampling frequency that can be used to classify training and testing data at the highest classification accuracy is the frequency of 8000Hz. This could be caused by the upscaling on a lower frequency that occurs to remove components with higher frequencies. The number of samples which that there was classified thereby reduced.

D. Experiment B

The first 24 shots were used for a training and remaining 24 shots for a testing. Individual shots were mixed with background in the following way:

- clear training and clear testing data
- clear training and testing data with noise
- training data with noise and clear testing data
- training and testing data with noise

Blending shots with the background was made with Audacity 1.3 beta software in such a way, that the shots with sampling frequency 96 kHz and the number of bits per sample 32 was added recording from the bus station on the same frequency and the number of bits as it was in the shots. Results of experiment B are summarized in the Table II.

TABLE I
ACCURACY OF TRAINING DATA FOR DIFFERENT SAMPLE RATE OF TRAINING AND TESTING DATA

Testing data and sampling frequency	***Training model***	***The classification accuracy of testing data (%)***	***The number of correctly classified frames of test data (all13077)***
24gunshots_ test 8kHz32b	modelrbf24gunshots_train_8kHz32b	**99.89**	**13063**
24gunshots_ test 16kHz32b	modelrbf24gunshots_train_16kHz32b	**99.30**	**12986**
24gunshots_ test 32kHz32b	modelrbf24gunshots_train_32kHz32b	**98.78**	**12918**
24gunshots_ test 48kHz32b	modelrbf24gunshots_train_48kHz32b	**98.20**	**12842**

TABLE II
ACCURACY OF TRAINING AND TESTING DATA WITH OR WITHOUT BACKGROUND

Testing data with or without background	***Training model with or without background***	***The classification accuracy of testing data (%)***	***The number of correctly classified frames of test data (all13077)***
24gunshots_test	modelrbf24gunshots_train	**97.17**	**12707**
24gunshots_test_ background	modelrbf24gunshots_train	**1.14**	**150**
24gunshots_test	modelrbf24gunshots_train _background	**22.61**	**2958**
24gunshots_test_ background	modelrbf24vystrelov_train _background	**38.99**	**5100**

The task of the experiment B was to determine the impact of the background (unwanted noise) to classify the different training and testing sounds (data). The Table II shows that the mixed background causes drastic reduction in classification accuracy of testing data as well as the overall prediction accuracy of these data. The lowest percentage of successful classification occurred in the case when the training model was created using training data (gunshots) with no background and testing was carried out by test shots with mixed background (second row of the table). It follows that the high sound pressure of blending background compared to the test data is causing almost complete overload of data with noise and thus it is then difficult to separate useful sounds (in our case the gun shots) form the noise [11].

Both experiments are carried out using 24 MFC coefficients. When we used 12 MFC coefficients for feature extraction, the classification accuracy was in almost cases increased by only about *0.5* to *1%*, which in this case is the negligibly small increase. Binary classification with two classes {0 – silent, background, 1- shots} (labels of training and testing frames) (3.1) was used for each experiment.

V. CONCLUSION

The method of support vector machines was used to classify the gun shots in this paper. Specifically, this classification method we used to determine the lowest possible sampling frequency that can be used to classify training and testing data at the highest classification accuracy and to determine the impact of background (unwanted noise) to classify these data. Results from the Table I and II shows that the SVM method is a suitable and relatively precise algorithm for the classification of gun shots already on the sampling frequency 8 kHz, but the sound pressure of the background has a significant impact on classification accuracy.

ACKNOWLEDGMENT

This work is the result of the project implementation: Development of the Center of Information and Communication Technologies for Knowledge Systems (ITMS code: 26220120030) supported by the Research & Development Operational Program funded by the ERDF.

REFERENCES

[1] SAMUDRAVIJAYA, K.: *Gaussian Mixture Model and Hidden Markov Model*. [online], presentation, Tata Institute of Fundamental Research, Mumbai, 09-JAN-2009. Available on the Internet: [http://speech.tifr.res.in/tutorialSlides/gmmHmmTutoChief_wissap09.pdf].

[2] KECMAN, V.: *Learning and Soft Computing, Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. The MIT Press, Massachusetts Institute of Technology, 2001. ISBN 0-262-11255-8, pp.121-189.

[3] SCHOLKOPF, B. - SMOLA, A.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, release date: March 10, 2006, ISBN: 0262194759, pp.23-404.

[4] LIHAN, S. - JUHÁR, J. – ČIŽMÁR. A.: *Crosslingual and Bilingual Speech Recognition with Slovak and Czech Speech Dat-E Databases. Proceedings of the 9th European Conference on Speech Communication and Technology Interspeech 2005*, Lisbon, Portugal, 5-8 Sept., 2005,(ISSN 1018-4074), pp.225-228.

[5] FAGERLUND, S.: *Bird Species Recognition Using Support Vector Machines*, Research article, Hindawi Publishing Corporation, Helsinky University of Technology, Finland, EURASIP Journal on Advances in Signal Processing, 2007, pp. 3-5.

[6] CHIN-CHUNG Chang and CHIN-JEN Lin: Library for Support Vector machines.[online], Department of Computer Science, National Taiwan University, Taipei 106, Taiwan, February 27, 2009. Available on the Internet: [http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf].

[7] PAI-HSUEN, Chen – CHIH-JEN, Lin, and SCHOLKOPF, B: *A tutorial on ν-support vector machines*. Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan, 2003.

[8] RABAOUI, A. – KADRI, H. – LACHIRI, Z. - ELLOUZE, N.: *One-Class SVMs Challenges in Audio Detection and Classification Applications*. [online]. Research article, Hindawi Publishing Corporation, EURASIP Journal on Advances in Signal Processing, 2008, 14 p., Article ID 834973. Available on the Internet: [http://www.hindawi.com/journals/asp/2008/834973.html].

[9] MOORE, W., A.: *Cross-validation for detecting and preventing overfitting*. [online], presentation. 63 slides. Available on the Internet: [http://www.autonlab.org/tutorials/overfit.html].

[10] Server of sound recordings. Available on the Internet: [http://www.freesound.org/].

[11] VOZÁRIKOVÁ, E. – PLEVA, M. – VAVREK, J. – ONDÁŠ, S. – JUHÁR, J. – ČIŽMÁR, A.: *Detection and classification of audio events in noisy environment.* Journal of Computer Science and Control Systems (JCSCS) Vol. 3, No. 1, 2010, ISSN 1844-6043, to be published.