Unsupervised learning of low-level audio features for music similarity estimation

Christian Osendorfer OSENDORF@IN.TUM.DE
TU München, Informatik VI, Germany

Jan Schlüter JAN.SCHLUETER@OFAI.AT
Austrian Research Institute for Artificial Intelligence (OFAI), Vienna

Jürgen Schmidhuber JUERGEN@IDSIA.CH
IDSIA, USI & SUPSI, Switzerland

Patrick van der Smagt SMAGT@DLR.DE
DLR, Institute of Robotics and Mechatronics, Germany

Abstract

While there is an enormous amount of music data available, the field of music analysis almost exclusively uses manually designed features. In this work we learn features from music data in a completely unsupervised way and evaluate them on a musical genre classification task. We achieve results very close to state-of-the-art performance which relies on highly hand-tuned feature extractors.

1. Introduction

We consider the problem of content-based music similarity estimation (MSE). A typical MSE system has two parts for modeling music (see (Seyerlehner et al., 2010) for more details): (a) extracting features from the audio data, and (b) aggregating these features into a global description for every music piece. While in (b) sometimes simple machine learning algorithms are applied, (a) usually is exclusively hand-designed (a few notable exceptions are (Lee et al., 2009; Hoffman et al., 2009; Hamel & Eck, 2010)). In our work, we substitute these hand-designed feature extractors with a mean-covariance Restricted Boltzmann Machine (mcRBM, (Ranzato & Hinton, 2010)), an unsupervised learning algorithm that shows excellent performance for various image and speech modelling tasks (Ranzato & Hinton, 2010; Dahl et al., 2010).

2. The mean-covariance RBM

An mcRBM is a generative model that defines a probability distribution over its input variables $\mathbf{v}$ and two groups of binary hidden units: mean units $\mathbf{h}_m$ and precision units $\mathbf{h}_c$. Without the precision units, the mcRBM would be a Gaussian-Bernoulli RBM guided by the following energy function:

$$E_m(\mathbf{v}, \mathbf{h}_m) = \frac{1}{2}(\mathbf{v}-\mathbf{b})^T(\mathbf{v}-\mathbf{b}) - \mathbf{c}^T\mathbf{h}_m - \mathbf{v}^T\mathbf{W}\mathbf{h}_m$$

Here $\mathbf{W}$ denotes the matrix of visible/hidden connection weights and $\mathbf{b}/\mathbf{c}$ the visible/hidden bias[1]. With only the precision units this model is a particular type of factored third-order Boltzmann machine (Ranzato et al., 2010). The energy function of this *cRBM* can be written as:

$$E_c(\mathbf{v}, \mathbf{h}_c) = -\mathbf{d}^T\mathbf{h}_c - (\mathbf{v}^T\mathbf{R})^2\mathbf{P}\mathbf{h}_c$$

$\mathbf{R}$ is the visible-factor weight matrix, $\mathbf{P}$ the factor-hidden pooling matrix, and $\mathbf{d}$ the hidden bias vector. If we add the two energy functions, we obtain the energy function of the mcRBM: $E_{mc} = E_m + E_c$. The resulting conditional distribution for the visible units is

$$P(\mathbf{v}|\mathbf{h}_m, \mathbf{h}_c) \propto \mathcal{N}(\mathbf{\Sigma}\mathbf{W}\mathbf{h}_\mathrm{m}, \mathbf{\Sigma})$$

with $\mathbf{\Sigma} = (\mathbf{R}(\mathrm{diag}(-\mathbf{P}^T\mathbf{h}_c))\mathbf{R}^T)^{-1}$. The conditional distribution for the mean units is the same as in a standard RBM,

$$P(\mathbf{h}_m|\mathbf{v}) = \sigma(\mathbf{b} + \mathbf{W}^T\mathbf{v}),$$

[1] We are closely following (Dahl et al., 2010), Section 3. For a detailed account of mcRBMs, see (Ranzato & Hinton, 2010).

Table 1. k-NN classification accuracies of various methods on the two datasets for $k = 20$.

METHOD	1517-ARTISTS	HOMBURG
RANDOM BASELINE	4.7	22.6
MVG-MFCC	25.6	48.8
CMB	41.1	61.2
OUR WORK	35.0	55.3

while for the precision units it is

$$P(\mathbf{h}_c|\mathbf{v}) = \sigma(\mathbf{d} + ((\mathbf{v}^T\mathbf{R})^2\mathbf{P})^T).$$

So once an mcRBM has been trained, inferring the latent representation for a given data vector is computationally cheap enough to use the model for large-scale feature extraction. See (Ranzato et al., 2010) and (Ranzato & Hinton, 2010) for details on the training procedure.

3. Experiments

Ground truth data for evaluating music similarity measures is hard to obtain. However, *music genre classification* has been shown to be a good proxy for music similarity estimation, allowing comparison of different methods (Pohle, 2010).

3.1. Datasets

We use two different genre classification datasets, *1517-Artists* (Seyerlehner et al., 2010) and *Homburg* (Homburg et al., 2005). Artist information is available for both datasets, so we can use an artist filter to eliminate any artist or album effect. 1517-Artists encompasses 3,180 tracks by 1,517 different artists, distributed almost uniformly over 19 genres. The Homburg dataset contains 1,886 songs by 1,463 different artists. The short song excerpts (10 seconds long) are unequally distributed over 9 genres, the largest class contains 26.7%, the smallest class 2.5% of all songs.

3.2. Preprocessing

For all our experiments we preprocessed the acoustic signal similar to (Dahl et al., 2010): The signal is divided into *frames* of 64 ms length, with successive frames having an overlap of 32 ms. Each frame is represented by the 40-log magnitudes of a mel filter bank. 39 Consecutive frames form a *block* that is whitened with PCA and truncated to the 310 most important principal components, retaining 99% of the total variance.

3.3. Training Details

On these blocks, we train an mcRBM with 2,500 factors, 625 covariance hidden units and 512 mean hidden units. $\mathbf{P}$ is initialized to a topography over the filter outputs of matrix $\mathbf{R}$, enforcing similarity between neighbouring filters. Similar results are obtained with 2d and 1d topographies; below, we only report results for a 2d topography. $\mathbf{P}$ is not updated for the first 11 epochs to let the filters converge first, and then masked to retain the topography, only allowing changes to the weights of already nonzero factor/hidden connections. All other parameters for learning the model are those that can be found in the example configuration of the code accompanying (Ranzato & Hinton, 2010).

3.4. Feature aggregation

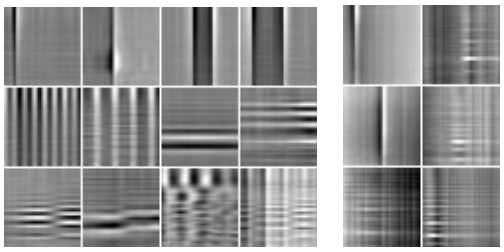
Given the trained mcRBM, we infer latent representations for all blocks of a music piece. A global descriptor for a piece is built by adding up all its block representations and normalizing this vector such that it sums to 1. Distances between different music pieces are determined via the L1 norm. Following (Seyerlehner et al., 2010), we also apply a distance space normalization to the distance matrix.

3.5. Results

We performed the genre classification experiments with a k Nearest Neighbour (k-NN) classifier, with k ranging from 1 to 20. In Table 1 we compare previously published results for both datasets with our work. MVG-MFCC (Mandel & Ellis, 2005) represents songs as a multivariate Gaussian of MFCCs, a short-term spectral feature borrowed from speech processing. CMB (Seyerlehner et al., 2010) constructs a representation for a song through a weighted combination of more than ten hand-crafted feature extractors. On the two datasets, CMB represents the current state of the art for MSE. Additionally, we visualize some typical examples of the learned filters in Figure 1. The filters seem to capture different musical concepts, such as note onsets. See the caption for more details.

4. Conclusion

We used a recently proposed unsupervised learning algorithm to find useful features of music data. By leveraging a large amount of unlabeled data our learned features achieve results nearly as good as state-of-art algorithms, which are hand-crafted and finely tuned to music data. In order to improve our results, we are currently investigating hierarchical deep architectures, mainly with a focus on better modeling of rhythms.



(a) (b)

Figure 1. Exemplary filters of an mcRBM trained on music excerpts. Each block represents 1,248 ms of a spectrogram. Time increases from left to right, mel-frequency from bottom to top, bright and dark indicate positive and negative values, respectively. On the left side (a) we see filters learned by the covariance part of the mcRBM. In reading order, we show two examples each for recurring themes: note onsets, fixed-length notes, repetitive percussion, chords, note transitions, and mixed tones. On the right side (b) we show filters learned by the mean part of the mcRBM. These filters only develop as onset detectors or mixed tones.

References

Dahl, G., Ranzato, M.A., Mohamed, A., and Hinton, G.E. Phone recognition with the mean-covariance restricted boltzmann machine. In *Proc. NIPS*, 2010.

Hamel, P. and Eck, D. Learning features from music audio with deep belief networks. In *Proc. ISMIR*, 2010.

Hoffman, M.D., Blei, D.M., and Cook, P.R. Finding latent sources in recorded music with a shift-invariant hdp. In *Proc. DAFx*, 2009.

Homburg, H., Mierswa, I., Möller, B., Morik, K., and Wurst, M. A benchmark dataset for audio classification and clustering. In *Proc. ISMIR*, 2005.

Lee, H., Largman, Y, Pham, P., and Ng, A. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Proc. NIPS*, 2009.

Mandel, M. and Ellis, D. Song-level features and support vector machines for music classification. In *Proc. ISMIR*, 2005.

Pohle, T. *Automatic Characterization of Music for Intuitive Retrieval*. PhD thesis, Johannes Kepler University, Linz, 2010.

Ranzato, M.A. and Hinton, G. Modeling pixel means and covariances using factorized third-order boltzmann machines. In *Proc. CVPR*, 2010.

Ranzato, M.A., Krizhevsky, A., and Hinton, G. Factored 3-way restricted boltzmann machines for modeling natural images. In *Proc. AISTATS*, 2010.

Seyerlehner, K., Widmer, G., and Pohle, T. Fusing block-level features for music similarity estimation. In *Proc. DAFx*, 2010.