Support Vector Machines for Bass and Snare Drum Recognition

Dirk Van Steelant[1], Koen Tanghe[2], Sven Degroeve[3], Bernard De Baets[1], Marc Leman[2], and Jean-Pierre Martens[4]

[1] Department of Applied Mathematics, Biometrics and Process Control, Ghent University, Coupure links 653, 9000 Gent, Belgium
[2] Department of Musicology (IPEM), Ghent University, Belgium
[3] Department of Bioinformatics, Ghent University, Belgium
[4] Department of Electronics and Information Systems (ELIS), Ghent University, Belgium

Abstract. In this paper we attempt to extract information concerning percussive instruments from a musical audio signal. High-dimensional vectors of descriptors are computed from the signal and classified by means of Support Vector Machines (SVM). We investigate the performance on 2 important classes of drum sounds in Western popular music: bass and snare drums, possibly overlapping. The results are encouraging: SVM achieve a high accuracy and F_1-measure, with linear kernels performing (nearly) as good as Gaussian kernels, but requiring 1000 times less computation time.

1 Introduction

With the explosive growth of the amount of digital music available on the Internet, Musical Information Retrieval has become a topic that has attracted the attention of researchers in a wide range of disciplines. The quality of content-based retrieval however depends heavily on how well the individual components for representation and matching of the data perform. Most existing commercial music information retrieval systems use text as the main supplier of meta-data of music, such as the name of the artist/performer and the title of the song. For text, rapid matching methods are available and are applied extensively in search engines on the World Wide Web. However, as soon as such meta-data is incomplete or unavailable, all of the existing commercial systems will fail to deliver.

The MAMI-project (Musical Audio MIning) aims at working out methodologies and software tools for content-based audio-mining by bundling the efforts of musicologists, engineers, mathematicians and computer scientists. MAMI is centered on the 'query-by-imitation' paradigm, where users can retrieve a musical piece by means of its sound characteristics, either by describing, playing or vocally imitating the piece.

In order to supply a ranked list of candidate songs to the user, the system has to match an intermediate representation of the (melodic or rhythmic)

input with a similar representation of all the songs in the database; this will typically be done by means of a (time-consuming) dynamic programming technique. To speed up the query, any additional information that can narrow down the search space is welcome; not only meta-data, but also a description of the content of the target song or the musical genre to which it belongs can be used for this purpose.

A user study (Lesaffre et al. (2003)) has shown that when users are asked to imitate a song they are familiar with, some of them will reproduce the rhythmic structure of the piece. This is one of the motivations for analyzing the percussive content of musical audio; if a transcription can be obtained, it can be matched with the description delivered by the user, used as a feature for genre classification or provide valuable information for the determination of beat, tempo and rhythmic structure.

For the recognition of drum sounds three levels of difficulty can be distinguished: (i) Isolated drum sounds; (ii) Overlapping drum sounds; (iii) Overlapping drum sounds layered with other instruments and voices.

Obtaining a full transcription of the percussive content of musical audio is a challenging task and, to our best knowledge, has never been attempted using SVM. We will therefore concentrate on two important classes of sounds (omnipresent in Western popular music): bass drums (typically low-pitched and strongly indicating the beat) and snare drums (with highly noisy components, delivering important clues about the metrical structure of the song). In this paper we will concentrate on musical audio situated at the first and second level, since Virtanen (2001) has shown recently that it is possible to extract drum tracks from musical audio by subtracting the harmonic parts from the signal.

The rest of this paper is organized as follows. In Section 2 we give an overview of previous work. In Section 3 we describe how data were generated using samples gathered from commercial CD's, standard MIDI songs and sequencer software. In Section 4 relevant descriptors for audio data are presented and in Section 5 Support Vector Machines are formally introduced. In Section 6 we report results for two experiments and in Section 7 we comment on these results and give directions for future research.

2 Previous work

A recent overview of classification techniques for musical instrument sounds in general can be found in Herrera-Boyer et al. (2003). Percussive instruments represent a special case as they can be considered to be pitch-independent, so their appearance throughout a musical piece is much more constant. Although this makes them good candidates for localization/classification, they only represent a small part of previous research and in most cases only recognition of isolated sounds is investigated.

McDonald and Tsang (1997) use Spectral Centre Trajectories to classify percussive sounds but tests are only conducted on a very small database. In Zils et al. (2002) a percussion transcription is obtained by an analysis by synthesis technique, whereby the sound searched for is gradually synthesized from the signal; a success rate of over 75% is reported. A large-scale study in Herrera et al. (2003) uses different subsets of temporal and spectral descriptors (up to 207) for the recognition of thirty different classes of isolated percussion instruments. K-NN, Kernel Density (KD) estimation, canonical discriminant analysis and decision trees (C4.5) were investigated as classification techniques. KD combined with correlation-based feature selection yielded a 85% hit rate.

3 Data gathering

We have gathered samples belonging to two classes of percussive instruments (bass drum and snare drum) from commercial sample CD's. Such CD's typically indicate the class to which a sound belongs by the name of the sample or its location in the directory structure, but this information is not always equally reliable. Listening to the sounds we realized that some of them were mixed with other (percussive) instruments and therefore we had them classified by two users; only the samples that were considered to be "pure" and correctly classified by both users were retained. This yielded 656 bass drums and 604 snare drums; in all classes samples of the acoustic as well as of the electronic type were selected.

To gather realistic data, MIDI (Musical Instrument Digital Interface) files were exploited. Standard MIDI assigns classes of instruments to predefined tracks which makes it possible for an electronic sound device supporting standard MIDI to play songs with its own internal sounds. From 32 songs in standard MIDI format we selected 16 measures of the drum track. These 32 files were loaded into a sequencer program; 8 variations for each track were generated by selecting at random pairs of bass and snare drum from the set of samples while the other drum sounds were drawn from a standard MIDI drum set. The audio generated by playing back the MIDI files using these sets of drum sounds was recorded, yielding 256 audio files in total. The isolated drum sounds were added to the data set. This yielded a positive/negative example ratio of 1472/2508 for the bass drums and 1315/2729 for the snare drums. All files were saved as mono wave files sampled at 44.1 KHz.

In order not to introduce any errors due to the incorrect localization of events, we did not perform any onset detection but instead used the timing and labelling information available in the MIDI files to determine at what position in time descriptors need to be extracted and whether an event is a positive or negative instance for our binary classifiers. The information in the MIDI files thus represents the "ground truth" for the corresponding recorded audio renderings.

4 Descriptors for audio

Digital audio corresponds to a very high data rate (88 Kbyte/s for mono CD quality). To arrive at a manageable data rate, one needs to select descriptors that capture the characteristics of the audio while suppressing details that are redundant for the problem at hand. This data reduction will typically be done by sliding a window with a fixed step over the raw audio signal (e.g. a 20 ms window every 10 ms) and by computing at every step descriptors over that window.

The events we are trying to classify do not have a fixed length; the bass drums in our database for example have a duration ranging from 71 ms to 1.892 s. Although SVM are able to handle variable temporal representations by applying specific kernels, e.g. Shimodaira et al. (2001), determining the end of an event in musical audio (offset detection) is difficult. We therefore decided to use a fixed context at the beginning of the events over which descriptors are to be computed. In Section 6 we determine the most appropriate context length for each class. In order not to confuse the binary classifiers, we excluded any negative examples that lie within the range of 50 ms of a positive example.

A first set of descriptors concerns the energy in the signal computed by means of a Root Mean Square (RMS) formula. When inspecting the accumulated spectra of hundreds of bass drums and snare drums, it can be seen that the spectral energy distributions of these different sounds are located in more or less distinct frequency bands (although not completely separated). Hence we divided the spectrum into three frequency bands and computed energy-related descriptors over these bands: RMS in the whole signal, RMS per frequency band, ratio of RMS to overall RMS (per band) and RMS per band relative to RMS of other bands (1 to 2, 1 to 3 and 2 to 3).

Temporal descriptors are computed on the sample signal. The following descriptors were withheld: Zero Crossing Rate (ZCR): number of times per second the signal changes sign; Crest Factor: ratio of maximum absolute value sample signal to RMS in the segment; Temporal Centroid: the center of gravity of the distribution of the absolute values of the samples in the window. Spectral descriptors are computed using the Fast Fourier Transform, which converts the time domain data into the frequency domain: spectral centroid, skewness and kurtosis; and the spectral rolloff.

Logan (2000) shows that Mel Frequency Cepstral Coefficients (MFCC), short-term spectral-based features widely used for speech recognition, are appropriate as a representation for music by examining the functionality of a music/speech discriminator. MFCC are especially interesting for complex music analysis because they combine low-dimensionality and the ability to discriminate between different spectral content. The amount of detail in the description depends on the number of coefficients extracted; for our experiments 12 coefficients were computed. The temporal deployment of these descriptors is further captured by computing their first and second order derivatives. As a window size of 20 ms and frame step of 10 ms for the extrac-

tion of this kind of descriptors is often advised, we used these settings and we computed the mean and standard deviation of the coefficients and their first and second order derivatives over the context.

5 Support Vector Machines

Formally, a data set T contains l instances $\mathbf{x_i}$ $(i = 1, \ldots, l)$ with each $\mathbf{x_i}$ labelled as $y_i = 1$ or $y_i = -1$ (known as *classes*), indicating a positive or negative instance, respectively. Each index x_{ij} $(j = 1, \ldots, n)$ in vector $\mathbf{x_i}$ is a descriptor as described above.

The Support Vector Machine (Vapnik (1995)) is a data-driven method for solving two-class classification tasks. The Linear SVM (LSVM) separates the two classes in T with a hyperplane in the input space such that:

(a) the "largest" possible fraction of instances of the same class are on the same side of the hyperplane, and
(b) the distance of either class from the hyperplane is maximal.

The prediction of an LSVM for an unseen instance $\mathbf{z}$ is given by the decision function

$$pred(\mathbf{z}) = \mathrm{sgn}(\mathbf{w} \cdot \mathbf{z} + b)\,. \tag{1}$$

The hyperplane is computed by means of a vector of Lagrange multipliers α maximizing

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i\, \alpha_j\, y_i\, y_j\, (\mathbf{x_i} \cdot \mathbf{x_j})\,,$$

subject to:

$$0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^{l} \alpha_i\, y_i = 0\,, \tag{2}$$

where C is a parameter set by the user to regulate the effect of outliers and noise, i.e. it defines the meaning of the word "largest" in (a). Some tolerance (denoted as ϵ) on the constraints in Equation 2 is acceptable.

A function K (called a kernel function) maps the descriptors in T, called the input space, into a feature space defined by K in which then a linear class separation is performed. For the LSVM this mapping is a linear mapping:

$$K(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i} \cdot \mathbf{x_j}\,. \tag{3}$$

The non-linear mapping used in this paper is the Gaussian-SVM (GSVM)

$$K(\mathbf{x_i}, \mathbf{x_j}) = e^{-|\mathbf{x_i} - \mathbf{x_j}|^2/\gamma^2}\,. \tag{4}$$

After calculating the α_i's in (2), the decision function (1) becomes:

$$pred(\mathbf{z}) = \text{sgn}(\sum_{i=1}^{l} \alpha_i \, y_i \, K(\mathbf{x_i}, \mathbf{z}) + b) \, . \tag{5}$$

An instance x_i for which α_i is not zero is called a Support Vector (SV). Note that the prediction calculated in (5) uses the support vectors only. As such, the support vectors are those instances that are closest to the decision boundary in the feature space.

All SVM in our experiments were trained using SVM$^{\text{light}}$ 5.0 (Joachims (1999)[1]) in classification mode with all parameters at their default values, except for C and the kernel-related parameter γ. The data were scaled so that every descriptor lies within the range $[-1, 1]$.

6 Experiments and results

In order to determine an appropriate context length for the two classes of drum sounds we computed the descriptors over various lengths (50, 70, 100, 140, 170 and 200 ms) and performed 3-Fold Cross Validation (3-FCV) using LSVM with $C = 2^i$ $(i = -8, \ldots, 0, \ldots, 10)$. As a performance measure we combined the obtained average precision and recall into

$$F_\beta = \frac{(\beta^2 + 1) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall}$$

with β a user-controlled parameter expressing the preference for either high precision or high recall ($\beta = 1$ in the sequel). Table 1 shows the best F_1 for various context lengths. For both bass drum and snare drum the best performance was obtained using a context length of 100 ms.

Context (ms)	50	70	100	140	170	200
F_1 BD	93.91	95.11	**95.15**	94.94	94.59	94.64
F_1 SD	97.39	97.69	**98.18**	97.61	97.30	96.58

Table 1. F_1 with 3-FCV on the whole data set for different context lengths

Using the obtained context lengths, we investigated the difference in performance between a linear and (the more powerful) Gaussian kernel. It needs to be pointed out that there is no guarantee that the optimal context length for LSVM is also optimal for GSVM; ongoing research will have to clarify this point.

[1] http://svmlight.joachims.org/

The data were split into a 87.5% training set (for model selection and training) and a 12.5% test set (while respecting the balance between positive and negative examples). We had the optimal parameters for Gaussian kernels (C, γ) established by *looms* (Lee and Lin (2000)[2]) which estimates the leave-one-out error rate over a grid of candidate values using a loose stopping criterion in the optimization phase. For LSVM we obtained the optimal C using 3-FCV on the training set and F_1 as performance measure. The results in Table 2 also contain overall accuracy and the number of support vectors for the obtained models. These results show a very minor difference in performance for BD and no improvement at all for SD; despite bigger computational effort for model selection and the fact that the resulting model is more complex (the number of support vectors has almost doubled), exactly the same misclassifications are done with the Gaussian kernel as with the linear one.

	LSVM		GSVM	
	BD	SD	BD	SD
C	0.5	8	0.25	8
γ	-	-	0.256	0.064
accuracy	94.98	97.63	95.58	97.63
F_1	93.30	96.34	94.12	96.34
#SV	391	179	965	350

Table 2. Classification of the 12.5% test set using LSVM (model selection by 3-FCV) and GSVM (model selection by estimating leave-one-out error).

7 Conclusions and future work

The results show that our audio descriptors and SVM classifiers combine well into a technique for the recognition of drum sounds in an audio signal. We expect that the methodology can be extended for the detection of a wider range of percussive instruments.

The observation that linear kernels perform only slightly worse than the Gaussian ones is an important finding for applications in time-critical environments. An LSVM with approximately 1300 support vectors classifies 5000 examples (89-dimensional) in less than 10 ms while it takes a GSVM with the same number of SV close to 10 s (done on a mobile Pentium III 1.2 GHz with 256 DDR RAM); this difference could turn out to be crucial in a real-time system that, besides classification, also needs to perform onset detection and compute appropriate descriptors.

[2] http://www.csie.ntu.edu.tw/~cjlin/looms/

As there is a vast amount of candidate descriptors for the modelling of audio and various ways of encoding them, future research should try to extend the set of descriptors and at the same time, for the sake of simplicity, reduce it by means of variable selection methods (e.g. as in Degroeve et al. (2002)). Findings related to what kind of descriptors are relevant for the recognition of percussion would also provide interesting feedback to researchers in the field of musicology and perceptual psychology.

Acknowledgements

This research is funded by the Flemish Institute for the Promotion of Scientific and Technical Research in Industry.

References

DEGROEVE, S., DE BAETS, B., VAN DE PEER, Y. and ROUZE, P. (2002): Feature Subset Selection for Splice Site Prediction. *Bioinformatics, 18, 75–83.*

HERRERA, P., DEHAMEL, A. and GOUYON, F. (2003): Automatic Labeling of Unpitched Percussion Sounds. In: *Proc. 114th Convention of the Audio Engineering Society, Amsterdam, The Netherlands.*

HERRERA-BOYER, P., PEETERS, G. and DUBNOV, S. (2003): Automatic Classification of Musical Instrument Sounds. *Journal of New Music Research, 32, 3–21.*

JOACHIMS, T. (1999): Making Large-scale SVM Learning Practical, In: B. Schölkopf, C. Burges and A. Smola (Eds.): *Advances in Kernel Methods - Support Vector Learning.* MIT Press.

LEE, J.-H. and LIN, C.-J. (2000): Automatic Model Selection for Support Vector Machines. Technical report, Dept. of Computer Science and Information Engineering, National Taiwan University.

LESAFFRE, M., MOELANTS, D., LEMAN, M., DE BAETS, B., DE MEYER, H., MARTENS, G. and MARTENS, J.-P. (2003): User Behavior in the Spontaneous Reproduction of Musical Pieces by Vocal Query. In: *Proc. 5th Triennial ESCOM Conference, Hannover, Germany.*

LOGAN, B. (2000): Mel Frequency Cepstral Coefficients for Music Modelling. In: *Proc. Internat. Symposium on Music Information Retrieval, Plymouth, MA, USA.* 23–25.

MCDONALD, S. and TSANG, C.P. (1997): Percussive Sound Identification using Spectral Centre Trajectories. Technical Report Departmental Conference, Yanchep.

SHIMODAIRA, H., NOMA, K., NAKAI, K. and SAGAYAMA, S. (2001): Support Vector Machine with Dynamic Time-alignment Kernel for Speech Recognition. In: *Proc. Eurospeech, Aalborg, Denmark.*

VAPNIK, V.N. (1995): *The Nature of Statistical Learning Theory.* Springer-Verlag.

VIRTANEN, T. (2001): Audio Signal Modeling with Sinusoids plus Noise. MSc thesis, Tampere University of Technology.

ZILS, A., PACHET, F., DELERUE, O. and GOUYON, F. (2002): Automatic Extraction of Drum Tracks from Polyphonic Music Signals. In: *Proc. 2nd Internat. Conference on Web Delivering of Music, Darmstadt, Germany.*