A Simulated Annealing Optimization of Audio Features for Drum Classification

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

[1]Department of Applied Mathematics, Biometrics and Process Control, Ghent University, Belgium
[2]Department of Musicology (IPEM), Ghent University, Belgium
[3]Department of Electronics and Information Systems (ELIS), Ghent University, Belgium

Sven.Degroeve@UGent.be

ABSTRACT

Current methods for the accurate recognition of instruments within music are based on discriminative data descriptors. These are features of the music fragment that capture the characteristics of the audio and suppress details that are redundant for the problem at hand. The extraction of such features from an audio signal requires the user to set certain parameters. We propose a method for optimizing the parameters for a particular task on the basis of the Simulated Annealing algorithm and Support Vector Machine classification. We show that using an optimized set of audio features improves the recognition accuracy of drum sounds in music fragments.

Keywords: drum classification, Mel Frequency Cepstral Coefficients, Support Vector Machine, Simulated Annealing

1 INTRODUCTION

With the tremendous growth of the amount of digital music available either locally or remotely through networks, Music Information Retrieval (MIR) has become a topic that has attracted the attention of researchers in a wide range of disciplines. An important part of MIR research is concerned with automatic methods for (musical) audio content description.

Automatic localization and classification of the percussive content of musical audio can be employed in various ways. The recognition of isolated drum sounds would be beneficiary for the organization of sample libraries while the more challenging task of transcribing mixtures of percussive sounds (such as drum loops, break beats or complete drum tracks) can assist in the process of music production. If the percussive content of complete songs (containing other instruments, voices and audio effects) can be analyzed, this information can be used for the determination of beat/tempo and genre/style.

Digital audio corresponds to a very high data rate, e.g. 88 Kbyte/s for mono CD quality. Current instrument recognition methods require the extraction of discriminative data descriptors, known as features. These features represent one specific property of the signal. They capture the characteristics of the audio and suppress details that are redundant for the problem at hand.

One set of features that is known to provide valuable information for the recognition of music are the Mel Frequency Cepstral Coefficients (MFCC). They were shown to be appropriate for e.g. music/speech classification (Logan, 2000; Tzanetakis and Cook, 2002; West and Cox, 2004). They are also interesting for complex music analysis because they combine low-dimensionality and the ability to discriminate between different spectral contents. Recent studies in which MFCC features are compared to other signal representations have shown the potential of MFCC features for speaker/sound recognition and audio segmentation (McKinney and Breebaart, 2004; Kim and Sikora, 2004).

The extraction of MFCC features from an audio signal requires the user to set certain parameters such as the length of the windows used to extract the information. A detailed description of these parameters is given in Section 3.1. Until now people have used values for these parameters that seem intuitively acceptable. In this research we investigate the impact of optimizing these parameter values for the recognition of drum sounds using a Linear Support Vector Machine algorithm.

The rest of this paper is organized as follows. Section 2 describes the drum data sets used in the experiments. The optimization procedure we propose is based on the Simulated Annealing (SA) algorithm (Kirkpatrick et al., 1983) and is described in Section 3. Section 4 presents experimental results demonstrating the influence of an optimized set of feature parameter values on the task of drum recognition.

2 DATA

The musical data used for this paper was collected from various commercial music CDs, mostly from popular genres. We chose to use ‘real, fully produced music’ because that is exactly the type of music that will be handled by music information retrieval systems, which is the application domain of our research. We selected 49 fragments of 30 seconds length each in 16 bit 44100 Hz stereo PCM WAV format and asked experienced drummers to annotate the drums in these fragments (Tanghe et al., 2005). Annotation involved localizing and labeling all the drum events that are present in the fragments. This was done through a combination of live playing on a MIDI keyboard and visual editing/correcting in a standard audio and MIDI sequencer. During the annotations, 18 different drum types were available, but for this paper we have reduced them to the 5 main types: bass drums (BD), snare drums (SD), hihats (HH), cymbals (CY) and toms (TM). 25 music fragments were randomly selected as a training set and the other 24 fragments as a test set. The total number of instances for each drum in each set is shown in the first three columns of Table 1.

3 METHOD

The recognition of drum sounds in an audio signal can be formulated as a context classification task. For each drum sound a separate data set is created from the annotated music fragments, i.e. the task of recognizing the drums in an audio signal is reduced to the recognition of each drum sound individually. Each data set contains one feature vector for each annotated onset. A feature vector (known as an instance) is computed from the signal properties found in the neighbourhood of the onset. The true classification of an instance (known as the label) depends on the task. For instance, for the task of recognizing BD a data set is created in which all onsets annotated as BD are represented by an instance labeled as positive, and all other onsets are represented by instances labeled as negative. If more than one drum sound is annotated for the same onset then this onset is used only once in the data set: as a positive instance if one of the annotated drums is the drum sound that needs to be recognized, negative otherwise. So, for each task the data set contains 9848 feature vectors (known as instances) where each instance has one label: positive or negative.

An inductive learning algorithm is adapted to create a drum classification system. Given that the data and the inductive learning method are fixed, the classification performance of this classification system can be used as an estimation of how suitable a certain feature vector representation is for the classification of drums.

3.1 Feature Vector Representation

For each annotated drum event a feature vector is extracted from the signal properties found in the neighbourhood of that event. This neighbourhood is defined as an interval of length p_1 (measured in milliseconds) starting at the onset. We will refer to the signal in this interval as the context of the onset. For each drum sound, all features will be computed from this context. As such, the length p_1 influences the value of all the features described below and it is the first parameter that requires optimization.

The amplitude of the onset context is described by means of a Root Mean Square (RMS) formula. When inspecting the accumulated spectra of hundreds of bass drums, snare drums and hihats, it can be seen that the spectral energy distributions of these sounds are located in more or less distinct frequency bands (although not completely separated). Hence we divide the spectrum into three frequency bands and compute energy-related features over these bands: RMS in the whole signal; RMS per frequency band; ratio RMS per band to overall RMS; and RMS per filter band relative to RMS of other bands (1 to 2, 1 to 3 and 2 to 3).

The temporal nature of the onset context is described by the following features: Zero Crossing Rate (ZCR): number of times per second the signal changes sign; Crest Factor: ratio of the maximum signal amplitude value to the RMS of the signal; and Temporal Centroid: the center of gravity of the power values of the samples in the segment.

The spectral features of the onset context are computed from a Fast Fourier Transform (FFT) computed on the whole onset context signal. The following features are added to the feature vector: Spectral Centroid: the centre of gravity of the power spectrum; Spectral Skewness: the third order central moment of the power spectrum; Spectral Kurtosis: the fourth order central moment of the power spectrum; Spectral Flatness: the ratio of the geometric mean to the arithmetic mean of the power spectrum; and Spectral Rolloff: The value R such that

$$\sum_{i=1}^{R} P[f_i] = 0.85 \sum_{i=1}^{N} P[f_i]$$

where $P[f_i]$ is the power value for the frequency at bin i and N is the number of frequency bins.

Another set of features are the MFCCs and their derivatives. The MFCCs are derived from a sequence of fixed length audio frames; the first frame starts at the drum onset t_0, and the last one ends at $t_0 + p_1$, and consecutive frames are shifted over a fixed length frame step. For each frame the MFCCs are calculated using the following FFT-based method: apply a Hanning window to the audio frame, perform an FFT, apply a triangular shaped Mel filter bank to the FFT bin values and sum the results per band, (optionally) apply the log operator to the filter outputs and finally apply a Discrete Cosine Transform (DCT). In order to capture the temporal changes of the MFCCs, we also calculate their first and second order deltas. The features we considered are the mean and standard deviation for each of these values over the whole onset context.

Several parameters come into play when calculating these MFCC-related features, and together with the above mentioned context length p_1, these are the parameters that were varied during the optimization (see also Table 2). Parameter p_2 specifies the width of the audio frames (in milliseconds) for which the spectrum is calculated, parameter p_3 specifies the frame step (in milliseconds) and parameter p_4 specifies the size of the FFT. For the Mel filter bank, the number of filters can be chosen (parameter p_5) and it is possible to normalize the FFT bin weights so that the total weight is the same for each filter (parameter p_6). Furthermore, parameter p_7 specifies whether the logarithm of each filter band output should be taken or not, and parameter p_8 specifies the number of coefficients that should

Table 1: Data set statistics and baselines. For each drum classification task the table shows the number of instances in the training set (25 music fragments) in the second column and the number of instances in the test set (24 music fragments) in the third column. For both the format is (positives/negatives). The fourth column shows the precision ratio for a baseline classifier that classifies all instances in the test set as positive (recall=1). The fifth column shows the FN5% ratio obtained on the test set using an LSVM trained on the training set with all features except for the MFCC features ($p_1 = 0.1$). For the fifth column results shown in bold indicate a statistical signifcant difference as compared to the baseline method. The last column shows the FN5% ratio obtained on the test set using an LSVM trained on the training set with all features extracted using default parameter settings as shown in Table 2. For the last column results in bold indicate a statistical significant difference as compared to not using the MFCC features (column five).

drum	train	test	baseline	no MFCC	default MFCC
BD	972/3334	1230/4310	22.2	49.9	52.0
SD	563/3743	919/4621	16.6	23.0	**28.4**
CY	42/4264	164/5376	2.9	3.2	**5.9**
HH	1656/2650	2128/3412	38.4	54.0	**55.2**
TM	123/4183	81/5459	1.5	–	**–**

be kept after the DCT. Then finally, for the calculation of the derivatives, parameter p_9 specifies the type of delta, and parameter p_{10} the window size (in number of frames) over which the deltas are calculated (see (Young et al.) for detailed info).

3.2 Linear Support Vector Machines

The drum classification system is a Linear Support Vector Machine (LSVM), trained by means of the inductive learning algorithm that is explained in Section 3.2 (Boser et al., 1992; Vapnik, 1995). The LSVM has been shown to perform well for the task of classifying BD and SD drum sounds (Van Steelant et al., 2004). The LSVM separates the two classes in a data set D using a hyperplane such that:

(a) the "largest" possible fraction of instances of the same class is on the same side of the hyperplane, and

(b) the distance of either class from the hyperplane is maximal.

The algorithm has a parameter C that needs to be set by the user and regulates the effect of outliers and noise, i.e. it defines the meaning of the word "largest" in (a).

For the induction of the LSVM we used SVM$^{\text{light}}$ 5.0 (Joachims, 1999)[1] in classification mode with all parameters at their default values, except for the cost parameter C, which will be optimized from the data.

3.3 Measure of Classification Performance

The classification performance of an LSVM on a data set D is measured in terms of recall and precision. Recall quantifies the proportion of positive vectors that are classified correctly while precision quantifies the proportion of positive classifications that are correct. Both are required to address the performance of the classification system. To allow for an automated optimization procedure, we quantify the performance of an LSVM by its precision at a 95% recall, i.e. we allow for only 5% false negative classifications. The precision measure is referred to as the FN5% ratio (5% False Negatives).

Given the data set D the classification performance of an LSVM induced classification system is computed using 10-fold cross-validation which divides D into ten partitions, uses each partition in turn as a test set and the other nine as the training set, and which computes evaluation criteria as averages over the ten test sets. The partitions we use are equally sized, and have the same class distribution.

McNemar's test is applied to decide whether any apparent difference in error rates between two algorithms (feature vector representations in our experiments) tested on the same set of music fragments is statistically significant (Dietterich, 1998). The test uses those classifications that are correct for only one of the algorithms (n_{01} and n_{10}). Let h_0 be the hypothesis that the underlying error rates are the same. Then under h_0 an error is as likely to be made by either of the two algorithms and the distribution of n_{01} and n_{10} is the distribution obtained when tossing a fair coin. This is a binomial distribution and the p-values are easily obtained from tables. Results that have a p-value greater than 0.05 are not statistically significant.

3.4 Optimization Strategy

Simulated Annealing (SA) is an optimization algorithm based on a Metropolis Monte Carlo simulation. The goal is to find a parameter setting s_{min} in the space of all candidate settings S for which a real-valued energy function E is minimized. The algorithm performs a series of random walks through S at successively lower temperatures T, where the probability P of making a step from s_{old} to s_{new} is given by a Boltzmann distribution:

$$P = \begin{cases} 1 & \text{if } \Delta E \leq 0 \\ e^{-\frac{\Delta E}{T}} & \text{otherwise} \end{cases} \tag{1}$$

with

$$\Delta E = E(s_{new}) - E(s_{old}). \tag{2}$$

The function $E(s)$ in our optimization strategy quantifies the classification error of an LSVM induced in a feature space defined by s. To avoid features in greater numerical ranges to dominate those in smaller ranges, the data is scaled such that every feature lies within the range of $[-1, 1]$. Let the function $10CV(s, c)$ return the FN5%

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

Table 2: Feature vector parameters. For each parameter p_i in the first column the table shows the default value in the second column. The third column shows the values considered during the optimization. The fourth column provides a short description of the parameters. The last five columns show optimized parameter settings for each of the drum sounds.

Parameter	Default	Values	Description	BD	SD	CY	HH	TM
p_1	0.1	grid search	onset context length	0.1	0.1		0.14	–
p_2	0.02	$\{0.01, 0.02, 0.03\}$	FFT window length	0.03	0.03		0.03	–
p_3	0.01	$\{0.005, 0.01, 0.015\}$	FFT window interval	0.01	0.005		0.01	–
p_4	1024	$\{1024, 2048, 4096\}$	FFTsize	2048	1024		4096	–
p_5	40	$[5, 40]$	number of filters	37	31		34	–
p_6	1	$\{0, 1\}$	normalize	1	0		0	–
p_7	1	$\{0, 1\}$	logarithm	1	0		0	–
p_8	13	$[1, p_5]$	number of coefficients	16	22		30	–
p_9	0	$\{0, 1\}$	delta type	0	0		0	–
p_{10}	2	$\{1, 2, 3, 4\}$	delta window length	4	3		3	–

Table 3: Optimal classification performance for all SA procedures. Each row represents one of the five drum classification tasks. The columns represent values for p_1 (in milliseconds). All results (FN5% ratio) are obtained on the test set. Results in bold represent the best performing context lengths for the training set.

	0.06	0.08	0.10	0.12	0.14	0.16	0.18	0.20	0.22	0.24	0.26
BD	47.9	–	**51.6**	–	51.4	–	–	–	–	–	–
SD	27.0	–	**27.8**	–	25.7	–	–	–	–	–	–
CY	4.1	–	6.4	5.5	6.0	–	5.9	8.9	8.9	**8.9**	6.1
HH	55.3	–	55.2	55.5	**54.8**	55.0	–	–	–	–	–
TM	–	–	–	–	–	–	–	–	–	–	–

ratio of a 10-fold cross-validation procedure in a feature space defined by s using an LSVM with cost $C = c$, then the energy $E(s)$ is computed as:

$$E(s) = 1 - \max_c 10CV(s, c). \qquad (3)$$

We chose to optimize C within each computation of E, as opposed to adding C to the set of parameters (Table 2) that are optimized within the SA procedure. The reason is that most of the computation time required for computing E is spent at the extraction of the features from the music fragments, and not at the cross-validation procedure. As such, optimizing C within the energy function can be done at little additional computation cost.

This computation cost is further reduced by parallelizing the SA search using a grid search based on the onset context length p_1. This is done by performing three SA optimizations (optimizing all parameters except p_1), one for $p_1 = 0.06$, one for $p_1 = 0.1$, and one for $p_1 = 0.14$. If $p_1 = v$ is the best performing context length, then two other SA optimizations are performed for $p_1 = v - 0.02$ and $p_1 = v + 0.02$. This process is repeated until no further improvement is observed. The SA procedure was implemented as follows:

```
(1) initialize $s_{old}$
(2) $T$ = 0.043
(3) repeat 100 times
(4)     repeat 10 times
(5)         $s_{new}$ = $step(s_{old})$
(6)         $s_{old}$ = $s_{new}$ with probability P
(7)     $T = \frac{T}{1.05}$
```

The probability P in (6) depends on T and is computed as in Eq. (1). The procedure $step$ in (5) randomly selects one of the features from $\{p_2, \ldots, p_{10}\}$ and changes the value of this parameter at random. For the parameters p_5 and p_8 the random increase or decrease was limited to 5 to keep the step local.

It is known that choosing proper values for the various SA parameters is hard and depends very much on the characteristics of the energy landscape defined by S. Initializing $T = 0.043$ means that during the first iterations a step is taken with $P = 0.5$ when $\Delta E = 0.03$. As the computational cost for computing the energy function is high, the number of SA iterations is limited to 100, which results in a total of 1000 energy computations for each drum data set and each context length p_1 considered during the grid search.

4 RESULTS

All LSVM classifiers in the following experiments are induced from the training set. The results shown in the tables are obtained on the test set. LSVM parameter C was optimized using 10-fold cross-validation on the training set. The significance of the differences between the results is computed using McNemar's test.

In a first experiment we evaluated the importance of MFCC features for the task of drum classification. We evaluated the classification performance of an LSVM computed from feature sets including and excluding the MFCC features with default parameter settings $(p_1 = 0.1)$. These are values often used in literature and are shown in column 3 of Table 2 (Logan, 2000). The performance was measured as the FN5% ratio. The results were compared to a baseline classifier that classifies all instances as positive, i.e. recall = 1. These results are shown in the last three columns of Table 1. For HH and TM, the baseline method does not perform worse than the LSVM classifier that uses all but the MFCC features. Adding the MFCC features improves the classification significantly for SD: the number of of false drum classifications (false positives) decreased by 21.5%. Also for HH and TM we observe minor improvements.

Next, the SA optimization method was executed on the 25 music fragments in the training set. The procedure was run for each of the five drum classification tasks. This resulted in an optimal parameter setting for each context length p_1 tested during the grid search procedure (described in Section 3.4) for each of the five drum sounds. We then tested these optimal parameter settings on the test set. This was done for each drum sound by computing the FN5% test set classification performance of an LSVM induced from the training set using the optimal parameter settings for that drum sound. Table 3 shows the optimal FN5% ratio for each onset context length p_1 and each drum sound. For each row one result is shown in bold. The column associated with the result in bold represents the context length p_1 that showed the best 10-fold cross-validation performance on the training set at the end of each SA procedure. For BD, SD and CY the SA procedure finds parameter settings that perform significantly better than the default ones that are shown in Table 2. For BD the number of false drum classification decreased by 28.5%, for CY this was 24.4%. Also, for BD and CY we observe that when using optimized audio features, the addition of MFCC features does actually improve classification performance. This was not the case when using the default feature parameter settings. If we use Table 3 to evaluate the final solutions found by the SA optimization procedure, then we notice that the procedure failed to find the best parameter settings for most of the tasks.

Figure 1 summarizes the SA procedure for the best performing (on the training set) context lengths. The optimal parameter settings for each of the data sets are shown in the last five columns of Table 2. From Figure 1 we conclude that, given the limitation of 100 iterations for each SA run, the SA algorithm converges nicely at around 750 energy evaluations. The figure also indicates that, next to p_1, other parameters have an impact on the drum classification performance of the LSVM. If we look at the optimal parameter settings in Table 2 we notice that the boolean parameter p_7 has the same optimal value for all five drum sounds. Also p_2 shows a stable behaviour as the length of the FFT windows is preferred to be larger. All other parameters can differ significantly between the tasks.

Next, we evaluated the impact of each optimal parameter value on the classification performance individually. For each p_i $(i = 2 \ldots 10)$ we quantified the drum classification performance using the optimal parameter values found by the SA, but with p_i set to its default value if this was not already the case. Again all training was done on the training set and the FN5% ratios shown in the Table 4 are computed on the test set. We observe that slight changes (setting parameters back to their default values) in the optimal parameter settings can improve the classification of SD, CY and TM further. The SA optimization procedure proposed in this paper does not necessarily find the optimal solution, but it does find good sub-optimal solutions that improve drum classification accuracy.

Table 4: Impact of setting each parameter p_i to its default value (Table 2) in the optimal parameter setting for each of the drum sounds. The results shown are the FN5% ratios obtained using 10-fold cross-validation on the test set. Results in bold show differences that are statistically significant $(p < 0.05)$ as compared to using the optimal parameter settings found by the SA procedure. The symbol '-' means that this parameter was already set to its default value.

	BD	SD	CY	HH	TM
p_2	**49.4**	**26.4**	9.1	54.6	-
p_3	-	27.3	**9.1**	-	-
p_4	51.3	**26.9**	-	54.7	-
p_5	**51.1**	-	**9.5**	54.6	-
p_6	-	**24.9**	8.9	54.6	-
p_7	-	27.1	-	**54.4**	-
p_8	**52.9**	27.2	8.7	54.6	-
p_9	-	-	-	-	-
p_{10}	**50.7**	27.8	-	54.9	-

In a final experiment we wanted to evaluate to what extent the (sub-)optimal parameter settings for each drum sound are specific to this drum classification task. As the SA algorithm does not find the optimal solution, there might be more than one sub-optimal solution. This means that the parameter settings found for a certain drum sound might also work well for other drum classification tasks. To check this, we compared optimal parameter settings between the different drum tasks. For each pair of drum sounds (X,Y) we evaluated the classification performance on the task of recognizing drum X with an LSVM induced using the optimal parameter settings of drum sound Y. Performance was again measured as the FN5% ratio computed on the test set. Table 5 shows the results with drum sound X as rows and drum sound Y as columns. For all

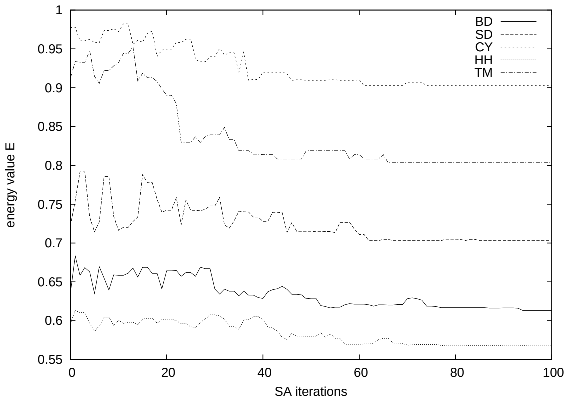

Figure 1: Summary of the SA procedure. The x-axis represents the number of SA iterations. It shows the energy of the current parameter setting every time the temperature is decreased in step (4) of the SA procedure. The y-axis shows the energy of the current parameter setting.

Table 5: A comparison of different optimal parameter settings for the classification of different drum sounds. Each row represents a drum classification task. Values then represent the FN5% performance ratios obtained using the optimal parameter settings for the drum sounds in the associated columns.

	BD	SD	CY	HH	TM
BD	31.9	**32.9**	**33.0**	**33.2**	**32.9**
SD	**20.2**	22.1	22.3	**22.6**	22.2
CY	4.1	**3.4**	4.0	**3.7**	**4.4**
HH	38.6	**39.2**	**38.0**	38.4	**39.1**
TM	**1.5**	1.6	**1.7**	1.6	1.6

tasks we observe statistically significant differences between the different sub-optimal parameter settings. But these differences are not all in favour of the SA optimization procedure. For instance, for CY we find that using the TM optimal parameter settings increases performance to 4.4%, as compared to 4.0% when using the CY optimal parameters found using the SA procedure. When using these optimized audio features in a real situation in which all instruments in an audio signal need to be recognized, other performance criteria arise. Using different feature extraction parameter settings for each of the instruments that need to be classified increases the computation cost for analyzing the signal significantly. The optimal parameter setting for the TM classification task (last column of Table 5) performs well for all five tasks. Using this setting for all five drum sounds would drastically reduce the computation cost.

5 CONCLUSIONS

In this study we evaluated the importance of using optimized audio features for drum classification in music fragments. We proposed an optimization procedure based on Simulated Annealing and we have shown that the feature parameters evaluated in the research have a statistically significant impact on the drum sound classification performance. We believe that this should be taken into account when comparing the use of different types of audio features. When using non-optimized features, the results might not fully capture the true discriminative potential.

We have also shown that the optimization procedure proposed in this research does not always find the optimal solution. Better optimization algorithms should be investigated, but the optimization task is hard because of the long computation time required to evaluate one audio feature parameter setting.

ACKNOWLEDGEMENTS

This work was done in the context of the Musical Audio Mining (MAMI) project, which is funded by the Flemish Institute for the Promotion of Scientific and Technological Research in Industry. The authors wish to thank Micheline Lesaffre, Liesbeth De Voogdt and Dirk Van Steelant for their contribution to the creation of the music fragments data set.

References

B.E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proc. of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, 1992.

T.G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1924, 1998.

T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 169–185. MIT Press, 1999.

H.-G. Kim and T. Sikora. Comparison of MPEG-7 audio spectrum projection features and MFCC applied to speaker recognition, sound classification and audio segmentation. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004)*, Montreal, Canada, 2004.

S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, (2):671–680, 1983.

B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proc. of the Internatonal Conference on Music Information Retrieval (ISMIR 2000)*, Plymouth MA, 2000.

M.F. McKinney and J. Breebaart. Features for audio and music classification. In *Proc. of the Internatonal Conference on Music Information Retrieval (ISMIR 2004)*, pages 151–158, Plymouth MA, 2004.

K. Tanghe, M. Lesaffre, S. Degroeve, M. Leman, B. De Baets, and J.-P. Martens. Collecting ground truth annotations for drum detection in polyphonic music. In *Proc. of the Internatonal Conference on Music Information Retrieval*, London, 2005.

G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

D. Van Steelant, K. Tanghe, S. Degroeve, B. De Baets, M. Leman, and J.-P. Martens. Classification of percussive sounds using support vector machines. In *Proc. of the Annual Machine Learning Conference of Belgium and The Netherlands (BENELEARN)*, pages 146–152, Brussels, 2004.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

K. West and S. Cox. Features and classifiers for the automatic classification of audio signals. In *Proc. of the Internatonal Conference on Music Information Retrieval (ISMIR 2004)*, Barcelona, 2004.

S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book*. http://htk.eng.cam.ac.uk/.