

©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Appearing in: Proceedings of the Tenth International Conference on Machine Learning and Applications (ICMLA 2011). In press.

Music Similarity Estimation with the Mean-Covariance Restricted Boltzmann Machine

Jan Schlüter*[†]

[†]Austrian Research Institute for Artificial Intelligence (OFAI)

Vienna, Austria

jan.schlueter@ofai.at

Christian Osendorfer*

*Technische Universität München

Munich, Germany

osendorf@in.tum.de

Abstract—Existing content-based music similarity estimation methods largely build on complex hand-crafted feature extractors, which are difficult to engineer. As an alternative, unsupervised machine learning allows to learn features empirically from data. We train a recently proposed model, the mean-covariance Restricted Boltzmann Machine [1], on music spectrogram excerpts and employ it for music similarity estimation. In k-NN based genre retrieval experiments on three datasets, it clearly outperforms MFCC-based methods, beats simple unsupervised feature extraction using k-Means and comes close to the state-of-the-art. This shows that unsupervised feature extraction poses a viable alternative to engineered features.

I. INTRODUCTION

Fostered by the advancement of digital technologies, both catalogs of music distributors and personal music collections have grown to sizes that call for automated methods to manage them. Classification algorithms, for instance, help grouping music according to a given taxonomy. Here we consider the more difficult task of estimating perceived music similarity, which may be used to recommend music based on examples or to generate well-sounding playlists. In particular, we are interested in content-based methods that rely on sound only, not requiring any metadata.

Existing content-based music similarity estimation systems usually compare music in three stages: (a) extracting local features from the audio signals, (b) aggregating these features into a global descriptor for each song and (c) calculating the distance between global models. While machine learning techniques have been employed for global modelling (e.g., GMMs and HMMs [2] or HDPs [3]) and metric optimization [4], feature extraction is mostly hand-crafted. Supplementing generic low-level audio features such as MFCCs [5], state-of-the-art systems rely on elaborate transformations of spectrograms, tuned to capture different musical aspects (e.g., fluctuation patterns [6], onset patterns [7] and block-level features [8]). An alternative to such knowledge engineering is empirical induction [9] by unsupervised machine learning, which has been shown to yield feature extractors similar to mammals' primary visual cortex [10] and cochlear [11] and thus might also learn what humans perceive in music [12]. A recently proposed method for unsupervised feature extraction from images, the mean-covariance Restricted Boltzmann Machine (mcRBM) [1], has successfully been applied to spectrogram excerpts to model speech [13], but never to music.

In this work, we build a music similarity estimation system based on an mcRBM. Section II briefly reviews former attempts at unsupervised music feature extraction, Section III details our approach and introduces the mcRBM, Section IV evaluates our system on three genre-labeled datasets, comparing it to baseline methods and the state-of-the-art, and Section V gives a conclusion and outlook on future work.

II. RELATED WORK

In literature, unsupervised learning on music has mostly been focusing on source separation. As an early example, Abdallah [12] showed that sparse coding of harpsichord music spectrograms could reveal notes. Hoffmann et al. [14] train a shift-invariant HDP on spectrograms and yield decompositions of songs into musically meaningful components such as drum sounds and vocals, as well as transcriptions of songs in terms of these components. While this demonstrates the power of unsupervised learning, such complex song descriptions are not directly usable for similarity estimation.

RBMs have been applied to music, but only for supervised classification: Lee et al. [15] train stacked convolutional RBMs on frame-wisely PCA-compressed spectrograms, yielding features that surpass MFCCs in 5-way genre and 4-way artist classification (not stating the method used). Hamel et al. [16] train a stack of RBMs on spectral frames, extensively fine-tune the network using genre labels and use it to extract frame-wise features. Afterwards, they train a non-linear SVM to classify these features into the genres they have been tuned for. Classifying songs with a winner-takes-all scheme, they report improved accuracy over MFCCs. Both results are interesting for genre classification, but it remains unclear if and how such features can form a genre-independent, song-level similarity measure. Moreover, both approaches compete with MFCCs, whereas we tackle music-specific state-of-the-art features.

Pohle et al. [17] are closest to our work. They applied ICA to PCA-compressed mel-sone music spectrogram excerpts and compared the components' activation histograms for music pieces to estimate their musical similarity. However, in 1-NN genre classification this approach proved inferior even to the classic MFCC-based approach of [2].

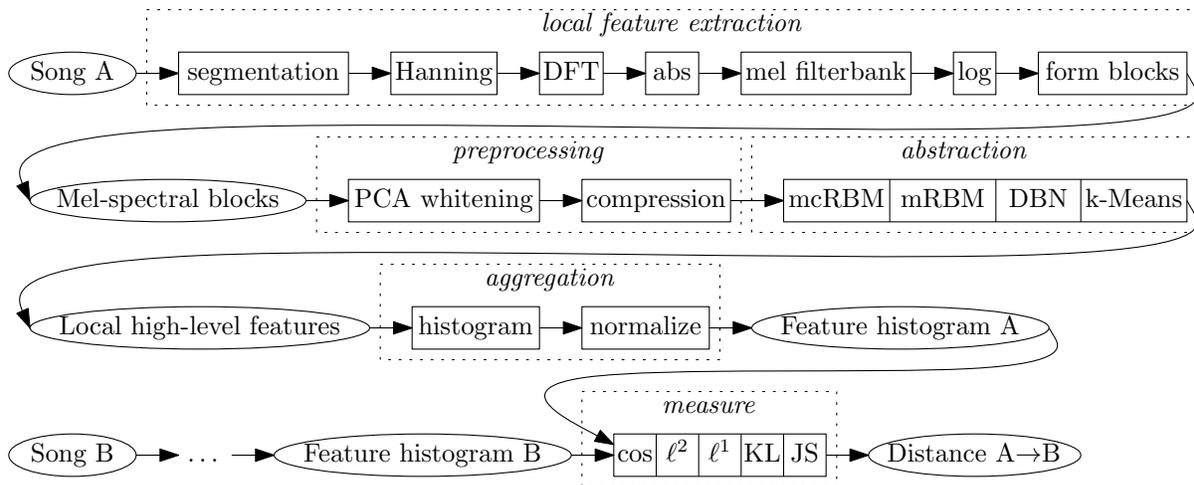


Fig. 1. Block diagram of our music similarity estimation system. Ellipses denote data, blocks denote transformations, tiled blocks represent alternatives.

III. OUR APPROACH

We designed our music similarity estimation system as depicted in Fig. 1. Its feature extraction/abstraction chain follows Dahl’s application of mcRBMs to speech recognition [13], but works with longer contexts as we assume musical features to last longer than phones. To build global song models, we calculate histograms of local features, which proved superior to Gaussian-based models in preliminary experiments.

In the following, we will discuss each stage in detail, including explanations of the methods for better understanding, choices of hyperparameters for easy reproduction and justifications of decisions wherever suitable.

A. Feature Extraction

Given the audio signal, our system extracts short mel-spectral frames using yaafe [18] (see first row of Fig. 1). We use a frame size of 64 ms with 32 ms overlap – this is in the typical range reported in literature, but has not been optimized – and a mel filterbank of either 40 bands from 130 to 6854 Hz (yaafe’s default) or 70 bands from 50 to 6854 Hz. Consecutive frames are then concatenated to form fixed-size blocks.¹ We tried block lengths of 9, 15, 39 and 75 frames with a hop size of 1 frame. For shorter blocks, subsequently learned features looked less interesting, and for longer blocks, our computational resources were insufficient. We did not try larger hop sizes as we do not expect a positive effect on the similarity estimation quality.

B. Feature Abstraction

The core of the system is a data-driven feature abstraction stage obtaining useful high-level descriptions of local features. In particular, we train a mean-covariance Restricted Boltzmann Machine (mcRBM) [1] on mel-spectral blocks sampled across

a collection of songs, then treat its *latent* representations conditioned on blocks as local high-level features.

To assess the benefit of an mcRBM over more basic feature abstractors, we compare it to standard RBMs and a variant of k-Means clustering suitable for large data sets. In addition, we experiment with Deep Belief Nets (DBNs) formed by training an RBM on another RBM’s feature descriptions. For all methods, data preprocessing turned out to be crucial. We will thus briefly describe how we preprocess data, then introduce RBMs and mcRBMs and finish with a short description of mini-batch k-Means++.

1) *Preprocessing*: The previously extracted mel-spectral blocks are both high-dimensional (up to 3000 dimensions) and highly correlated (in addition to correlations inherent in music, the components overlap both in time and frequency). We apply PCA whitening (decorrelation and normalization of each component to unit variance) and discard the least significant components to retain 99% of the original variance.

2) *The mean-covariance Restricted Boltzmann Machine*: A Restricted Boltzmann Machine (RBM) [19] is an undirected bipartite graphical model consisting of visible units v and latent variables h . It associates each configuration of v and h with an energy $E(v, h, \theta)$ inducing a probability density $p(v, h | \theta) = \frac{1}{Z(\theta)} e^{-E(v, h, \theta)}$, where $Z(\theta) = \sum_{u, g} e^{-E(u, g, \theta)}$ is the normalizing *partition function*. Training an RBM means finding θ such that $p(v | \theta)$ approximates the observed distribution of training data², after which the hidden unit probabilities $p(h | v, \theta)$ are interpretable as features extracted from v . A technique called Contrastive Divergence [19] allows fast training of RBMs without computing the partition function.

Different types of RBMs can be defined by choosing the energy function appropriately. Setting $E_m(v, h_m, \theta) = -v^T W h_m - \frac{1}{2}(v - a)^T (v - a) - b^T h_m$, where $\theta = (W, a, b)$ are the connection weights, visible and hidden bias terms, results in (conditionally) binary hidden units and Gaussian visible units $p(v | h_m, \theta) = \mathcal{N}(W h_m, I)$, such that columns

¹Effectively, these blocks are excerpts of a log-frequency log-magnitude spectrogram. This type of spectrogram is used for MFCCs, but also forms the basis for higher-level features engineered over the course of several years [6], [7], [8] – these are what we position our approach against, reusing their common low-level basis rather than trying to learn it as well.

²This equals maximizing the likelihood of the model under the data.

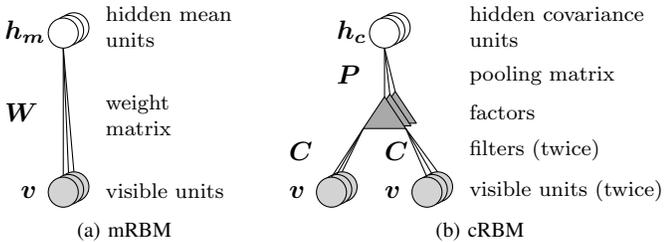


Fig. 2. Diagrams of the two parts of an mcRBM.

of \mathbf{W} can be interpreted as templates for representing input vectors. As only the means depend on \mathbf{h}_m , this model is called *mRBM* (Fig. 2a). It is the standard type of RBM to model real-valued inputs. However, independent Gaussian noise does not yield a good generative model for most real-world data.

To model pairwise dependencies of visibles gated by hidden, a third-order RBM can be defined, with a weight w_{ijk} for each triplet v_i, v_j, h_k . By factorizing and tying these weights [20], parameters can be reduced to a filter matrix \mathbf{C} connecting the input twice to a set of *factors* and a pooling matrix \mathbf{P} mapping factors to hidden variables (Fig. 2b). The energy function is $E_c(\mathbf{v}, \mathbf{h}_c, \boldsymbol{\theta}) = -(\mathbf{v}^T \mathbf{C})^2 \mathbf{P} \mathbf{h}_c - \mathbf{c}^T \mathbf{h}_c$, yielding $p(\mathbf{v} | \mathbf{h}_c, \boldsymbol{\theta}) = \mathcal{N}(0, (\mathbf{C} \text{diag}(\mathbf{P} \mathbf{h}_c) \mathbf{C}^T)^{-1})$. This model, termed *cRBM*, uses the states of its hidden units to represent abnormalities in the local covariance structure of a data point,³ combined from filters in \mathbf{C} . However, it can only generate Gaussian inputs of zero mean.

For general Gaussian-distributed inputs, the mRBM and cRBM can be combined into an *mcRBM* by simply adding their respective energy functions, resulting in $p(\mathbf{v} | \mathbf{h}_m, \mathbf{h}_c, \boldsymbol{\theta})$ becoming the product of the two models' conditional Gaussians. For more details on this model and its training procedure, we refer the reader to [1] and [20].

Replacing the mRBM's energy function with $E(\mathbf{v}, \mathbf{h}_b, \boldsymbol{\theta}) = -\mathbf{v}^T \mathbf{W} \mathbf{h}_b - \mathbf{v}^T \mathbf{a} - \mathbf{h}_b^T \mathbf{b}$, we obtain an RBM with both (conditionally) binary visible and hidden units. This *bRBM* can be trained to model the binary latent representations of another RBM, further abstracting the data. The resulting stack of two or more RBMs is referred to as a Deep Belief Net (DBN) [21].

3) *mini-batch k-Means++*: In order to assess the importance of using an mcRBM in our architecture, we compare it to a simple drop-in replacement for unsupervised feature learning: k-Means clustering. Despite its simplicity, k-Means has lately been shown to outperform both sparse RBMs and mcRBMs on an image classification task [22], making it an interesting benchmark candidate. It has also been applied to music feature extraction with both positive [3] and negative results [2], [23], but only in a frame-wise manner.

Lloyd's k-means algorithm [24] refines an initial clustering until convergence to a local optimum. We choose the initial cluster centres using *k-means++*, which gives an initial clustering that is $\Theta(\log k)$ -competitive with the optimal clustering

[25]. For refining, Lloyd's classic batch learning is slow when handling millions of data points. Instead we refine the clustering based on *mini-batches* (small random subsets of samples), converging orders of magnitude faster to only slightly worse solutions [26]. Having created a global codebook, features extracted from songs are abstracted by replacement with their cluster labels.

C. Global Modeling

We aggregate local features into a global descriptor by histogramming, which discards the order of features similar to Bag-of-Words approaches successfully used in text processing [27]. This is straightforward for k-Means features: We just count the number of occurrences of each cluster label. For mcRBMs, mRBMs and DBNs, we interpret hidden units as independent soft feature detectors and separately add up each unit's activations over the whole song.

To account for songs of different lengths, all histograms are normalized to unit ℓ^1 norm before comparing them.

D. Distance Measure

The feature histograms can be compared using any vector distance measure. Alternatively, normalized histograms can be interpreted as discrete probability distributions, for which further comparison methods exist. We tried the cosine, euclidean (ℓ^2) and manhattan (ℓ^1) distance as well as the symmetrized Kullback-Leibler and Jensen-Shannon (JS) divergence. Across all datasets, the best working measure was ℓ^1 , sometimes slightly surpassed by JS.

For better comparability to the state-of-the-art methods of [7] and [8], we optionally perform what Seyerlehner et al. termed Distance Space Normalization (DSN): After computing the full distance matrix, each entry is normalized with respect to the mean and standard deviation of its row and column.

IV. EXPERIMENTAL EVALUATION

In this section, we introduce the datasets used in our experiments, show and interpret features learned by the system and report results on its similarity estimation performance.

A. Datasets

We performed experiments on three freely available datasets. *1517-Artists* [8] consists of 3180 full tracks by 1517 artists spanning 19 almost uniformly distributed genres of Western music. *Homburg* [28] is composed of 1886 10-second song excerpts by 1436 artists unequally distributed over 9 genres that are similar to the ones in *1517-Artists*. The *Ballroom* dataset [29] contains 698 30-second snippets of music for 8 different ballroom dances, which is useful for evaluating rhythm features.

B. Training Details

Both RBM training and k-Means clustering were performed on GPU using cudamat [30]. To cut down computation times, models for *1517-Artists* were trained on a random subset of 1 million blocks. For mcRBMs, we used architectures of 1296 or 2500 factors, 324 or 625 hidden covariance and 256 or 512

³For example, in natural images, such abnormalities could be edges.

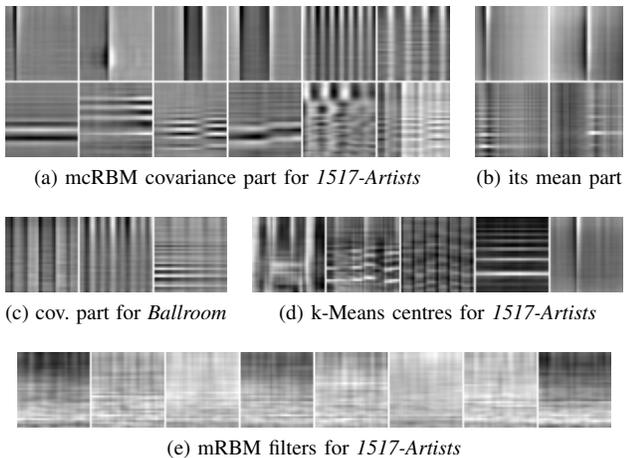


Fig. 3. Exemplary features learned by mcRBMs, mRBMs and k-Means on music excerpts of two datasets. Each block represents 1248 ms of a spectrogram: Time increases from left to right, mel-frequency from bottom to top, bright and dark indicate positive and negative values, respectively.

hidden mean units, referred to as *mcRBM 580* and *mcRBM 1137*, respectively. \mathbf{P} was initialized and constrained to a 2D topographic mapping, all other training parameters match the example configuration of the code accompanying [1]. mRBMs were trained for 100 epochs with momentum and weight decay, using 1024 hidden units to be of comparable size to the mcRBMs. k-Means++ was initialized on a subset of 350 000 blocks, then refined using a couple hundred mini-batches of 1024 blocks on the full datasets.

C. Learned Features

In Fig. 3, we visualize typical filters learned for the feature abstraction stage – for mcRBMs, we unwhiten the incoming weights of factors and mean units, for mRBMs, we unwhiten the incoming weights of hidden units, and for k-Means, we unwhiten the cluster centroids. Fig. 3a shows two examples each for recurring schemes in the covariance part (the factors) of an mcRBM trained on *1517-Artists* interpretable as (in reading order) note onsets, fixed-length notes, repetitive percussion, chords and note transitions, as well as two uninterpretable filters. The mean part of this model (Fig. 3b) only developed note onset and mixed tone filters. On *Ballroom*, results were similar, with a higher dominance of repetitive filters and fewer note onset or harmonic filters (Fig. 3c). k-Means learned less abstract features: When resynthesized, most cluster centres sound like noisy music excerpts including vocals; notable exceptions are strongly harmonic or look like note onsets (Fig. 3d). mRBMs fail to learn clearly-structured features (Fig. 3e). This corroborates Krizhevsky’s results [31], who only succeeded in training this model on image patches when reducing their size to 8×8 pixels.

D. k-NN Genre Classification and Retrieval

Under a good music similarity measure, pairs of songs assigned a low distance should be perceptually similar. Such pairs can be found by considering the k nearest neighbors to a

TABLE I
PRECISION AT 10 ON *1517-Artists* FOR SHALLOW MODELS

model	mel bands	block length				
		1	9	15	39	75
k-Means 1024	40	20.0	22.4	21.6	16.7	
mRBM 1024	40	12.7	12.8	12.5	12.9	
mcRBM 580	40	16.5	21.9	23.0	24.0	23.6
mcRBM 1137	40		20.1	22.4	24.2	23.4
mcRBM 580	70		22.9	23.6	23.6	

TABLE II
PRECISION AT 10 ON *1517-Artists* FOR STACKED RBMS

model	mel bands	block length	hidden layer			
			1.	2.	3.	4.
mRBM 1024	40	9	12.8	13.0	12.2	12.6
mcRBM 580	40	9	21.9	22.4	22.3	21.4
mcRBM 580	40	39	24.0	23.7	23.7	23.9

query song. Ideally, we would ask human test subjects to judge their perceptual similarity, but this is impractical. Instead, we assume songs of a same genre to be similar, and report the fraction of songs among the k nearest neighbors to a query that have the same genre as the query, averaged over all possible query songs (precision at k). We exclude same-artist tracks when determining the neighbors, as otherwise results would be over-optimistic [32].

While this evaluation method seems simplistic, it has been shown to highly correlate with human judgements [33, p.28 and 26], [34, p.52 and 57], and both precision at k and the related k -NN genre classification accuracy are established methods for evaluating music similarity measures [2], [3], [7], [8], [17], [23]. We focus on precisions because they are more consistent over different choices of k (Fig. 4).

Note that we are not actually interested in achieving a high precision or accuracy on a particular dataset. Unlike in classification, we use genre labels merely as binary similarity ground truth and do not train on them, as this would tie the measure to a specific genre taxonomy and dataset.

1) *Model and Input Sizes*: In Table I, we compare k-Means with 1024 centres to mRBMs with 1024 hidden units and mcRBMs with 580 or 1137 hidden units. Interestingly, k-Means performs best for relatively short blocks of 9 frames and gets notably worse for larger ones, while mcRBMs profit from longer blocks and are worst for single frames. Across all block lengths, mRBMs perform above the random baseline of 5.3%, but well below the other models. The small mcRBM beats k-Means by 1.8 percent points, even using only half as many features, and the larger mcRBM at blocks of 39 frames is still marginally better. Increasing the number of mel bands only helps on smaller blocks, and further increasing the block length to 75 frames hurts performance – possibly the dimensionality of the data gets too high in these cases.

2) *Deep Architectures*: In the next set of experiments, we assess the use of stacking bRBMs on top of a trained mRBM

TABLE III
PRECISION AT 10 ON *1517-Artists*, *Homburg* AND *Ballroom*
COMPARED TO STATE-OF-THE-ART APPROACHES

model	dim.	1517-A.	Homb.	Ballr.
Random baseline	0	5.3	15.6	12.8
G1 of MFCCs [35]	230	16.1	40.3	37.4
GMM-20 of MFCCs [2]	800	15.6		
RTBOF [7]	1331	25.5	46.2	77.5
BLS [8]	9448	26.5	45.3	67.7
k-Means 1024, 40×9*	1024	22.4	43.5	42.1
k-Means 1024, 40×9, DSN	1024	23.5	42.8	45.3
mcRBM 580, 70×15	580	23.6	44.0	50.0
mcRBM 580, 70×15, DSN	580	25.0	45.5	53.1
mcRBM 580, 40×39	580	24.0	44.0	62.8
mcRBM 580, 40×39, DSN	580	24.8	44.7	65.1
mcRBM 1137, 40×39	1137	24.2	44.2	61.0
mcRBM 1137, 40×39, DSN	1137	25.1	45.5	63.4

*size of input blocks in mel bands × block length

TABLE IV
CROSS-DATASET GENERALIZATION OF THE MCRBM MODELS,
PRECISION AT 10

mcRBM 580, 40×39		evaluated on		
		1517-A.	Homb.	Ballr.
trained on	1517-Artists	24.0	43.5	51.0
	Homburg	22.0	44.0	47.3
	Ballroom	20.4	39.9	62.8

or mcRBM to form a DBN extracting more abstract features. We use 2048, 1024 and 580 hidden units for the second, third and forth hidden layer, respectively. Table II does not show any consistent positive or negative effect. Possibly, the additional layers are only useful when subsequently fine-tuned to a task.

3) *Comparison to Existing Approaches:* As shown in Table III and Fig. 4, our approach compares favorably to the Single Gaussian and the Gaussian Mixture Model of MFCCs [35], [2]. Being based on a simple feature, surpassing these popular baseline approaches serves as a sanity check. More importantly, our results are in the vicinity of the state-of-the-art results⁴ of Pohle et al. [7] (RTBOF) and Seyerlehner et al. [8] (BLS), when using Distance Space Normalization (DSN) as employed by their methods. This is an encouraging result for unsupervised feature extraction, considering that both are based on several carefully tuned hand-crafted features. Note that we achieve these results with 580-dimensional song models, while RTBOF and BLS use 1331 and 9448 dimensions, respectively.

As the only exception, there is a large margin to RTBOF on *Ballroom* (which RTBOF has been optimized on). This suggests that our system is not yet optimally suited for modeling rhythms, although it does learn some useful long-term features if given a chance to: On *Ballroom*, short blocks perform considerably worse than long blocks.

⁴reproduced using the distance matrices available at www.seyerlehner.info

4) *Generalization:* Up to now, we have only reported results for models trained unsupervisedly on the very same dataset they are evaluated on. For most practical applications, it is infeasible to always train a feature extractor on the collection we want to compute similarities for.

To investigate how well a learned feature set generalizes, we evaluate it on the datasets it has not been trained on.⁵ Table IV shows that the feature extractors of *1517-Artists* apply well to *Homburg*, but not vice versa, possibly because *Homburg* contains less training data. Extractors of *Ballroom* do not perform well on the other datasets and vice versa. This indicates that feature extractors trained on a particular collection are only applicable to collections of music similar to it. The *Ballroom* dataset contains a fairly different set of musical styles and possibly requires features the model could not learn from *1517-Artists* or *Homburg*. Indeed, in Section IV-C we noted that models trained on *Ballroom* seemed to develop more rhythmic feature detectors.

This can be seen as a disadvantage against the state-of-the-art, which performs well on all three datasets with a single set of features. However, future experiments will show if training on a more diverse collection yields more general, competitive features and whether our method can benefit from its adaptability when applied to non-Western music.

V. CONCLUSION

We have presented a system for content-based music similarity estimation based on a feature set learned unsupervisedly by an mcRBM. It does not surpass the carefully engineered state-of-the-art in similarity estimation, but performs remarkably close to it without requiring any time-intensive hand-crafting of features. It also turns out that simple k-means clustering for feature extraction reaches good performance, if data is appropriately preprocessed.

To improve on these initial results, we will explore several extensions to our method: By training deep architectures to hierarchically integrate information over longer contexts, we hope to improve modeling of temporal structure such as rhythms. To yield pitch-independent features, we will limit spectrogram excerpts not only in time, but also in frequency.

In addition to employing unsupervised learning for local feature extraction, we will leverage larger datasets to also unsupervisedly learn the global song models. More generally, we think that separating learning of feature sets from using them for encoding of music data, similar to [36], is an exciting approach for future research.

ACKNOWLEDGMENTS

The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry for Transport, Innovation, and Technology. The research is partly funded by the Austrian Science Fund (FWF): Z159.

⁵Ideally, we would also evaluate generalization on a train/test split of each dataset, but the datasets are too small, and n-fold cross-validation is unfeasible for the number of mcRBM models we are training.

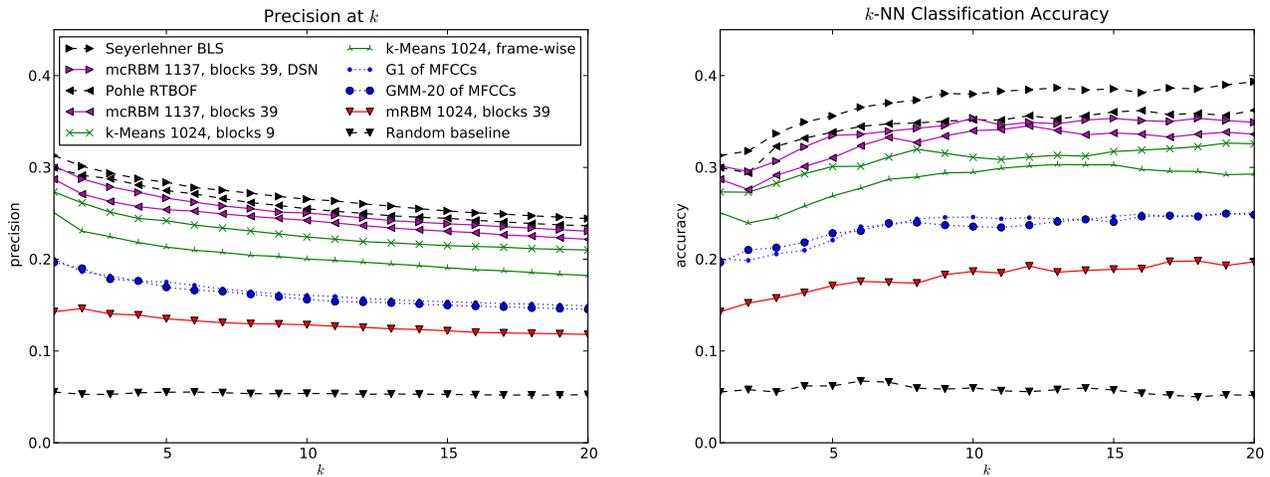


Fig. 4. Precision at k and k -NN classification accuracy for our best models, two baseline approaches and the state-of-the-art on *1517-Artists*

REFERENCES

- [1] M. Ranzato and G. Hinton, "Modeling Pixel Means and Covariances Using Factorized Third-Order Boltzmann Machines," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'10)*, 2010, pp. 2551–2558.
- [2] J.-J. Aucouturier, "Ten experiments on the modelling of polyphonic timbre," Ph.D. dissertation, University of Paris 6, France, May 2006.
- [3] M. Hoffman, D. Blei, and P. Cook, "Content-Based Musical Similarity Computation using the Hierarchical Dirichlet Process," in *Proc. of the 9th Int. Conf. on Music Information Retrieval (ISMIR 2008)*, 2008.
- [4] M. Slaney, K. Weinberger, and W. White, "Learning a metric for music similarity," in *Proc. of the 9th Int. Conf. on Music Information Retrieval (ISMIR 2008)*, 2008, pp. 313–318.
- [5] B. Logan, "Mel Frequency Cepstral Coefficients for Music Modeling," in *Proc. of 1st Int. Conf. on Music Information Retrieval*, 2000.
- [6] 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, pp. 570–579.
- [7] T. Pohle, D. Schnitzer, M. Schedl, P. Knees, and G. Widmer, "On rhythm and general music similarity," in *Proc. of the 10th Int. Conf. on Music Information Retrieval (ISMIR 2009)*, 2009, pp. 525–530.
- [8] K. Seyerlehner, G. Widmer, and T. Pohle, "Fusing Block-Level Features for Music Similarity Estimation," in *Proc. of the 13th Int. Conf. on Digital Audio Effects (DAFx-10)*, Graz, Austria, 2010.
- [9] E. Camboropoulos, "Towards a General Computational Theory of Musical Structure," Ph.D. dissertation, University of Edinburgh, UK, 1998.
- [10] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.
- [11] M. Lewicki, "Learning efficient codes of natural sounds yields cochlear filter properties," in *Int. Conf. on Neural Information Processing*, 2000.
- [12] S. A. Abdallah, "Towards Music Perception by Redundancy Reduction and Unsupervised Learning in Probabilistic Models," Ph.D. dissertation, King's College London, London, UK, 2002.
- [13] G. Dahl, M. Ranzato, A. Mohamed, and G. Hinton, "Phone recognition with the mean-covariance restricted Boltzmann machine," in *Advances in Neural Information Processing Systems 23*, 2010, pp. 469–477.
- [14] M. Hoffman, D. Blei, and P. Cook, "Finding Latent Sources in Recorded Music With a Shift-Invariant HDP," in *Proc. of the 12th Int. Conf. on Digital Audio Effects (DAFx-09)*, Como, Italy, 2009.
- [15] H. Lee, Y. Largman, P. Pham, and A. Ng, "Unsupervised Feature Learning for Audio Classification using Convolutional Deep Belief Networks," in *Adv. in Neural Information Processing Systems 22*, 2009.
- [16] P. Hamel and D. Eck, "Learning Features from Music Audio with Deep Belief Networks," in *Proc. of the 11th Int. Conf. on Music Information Retrieval (ISMIR 2010)*, 2010, pp. 339–344.
- [17] T. Pohle, P. Knees, M. Schedl, and G. Widmer, "Independent Component Analysis for Music Similarity Computation," in *Proc. of the 7th Int. Conf. on Music Information Retrieval (ISMIR 2006)*, 2006.
- [18] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard, "Yaafe, an easy to use and efficient audio feature extraction software," in *Proc. of the 11th Int. Conf. on Music Information Retrieval (ISMIR 2010)*, 2010.
- [19] G. E. Hinton, "Training Products of Experts by Minimizing Contrastive Divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [20] M. Ranzato, A. Krizhevsky, and G. Hinton, "Factored 3-Way Restricted Boltzmann Machines For Modeling Natural Images," in *Proc. of the 13th Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2010.
- [21] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, 2006.
- [22] A. Coates, H. Lee, and A. Ng, "An Analysis of Single-Layer Networks in Unsupervised Feature Learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.
- [23] K. Seyerlehner, G. Widmer, and P. Knees, "Frame-level Audio Similarity – A Codebook Approach," in *Proc. of the 11th Int. Conf. on Digital Audio Effects (DAFx-08)*, Espoo, Finland, 2008.
- [24] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [25] D. Arthur and S. Vassilvitskii, "k-means++: The Advantages of Careful Seeding," in *Proc. of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'07)*, 2007, pp. 1027–1035.
- [26] D. Sculley, "Web-Scale K-Means Clustering," in *Proc. of the 19th Int. Conf. on World Wide Web (WWW'10)*, 2010, pp. 1177–1178.
- [27] D. Lewis, "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval," in *Machine Learning: ECML-98*, ser. Lecture Notes in Computer Science. Springer, 1998, vol. 1398, pp. 4–15.
- [28] H. Homburg, I. Mierswa, B. Möller, K. Morik, and M. Wurst, "A Benchmark Dataset for Audio Classification and Clustering," in *Proc. of the 6th Int. Conf. on Music Information Retrieval (ISMIR 2005)*, 2005.
- [29] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer, "Evaluating Rhythmic Descriptors for Musical Genre Classification," in *Proc. of the 25th Int. AES Conf.*, 2004.
- [30] V. Mnih, "CUDAMat: A CUDA-based matrix class for Python," Dep. of Comp. Sci., Univ. of Toronto, Tech. Rep. UTML TR 2009-04, 2009.
- [31] A. Krizhevsky, "Learning multiple layers of features from tiny images," Master's thesis, Dept. of Comp. Science, Univ. of Toronto, 2009.
- [32] A. Flexer and D. Schnitzer, "Effects of album and artist filters in audio similarity computed for very large music databases," *Computer Music Journal*, vol. 34, no. 3, pp. 20–28, 2010.
- [33] T. Pohle, "Automatic characterization of music for intuitive retrieval," Ph.D. dissertation, Johannes Kepler University, Linz, Austria, 2010.
- [34] K. Seyerlehner, "Content-based music recommender systems: Beyond simple frame-level audio similarity," Ph.D. dissertation, Johannes Kepler University, Linz, Austria, 2010.
- [35] M. Mandel and D. Ellis, "Song-level features and support vector machines for music classification," in *Proc. of the 6th Int. Conf. on Music Information Retrieval (ISMIR 2005)*, 2005, pp. 594–599.
- [36] A. Coates and A. Ng, "The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization," in *Proc. of the 28th Int. Conf. on Machine Learning*, 2011.