USING BLOCK-LEVEL FEATURES FOR GENRE CLASSIFICATION, TAG CLASSIFICATION AND MUSIC SIMILARITY ESTIMATION

Klaus Seyerlehner, Markus Schedl, Tim Pohle, Peter Knees
Dept. of Computational Perception, Johannes Kepler University, Linz, Austria

ABSTRACT

In our submission we use a set of block-level features for three different tasks, namely genre classification, tag classification and music similarity estimation. This abstract presents the feature set that is used and some specific details of the three submitted algorithms.

1. INTRODUCTION

In our system we extract the same set of block-level features for all three tasks. The feature extraction is implemented in MATLAB. All three algorithms also contain a classification part, which is based on the WEKA machine learning toolbox [**?**]. Thus, we first briefly introduce the features set in section **??** and then in the subsequent sections we discuss the most important algorithmic details of the submitted algorithms (genre classification, tag classification and music similarity estimation).

2. BLOCK-LEVEL FEATURES

A detailed introduction to the block processing **framework** can be found in [**?**, **?**, **?**]. Here we only present the specific details of the extraction process for the individual block-level features.

2.1 Audio Preprocessing

All block-level features presented here are based on the same spectral representation: the cent-scaled magnitude spectrum. To obtain this, the input signal is downsampled to 22 kHz and transformed to the frequency domain by applying a Short Time Fourier Transform (STFT) using a window size of 2048 samples, a hop size of 512 samples and a Hanning window. Then we compute the magnitude spectrum $|X(f)|$ thereof and account for the musical nature of the audio signals by mapping the magnitude spectrum with linear frequency resolution onto the logarithmic Cent scale [**?**] given by Equation (**??**).

$$f_{\text{cent}} = 1200 \log_2(f_{\text{Hz}}/(440 * (\sqrt[1200]{2})^{-5700})) \quad (1)$$

The compressed magnitude spectrum $X(k)$ is then transformed according to Eq.**??** to obtain a logarithmic scale. Altogether, the mapping onto the Cent scale is a fast approximation of a constant-Q transform, but with constant window size for all frequency bins.

$$X(k)_{dB} = 20 \log_{10}(X(k)) \quad (2)$$

Finally, to make the obtained spectrum intensity-invariant, we normalize it by removing the mean computed over a sliding window from each audio frame as described in [?]. All features presented in the next section are based on the normalized cent spectrum. Note that the reported parameter settings for the audio features in the following subsections were obtained via optimization with respect to a set of genre classification experiments.

2.2 Spectral Pattern (SP)

To characterize the frequency or timbral content of each song we take short blocks of the cent spectrum containing 10 frames. A hop size of 5 frames is used. Then we simply sort each frequency band of the block.

$$\text{SP} = \begin{bmatrix} \text{sort}(b_{H,1} & \cdots & b_{H,W}) \\ \vdots & \ddots & \vdots \\ \text{sort}(b_{1,1} & \cdots & b_{1,W}) \end{bmatrix} \quad (3)$$

As summarization function the 0.9 percentile is used.

2.3 Delta Spectral Pattern (DSP)

The Delta Spectral Pattern is extracted by computing the difference between the original cent spectrum and a copy of the spectrum delayed by 3 frames, to emphasize onsets. The resulting delta spectrum is rectified so that only positive values are kept. Then we proceed exactly as for the Spectral Pattern and sort each frequency band of a block. A block size of 25 frames and a hop size of 5 frames are used, and the 0.9 percentile serves as summarization function. It is important to note that the DSP's block size differs from the block size of the SP; both were obtained via optimization. Consequently, the SP and the DSP capture information over different time spans.

2.4 Variance Delta Spectral Pattern (VDSP)

The feature extraction process of the Variance Delta Spectral Pattern is the same as for the Delta Spectral Pattern (DSP). The only difference is that the Variance is used as

summarization function over the individual feature dimensions. While the Delta Spectral Pattern (DSP) tries to capture the strength of onsets, the VDSP should indicate if the strength of the onsets varies over time or, to be more precise, over the individual blocks. A hop size of 5 and a block size of 25 frames are used.

2.5 Logarithmic Fluctuation Pattern (LFP)

To represent the rhythmic structure of a song we extract the Logarithmic Fluctuation Patterns, a modified version of the Fluctuation Pattern proposed by Pampalk et al. [**?**]. A block size of 512 and a hop size of 128 are used. We take the FFT for each frequency band of the block to extract the periodicities in each band. We only keep the amplitude modulations up to 600 bpm. The amplitude modulation coefficients are weighted based on the psychoacoustic model of the fluctuation strength according to the original approach in [?]. To represent the extracted rhythm pattern in a more tempo invariant way, we then follow the idea in [**?**, **?**, **?**] and represent periodicity in log scale instead of linear scale. Finally, we blur the resulting pattern with a Gaussian filter, but for the frequency dimension only. The summarization function is the 0.6 percentile.

2.6 Correlation Pattern (CP)

To extract the Correlation Pattern the frequency resolution is first reduced to 52 bands. This was found to be useful by optimization and also reduces the dimensionality of the resulting pattern. Then we compute the pairwise linear correlation coefficient (Pearson Correlation) between each pair of frequency bands, which gives a symmetric correlation matrix. The basic idea of using band intercorrelation as a frame-level audio descriptor has already been proposed by Aylon [**?**]. Within the block-processing framework the correlation matrix is computed on block-level, which is computationally efficient, and a song-level descriptor is derived via using the 0.5 percentile as summarization function. The Correlation Pattern can capture, for example, harmonic relations of frequency bands when sustained musical tones are present. Also rhythmic relations can be reflected by the CP. For example, if a bass drum is always hit simultaneously with a high-hat this would result in a strong positive correlation between low and high frequency bands. Visualizations of the CP show interesting patterns for different types of songs. For example the presence of a singing voice leads to very specific correlation patterns, which is even more obvious for the CP computed from time-frequency representations with higher frequency resolutions. A block size of 256 frames and a hop size of 128 frames is used.

2.7 Spectral Contrast Pattern (SCP)

The Spectral Contrast [**?**] is a feature that roughly estimates the *"tone-ness"* of a spectral frame. This is realized by computing the difference between spectral peaks and valleys in several sub-bands. As strong spectral peaks roughly correspond to tonal components and flat spectral

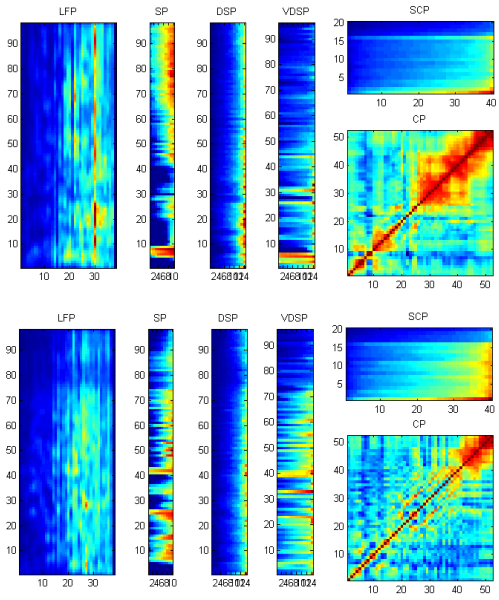

Figure 1. *Visualization of the proposed block-level patterns for a Hip-Hop song (upper) and a Jazz song (lower).*

excerpts are often related to noise-like or percussive elements, the difference between peaks and valleys characterizes the toneness in each sub-band. In our implementation the Spectral Contrast is computed from a cent scaled spectrum subdivided into 20 frequency bands. For each audio frame, we compute in each band the difference between the maximum value and the minimum value of the frequency bins within the band. This results in 20 Spectral Contrast values per frame. The values pertaining to an entire block are then sorted within each frequency band, as already described for the SP above. A block size of 40 frames and a hop size of 20 frames are used. The summarization function is the 0.1 percentile.

Figure **??** visualizes the proposed set of features for two different songs, a Hip-Hop and a Jazz song. It is worth mentioning that beside the introduced block-level features the **Onset Patterns**, another variant of the fluctuation patterns proposed by Pohle et al. [?], are used for music similarity estimation.

3. GENRE CLASSIFICATION

The genre classification approach itself is rather straight forward. The presented block-level features are combined into a single feature vector that is then used for classification. As classification approach the WEKA support vector machine classifier (SMO) is used. Comparing the results to our MIREX 2009 submission we obtain improved classification results on some well-known datasets (see table **??**). The additional block-level features improve the classification accuracy.

4. AUTOMATIC TAG PREDICTION

In general tag prediction can be viewed as a simple extension of the genre classification approach from single

Reference	Dataset	Accuracy
Tazanetakis et al. [**?**]	GTZAN	61.00%
Holzapfel et al. [**?**]	GTZAN	74.00%
Lidy et al. [**?**]	GTZAN	76.80%
Seyerlehner et al. [?]	GTZAN	77.96%
Panagakis et al. [**?**]	GTZAN	78.20%
Li. et al. [**?**]	GTZAN	78.50%
Bergstra et. al. [**?**]	GTZAN	83.00%
MIREX 2010 Submission	GTZAN	85.49%
Panagakis et al. [**?**]	GTZAN	92.40 %
Panagakis et al. [?]	ISMIR2004all	80.95%
Lidy et al. [?]	ISMIR2004all	81.40%
Seyerlehner et al. [?]	ISMIR2004all	83.72%
MIREX 2010 Submission	ISMIR2004all	88.27%
Pohle et al. [?]	ISMIR2004all	90.04%
Panagakis et al. [?]	ISMIR2004all	94.38 %
Holzapfel et al. [?]	Ballroom	86.90%
Jensen et al. [?]	Ballroom	89.00%
Pohle et al. [?]	Ballroom	89.20%
MIREX 2010 Submission	Ballroom	92.44%

Table 1. *Comparison of classification accuracies achieved by music genre classification approaches.*

to multi-label classification. In tag classification there is, instead of a single classifier like in genre classification, one classifier per tag. Unfortunately, we cannot directly use the block-level features as presented in section **??** for tag classification because of the high dimensionality (9448 dimensions) of the feature space. Directly using the features block-level features would not be computationally tractable and most likely break the allowed MIREX runtime limit for tag classification algorithms. To solve this issues the dimensionality of each block-level feature is reduced using the Principal Component Analysis (PCA) as proposed in [?]. The number of principal components (N) used to represent each block-level feature is determined by the cumulative variance that they capture. In our implementation N is chosen such that at least 80% of the total variance is captured. It is important to note that in our submission we do not use cross validation to estimate the optimal percentage of the total variance, but we have rather set this parameter according to other dataset of approximately the same size.

While in [?] our tag classification approach was based on the MARSYAS framework, in our submission we decided to use WEKA instead to be platform independent. An important parameter that has an influence on the evaluation metrics is the binarization threshold. To generate binary predictions based on the probabilistic output of a tag classifier, typically a binarization threshold $t = 0.5$ is chosen. However, the threshold can also be set lower or higher to trade between precision and recall. To maximize the average f-Score $t = 0.25$ was chosen based on a set of tag classification experiments (see figure **??**). We do not use the adaptive binarization threshold as proposed in [**?**], because using this approach no predictions/classification can be made for individual tracks, but only for sets of music pieces. Furthermore, it is important to note that we do not use *stacked classification* [?] in our submission as during our experimentation with tag classification we found that in some cases the stacked generalization classification approach can also have a negative influence on the classification result.

5. MUSIC SIMILARITY ESTIMATION

Our music similarity estimation approach is based on two distinct components: *Block-Level Feature Similarity* and *Tag Affinity Based Similarity*. The following two subsections present the algorithmic details of these two components.

5.1 Block-Level Feature Similarity

To directly estimate music similarity based on the presented block-level features we follow the approach presented in [?]. First, pairwise song similarities are estimated by computing the Manhattan distance for each of the presented block-level features separately. Then in a second step the individual distance matrices resulting from the individual patterns are combined into a single distance matrix. The main problem when several components are combined into a single similarity estimate is that the individual components potentially measure distances on different scales. To account for this, we follow the approach in [**?**] and perform a distance space normalization (DSN). However, we use a modified variant that allows for a more intuitive interpretation. Each distance of a distance matrix $D_{n,m}$ is normalized by subtracting the mean and dividing by the standard deviation (Gaussian normalization) over all distances in row n and column m (see figure **??**). Thus, each distance between two songs n and m has its own normalization parameters, as all distances to song m and all distances to song n are used for normalization. This way the normalization operation can also change the ordering within a column / row, which can even have a positive influences on the nearest neighbor classification accuracy according to [?].

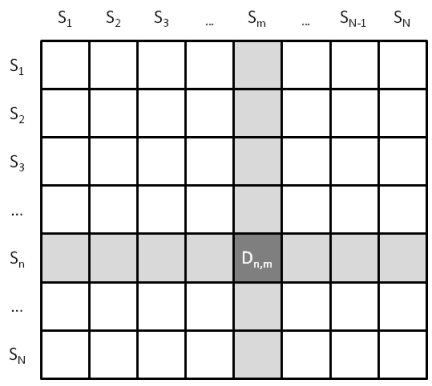

Figure 3. *Distance Space Normalization of a distance matrix.*

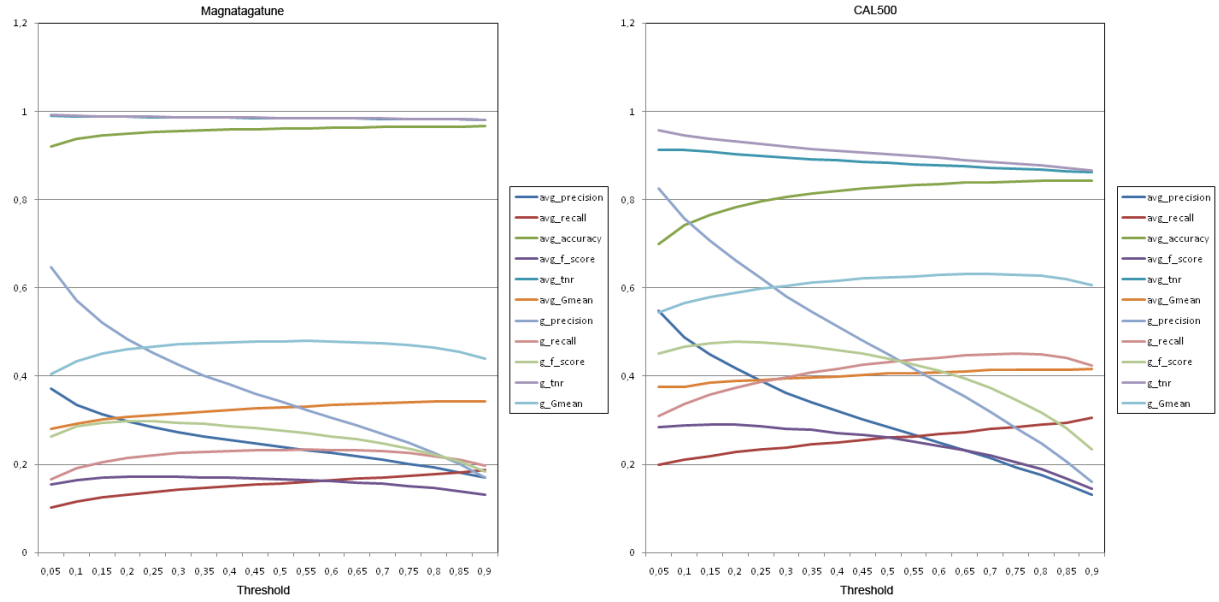

Figure 2. *Evaluation of different binarization thresholds for per tag averaged (avg) and global (g) evaluation metrics.*

After the distances resulting from the individual components have been normalized they are summed to yield an overall similarity measure. It is worth mentioning that we have evaluated several other normalization approaches to combine the individual components, but none outperformed the DSN approach described here.

5.1.1 Extended Distance Space Normalization (EDSN)

It is straightforward to extend this additive combination approach by assigning individual weights to each component. Additionally, based on the observation that the DSN approach all alone can help to improve nearest neighbor classification accuracy, we extend the combination approach and once more normalize the resulting distance matrix after the individually weighted components have been combined. This results in an improvement of the classification accuracy. The overall combination method we propose is visualized in Figure **??**. The only difference to the approach presented in [?] is that an additional pattern, the *Onset Pattern* [?], is used. The weights used for the combination are the same as reported in [?]. For the additionally added Onset Patterns a weight of 1 is used.

5.2 Tag Affinity Based Similarity

To compute tag affinity based similarity we use the output of a set of pre-trained support vector machine classifiers with probabilistic outputs as proposed in [**?**, **?**]. These classifiers were trained on well-known datasets. Seven different genre classification datasets were used to learn genre affinities. For each genre classification dataset we obtain on classifier that is used to estimate one genre affinity vector per dataset. The *CAL500* [**?**] and the *Magnatagatune* [**?**] dataset are used to train two different sets of tag predictors. Optimal PCA compression parameters for both datasets were chosen according to our experiments in [?].

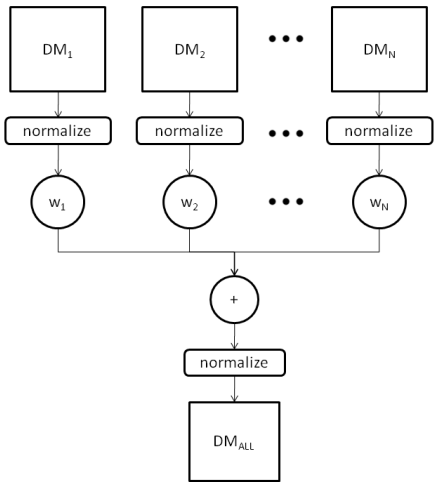

Figure 4. *Schematic structure of the combination approach.*

For each tag of the these classification datasets we obtain one classifier that is used to predict the probability of a tag for a specific song. Altogether, 371 pre-trained classifiers are used to estimate such predefined labels in our approach.

To come up with similarity estimates we compute the Manhattan distances between two songs' tag or genre affinity vectors separately for each dataset. Then the individual distances resulting from the different datasets are combined into a single distance matrix according to the extended distance space normalization approach (see **??**).

Finally, to generate the overall similarity matrix the matrices of both components (Block-Level Similarity and Tag Affinity Based Similarity) are simply added to combine them.

6. ACKNOWLEDGMENTS

This research was supported by the Austrian Research Fund (FWF) under grant L511-N15.

7. REFERENCES

[1] E. Aylon. Automatic detection and classification of drum kit sounds. Master's thesis, Universitat Pompeu Fabra, 2006.

[2] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl. Aggregate features and adaboost for music classification. *Machine Learning*, 2006.

[3] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere. Autotagger: A model for predicting social tags from acoustic features on large music databases. *Journal of New Music Research*, 2008.

[4] M. Goto. Smartmusickiosk: Music listening station with chorus-search function. In *Proc. of the 16th ACM Symp. on User Interface Software and Technology (UIST-03)*, 2003.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 2009.

[6] A. Holzapfel and Y. Stylianou. Musical genre classification using nonnegative matrix factorization-based features. *IEEE Transactions on Audio, Speech, and Language Processing (TASLP-08)*, 2008.

[7] A. Holzapfel and Y. Stylianou. A scale transform based method for rhythmic similarity of music. In *Proc. of the 2009 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP-09)*, 2009.

[8] J. H. Jensen, M. G. Christensen, and S.H Jensen. A tempo-insensitve representation of rhythmic patterns. In *Proc. of the 17th European Signal Processing Conf. (EUSIPCO-09)*, 2009.

[9] D. Jiang, L. Lu, H. Zhang, J. Tao, and L. Cai. Music type classification by spectral contrast feature. In *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME-02)*, 2002.

[10] E. Law and L. Ahn. Input-agreement: A new mechanism for collecting data using human computation games. In *Proc. of the 27th Int. Conf. on Human Factors in Computing Systems (CHI-09)*, 2009.

[11] T. Li, M. Ogihara, and Q. Li. A comparative study on content-based music genre classification. In *Proc. of the 26th ACM SIGIR Conf. on Research and Development in Informaion Retrieval*, 2003.

[12] T. Lidy, A. Rauber, A. Pertusa, and M. Inesta. Improving genre classification by combination of audio and symbolic descriptors using a transcription system. In *Proc. of the 8th International Conference on Music Information Retrieval (ISMIR-07)*, 2007.

[13] S. R. Ness, A. Theocharis, G. Tzanetakis, and L.G. Martins. Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs. In *Proc. of the 17th ACM Int. Conf. on Multimedia (MM -09)*, 2009.

[14] E. Pampalk, A. Rauber, and D. Merkl. Content-based organization and visualization of music archives. In *Proc. of the 10th ACM Int. Conf. on Multimedia*, 2002.

[15] I. Panagakis, E. Benetos, and C. Kotropoulos. Music genre classification: A multilinear approach. In *Proc. of the 9th International Conference on Music Information Retrieval (ISMIR-08)*, 2008.

[16] Y. Panagakis, C. Kotropoulos, and G.R. Arce. Music genre classification using locality preserving non-negative tensor factorization and sparse representations. In *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, 2009.

[17] T. Pohle and D. Schnitzer. Striving for an improved audio similarity measure. In *online Proc. of the 4th Annual Music Information Retrieval eXchange (MIREX-07)*, 2007.

[18] T. Pohle, D. Schnitzer, M. Schedl, P. Knees, and G. Widmer. On rhythm and general music similarity. In *Proc. of the 10th International Society for Music Information Retrieval Conference (ISMIR-09)*, 2009.

[19] K. Seyerlehner and M. Schedl. Block-level audio feature for music genre classification. In *online Proc. of the 5th Annual Music Information Retrieval Evaluation eXchange (MIREX-09)*, 2009.

[20] K. Seyerlehner, G. Widmer, and T. Pohle. Fusing block-level features for music similarity estimation. In *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, 2010.

[21] K. Seyerlehner, G. Widmer, M. Schedl, and P. Knees. Automatic music tag classification based on block-level features. In *Proc. of the 7th Sound and Music Computing Conference*, 2010.

[22] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech, and Language Processing*, 2008.

[23] G. Tzanetakis and P. Cook. Musical genre classification of audio signal. *IEEE Transactions on Audio and Speech Processing*, 2002.

[24] K. West, S. Cox, and P. Lamere. Incorporating machine-learning into music similarity estimation. In *Proc. of the 1st ACM Workshop on Audio and Music Computing Multimedia (AMCMM-06)*, 2006.