Feature Preprocessing with RBMs for Music Similarity Learning

Son N. Tran[1], Daniel Wolff[1], Tillman Weyde[1], Artur Garcez[1]

[1]*Department of Computer Science, City University London, Northampton Square, EC1V 0HB, UK*

Correspondence should be addressed to Son N. Tran (`son.tran.1@city.ac.uk`)

ABSTRACT

Computational modelling of music similarity constitutes a key element for music information retrieval and recommendation systems. Similarity models and analysis are also important for research in musicology and music perception. In this study, we test feature preprocessing with Restricted Boltzmann Machines in combination with established methods for learning distance measures. Our experiments show that this preprocessing improves the overall generalisation results of the trained models. We compare the effects of feature preprocessing on distance function learning using gradient ascent and support vector machines. The evaluation is preformed using similarity data from the MagnaTagATune dataset, which allows a comparison of our results to previous studies.

1. INTRODUCTION

Music similarity is a core concept used in many applications in Music Information Retrieval, such as music recommendation, exploration and classification. Moreover, similarity is important in music research into aspects such as provenance or originality as well as in music analysis, e.g. in paradigmatic analysis.

In this study we provide an analysis of the effect of preprocessing feature vectors with Restricted Boltzmann Machines (RBMs) on learning similarity measures on music audio. RBMs determine a non-linear transform the feature space in an unsupervised learning step. RBMs have been used successfully for learning and representing audio features in other applications (see [6, 15] and below), but to our knowledge, no research on audio similarity learning from user ground truth has been done with RMBs so far.

In this paper we model audio similarity using standard machine learning techniques (Support Vector Machines and Gradient Descent) for adapting a weighted distance measure to human similarity ratings. The weighted Euclidean distance measure is used for modelling the distance, or inverse similarity of two songs. The dataset used in these experiments is part of the MagnaTagATune dataset, consisting of relative similarity ratings between pairs of clips.

The unsupervised training of the Restricted Boltzmann Machine does not directly optimise the similarity measure, as it does not include the similarity ground truth. However, Restricted Boltzmann Machines have been shown to help in other tasks by transforming the feature space in a way that makes machine learning easier. The transformations change the space of functions that can be modelled by parameterising simple models, e.g. by including interactions between individual features. A transformation into a more suitable representation, determined by unsupervised training, can thus lead to better adaptation of the model to given similarity data.

The code used for the experiments in Section 5, containing an RBM toolbox for Matlab, can be retrieved online[1].

2. RELATED WORK

The weighted Euclidean distance used in this paper is a special case of the Mahalanobis metric [11], providing a standard model for a parametrized similarity measure. The weighted Euclidean distance weights the features but, in contrast to the full Mahalanobis matrix, not the interaction between features.

So far, different data sources have been used for learning distance measures to address specific scenarios and

[1] `http://mi.soi.city.ac.uk/blog/codeapps/camiraes2013`

availability of data sources: Working in a music recommendation scenario, McFee et al. [12] and Lim et al. [10] adapt a music similarity towards collaborative filtering data. They use Mahalanobis metrics to describe a parametrized linear combination of content-based features, using Metric Learning to Rank (MLR) for training. The similarity is calculated in kernel space. Ellis and Whitman [2] use relative similarity data from a comparative survey on artist similarity for comparison with similarity metrics learnt from similar artist lists from the All Music Guide[2].

This study is based on a subset of the MagnaTagATune dataset [7], which is based on music from the Magnatune label. We use the data from the bonus round where users where asked to identify an outlier within three audio clips. Stober and Nürnberger [22] used this dataset to compare algorithms for linear and quadratic optimisation of a similarity measure based on feature weighting. They applied early fusion of the feature data followed by adapting a linear model. The training approaches in [22] have been compared to MLR and SVM-Light by Wolff et al. [27]. By using the same features and similarity data, which are both available online, and the same SVM-Light implementation we aim to make our results comparable to these earlier findings.

The effect of the selection of feature information and their representation for similarity learning have been analysed in experiments by Wolff and Weyde [26]. These experiments show variable results, but combining features with complementary information leads to the best learning results. Feature dimension reduction with Principal Component Analysis (PCA) can have a positive or negative effect depending on the learning model used.

In this paper we use unsupervised training with Restricted Boltzmann Machines [20] to transform the feature space. Recently, several algorithms have been developed, which are able to learn features from datasets in different domains [4, 5, 9, 8, 16, 24]. In applications to computer vision, the state-of-the-art feature learning shows similar or better performance compared to non-learning algorithms [9, 8, 16].

Schlüter and Osendorfer [17] used RBMs to model similarity regarding musical genre. They applied a Mean-Covariance RBM on processed Mel Frequency Cepstral Coefficients (MFCC) to learn local high-level features and aggregated them to feature histograms for whole songs. The similarity of songs is then quantified as distance between the songs' feature histograms using five measurement methods: cosine distance, the Euclidean metric, Manhattan distance, and symmetrized Kullback-Leibler and Jenson-Shannon divergence. Hamel and Eck [3] also used Deep Belief Networks (DBN) for genre classification with a Gaussian kernel Support Vector Machine (SVM) and showed improvements on their baseline approach.

Nam et al. [13] used DBNs for automatic transcription of piano music using a similar SVM classifier. Their transformations of spectrogram features showed improved performance both when using the first hidden layer of an RBM and when fine tuning a DBN via backpropagation. A methodical overview for using learnt features for MIR tasks has been presented by Nam et al. [14]. They further show the effectiveness of their approach in tag classification with linear kernel SVM on the CAL500 dataset.

Schmidt et al. [18] applied DBNs to learn three types of emotion-based acoustic features. Their experimental results showed that the sort-time features learnt by DBN outperform MFCC, Chroma, Spectral Shape, ENT, and Spectral Contrast. The performance is further improved by the outputs from hidden layers of DBN trained on multi-frame and universal background model features.

Dieleman et al. [1] applied Convolutional Deep Belief Networks to learn from audio features and metadata in the 'Million Song Dataset for artist recognition, genre recognition, and key detection. In all three tasks, they first train the DBN and subsequently use it as initialization of a multilayer perceptron. As reported, their approach achieved better performance than naive Bayesian and windowed logistic regression models.

Our approach differs from previous work in that it transforms mid-level feature representations such as chroma, timbre and genre tag information instead of more low-level audio descriptors such as MFCCs or spectrogram frames. The transformations are achieved using RBMs with unsupervised learning, which are summarised in the following section.

3. FEATURE SPACE TRANSFORMATION WITH RESTRICTED BOLTZMANN MACHINES

A Restricted Boltzmann Machine (RBM) [20] is a two-layer connectionist system characterized by an energy function E

[2] http://www.allmusic.com/

$$E(\mathsf{v},\mathsf{h}) = -\sum_{ij} \mathsf{v}_i w_{ij} \mathsf{h}_j - \sum_i a_i \mathsf{v}_i - \sum_j b_j \mathsf{h}_j \qquad (1)$$

to represent the joint distribution:

$$P(\mathsf{v},\mathsf{h}) = \frac{1}{Z} \exp\left(-E(\mathsf{v},\mathsf{h})\right) \qquad (2)$$

with the partition function $Z = \sum_{\mathsf{v},\mathsf{h}} \exp\left(-E(\mathsf{v},\mathsf{h})\right)$. Here, v,h describe the states of the visible and hidden layers, W represents connection weights, and a,b hold biases for visible and hidden layers respectively. Since all units in one layer are independent of each other, the state of a hidden unit depends only on the states of visible units and vice versa. The probability of a unit being activated is:

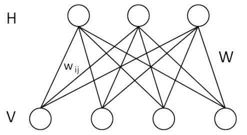

Fig. 1: Restricted Boltzmann Machine

$$P(\mathsf{h}_j = 1|\mathsf{v}) = \sigma(\sum_i \mathsf{v}_i w_{ij} + b_j) \qquad (3)$$

$$P(\mathsf{v}_i = 1|\mathsf{h}) = \sigma(\sum_j \mathsf{h}_j w_{ij} + a_i) \qquad (4)$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the logistic sigmoid function.

Unsupervised training of an RBM means to maximize its average log-likelihood $\hat{\ell}$ (or equivalently the product of probabilities) given a set of independent and identically distributed samples $\mathscr{V} = \{\mathsf{v}^{(1)}, \mathsf{v}^{(2)}, ..., \mathsf{v}^{(n)}\}$

$$\hat{\ell} = \frac{1}{N} \ln(\mathscr{L}(\theta|\mathscr{V})) = \frac{1}{N} \sum_k \ln P(\mathsf{v}^{(k)}|\theta) \qquad (5)$$

with $\theta = \{W, a, b\}$. This can be achieved using gradient ascent. However, to compute the exact gradient from the log-likelihood it is needed to compute the partition function Z. For this, the gradient ascent requires an expectation of data sampled from the model as

$$w_{ij} = w_{ij} + \eta(\langle \mathsf{v}_i p(\mathsf{h}_j|\mathsf{v})\rangle_0 - \langle \mathsf{v}_i \mathsf{h}_j \rangle_\infty). \qquad (6)$$

Here, $\langle . \rangle_0$ is the average with regards to the data distribution, $\langle . \rangle_\infty$ is the average with respect to distribution from the model, and η is the learning rate.

An approximate approach to this problem is to sample the states from the model using Markov Chain Monte Carlo (MCMC). This method, however, is very slow and unstable since the model needs to perform a long and unspecified pre-sampling process before reaching an equilibrium state and generating valid samples. Hinton [4] proposed an algorithm named *Contrastive Divergence* (CD) showing how even with small number of running steps of pre-sampling (even only 1 step) the learning can approximately minimize the divergence between data distribution and the distribution of model:

$$w_{ij} = w_{ij} + \eta(\langle \mathsf{v}_i \mathsf{h}_j \rangle_0 - \langle \mathsf{v}_i \mathsf{h}_j \rangle_n). \qquad (7)$$

When setting the visible layer of an RBM to the original feature values of a song, the hidden layer represents a non-linearly transformed feature space, which we evaluate for similarity learning in the experiments below. As an important hyperparameter, the number of units in the hidden layer (*hidNum*) determines the dimensionality of the new feature space, and is particularly examined in figs. 3 and 4. The Matlab source used for training Restricted Boltzmann Machines is available for download[3].

4. SIMILARITY MODELS

The similarity models we compare in this paper are based on a weighted Euclidean metric. The weighted distance of two song's feature vectors is used as the inverse indicator for those songs similarity. The weighted Euclidean distance of two feature vectors $x, y \in \mathbf{R}^N$ is defined as

$$dist(x,y) = \sqrt{\sum a_i d_i}, \qquad (8)$$

$$\text{where} \quad d_i(x,y) = (x_i - y_i)^2 \qquad (9)$$

Here, the *facet distances* d_i measure the distance of the single features. Each facet distance $d_i(x,z)$ is assigned a weight a_i. Note that $dist(x,y)$ only qualifies as a metric iff $a_i > 0 \forall i$. In other cases, the measure might still be suitable as a distance measure, but lacks properties such as non-negativity and identity of points with distance 0. Stober and Nürnberger [23], Wolff et al. [27] showed that using feature specific functions to calculate facet distances d_i for different types of data in the feature vectors can improve the results and the stability of model training. This is only possible if the interpretation of a

[3] `urlhiddenforblindreview`

facet x_i in the feature vector is clearly defined, and therefore does not straightforwardly apply when using RBM preprocessing of features. However, RBMs already supply a method of specialised treatment of different facets by the nature of the transformation they define. Early experiments have also shown possible further improvements through convolution of feature information which will be explored in future work.

4.1. Model Training

When modelling similarity based on user data, the contribution of each facet distance to the comparison may be different depending on the user's input. In order to satisfy the constraints given by users' similarity judgements, the weights a_i in Equation (8) are learnt from the observed data. A constraint $h(x,y,z)$ determines whether song x should be more similar to song y than to song z.

$$h(x,y,z) = \begin{cases} true & \text{iff } x \text{ is more similar to } y \\ false & \text{iff } x \text{ is more similar to } z \end{cases} \tag{10}$$

Given our distance measure in Equation (8), we can infer constraints on our similarity model. We indirectly optimizing the boolean functions by optimizing the weighted sum of differences between facet distances over all training triplets.

$$h(x,y,z) = \sum_i a_i(d_i(x,z) - d_i(x,y)) > 0 \tag{11}$$

Let our training data be in the form of

$$\mathscr{D} = \{d^{(k)} \mid k = 1,...,M\} \text{ where}$$
$$d^{(k)} = \{x^{(k)}, y^{(k)}, z^{(k)} \mid h(x^{(k)}, y^{(k)}, z^{(k)}) = true\}$$

with feature vectors $x^{(k)}, y^{(k)}, z^{(k)} \in \mathbf{R}^N$. Given a training set $\mathscr{D}$ we want to find a weight vector $a \in \mathbf{R}^N$ which maximizes

$$f(a) = \frac{1}{M}\sum_k^M \sum_i a_i(d_i(x,z) - d_i(x,y)). \tag{12}$$

4.2. Gradient Ascent

This standard optimisation method has been used by Stober and Nürnberger with the MagnaTagATune dataset, and served as a good measure for baseline performance. The function $f(w)$ in (12) is linear and its optima may be found at very large values of w if there is no constraint applied. In our experiments we use gradient ascent with regularization and early stopping to iteratively search for the optimal weights a_i. The iterative process uses the following update rule:

$$a = a + \eta(\Delta a - \gamma a), \quad \text{with}$$
$$\Delta a_i = \frac{\partial f(a)}{\partial a_i} = \frac{1}{M}\sum_k^M (d_i(x^{(k)}, z^{(k)}) - d_i(x^{(k)}, y^{(k)}))$$

4.3. Support Vector Machine

The results of the gradient descent method are furthermore compared to the method of using Support Vector Machines for distance metric learning as introduced by Schultz and Joachims [19]. We apply this method as it has been used on the MagnaTagATune dataset for learning distance metrics in [27].

For learning a weighted distance measure with SVMs, the classifier is optimized to produce a vector of weights a that fulfils the distance constraints. Here, for each constraint $h(x,y,z)$, we construct a feature distance difference vector $\delta^{(x,y,z)} \in \mathbf{R}^N$ with

$$\delta_i^{(x,y,z)} = d_i(x,z) - d_i(x,y), \tag{13}$$
$$= (x_i - z_i)^2 - (x_i - y_i)^2 \tag{14}$$

Optimization is performed as follows:

$$\begin{aligned} \min_{a,\xi} \quad G(a) = \quad & \frac{1}{2}a^T a + c\sum_{(x,y,z)} \xi_{(x,y,z)} \\ \text{s.t.} \forall (x,y,z) \quad & a^T \delta^{(x,y,z)} \geq 1 - \xi_{(x,y,z)} \\ & \xi_{(x,y,z)} \geq 0 \\ & a_i \geq 0 \end{aligned} \tag{15}$$

Here, c determines a trade-off between regularisation and the enforcement of constraints. The slack variables $\xi_{(x,y,z)}$ allow for some constraints to be violated whilst adding a penalty value to the optimisation result.

5. EXPERIMENTS

The MagnaTagATune dataset used for our experiments contains audio features for over 25863 clips extracted by The Echo Nest API as well as relative similarity data for 1019 of the clips. In order to allow for comparison with previous results, our experiments use the similarity data and cross-validation segmentation from [27]. The corresponding similarity data and audio features are available online[4]. For details on the extraction of features and similarity data we refer to [27]. All experiments with SVM-Light were performed with the regularisation parameter $c = 1$, which was found to give good results in previous experiments.

5.1. RBM feature extraction

In this experiment we evaluate how the performance of similarity learning models is affected by using the RBM feature transformation as a preprocessing step. First, the parameters of the RBM and its unsupervised training are explored and then fixed to compare results with the data published in [27] using the same 10-fold cross-validation.

Possible parameter combinations are tested using a grid search over a predefined range of values as displayed in Table 1 with following similarity modelling using the similarity training sets. We then use the mean training set accuracy of each tested model to fix a model and its parameters used for the final evaluation. Optimal parameters were selected for each of the similarity learning algorithms, and the configurations selected for testing are depicted in Table 2. Since using training accuracy for model selection is risky to overfitting, we carefully apply strict regularlisation for training a model.

Param.	Values Tested
$hidNum$	$30, 50, 100, 500, 1000$
$lrate1$	$0.02, 0.05, 0.1, 0.5, 0.7$
$lrate2$	$0.1, 0.5, 0.7$
$momentum$	$0.05, 0.1$
$cost$	$0.00002, 0.01$

Table 1: Values used for the RBM grid search

The "Original Features" used in our experiments are the same as in [27]. They contain audio features from The Echo Nest API: chroma and timbre information, as well as features derived through classification (e.g. tempo and meter). Furthermore, genre and other tag information is included in binary features. Figure 2 shows an example of feature transformation from original features to outputs from the hidden layer of an RBM trained on the dataset.

	Approach	
Param.	GRAD	SVM-Light
$hidNum$	500	1000
$lrate1$	0.70	0.05
$lrate2$	0.70	0.10
$momentum$	0.05	0.10
$cost$	$2.0e-5$	$2.0e-5$

Table 2: Parameters chosen for Gradient Descent (GRAD) and SVM-Light in the final experiments.

	Features		
Appr.	Original	PCA	RBM
GRAD	70.47 / 71.68	70.54 / 70.52	73.14 / 73.28
SVM	71.20 / 83.54	70.17 / 75.29	72.18 / 80.17

Table 3: Comparison of original features and those with PCA and RBM preprocessing. Displayed are test/training set results for configurations with best training success. The SVM Original values are taken from [27].

Table 3 shows the performance of different feature preprocessing strategies. For Gradient Ascent, the average result over several runs with the same RBM parametrisation is taken into the comparison to allow for a more general evaluation of the probabilistic model. Unfortunately, this was not possible with SVM-Light because of time constraints, and the results of single runs are displayed for this approach.
For the original features, the results for gradient ascent (GRAD) are comparable to those published in [27]. Note that this gradient ascent approach differs slightly from that in [22, 27], where the weights a_i are constrained to $\sum w_i = 1$. The SVM-Light results for original features are reproduced from that publication. When using PCA, the results for SVM-Light are slightly worse than in the original features, while the gradient approach does improve very little.

The RBM features significantly improve the results for all approaches, with gradient ascent the best test results, improving by 2.67% over the original features, while SVM-Light gains 0.92%.

Figures 3 and 4 show the train and test set performances of all learning algorithms with respect to the number of hidden units in the RBM preprocessing. For these experiments configurations of the RBM have been fixed to those reported in Table 2, except for the $hidNum$ pa-

[4] http://mi.soi.city.ac.uk/datasets/ismir2012/

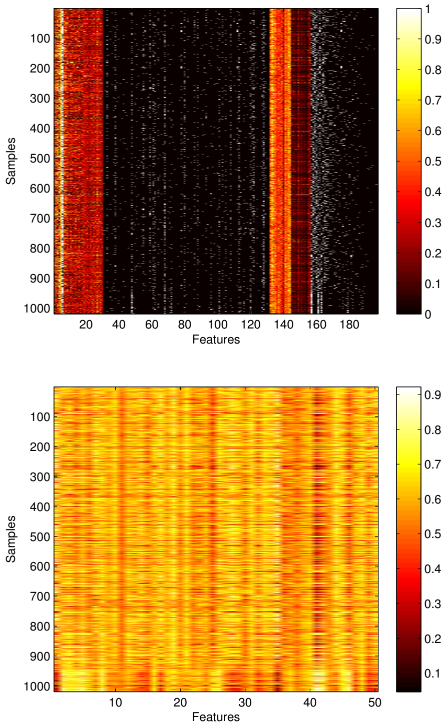

Fig. 2: Example feature transformation: (a) Original features(audio and genre tags), (b) Transformed features from *hidNum* = 50 hidden layer outputs of an RBM

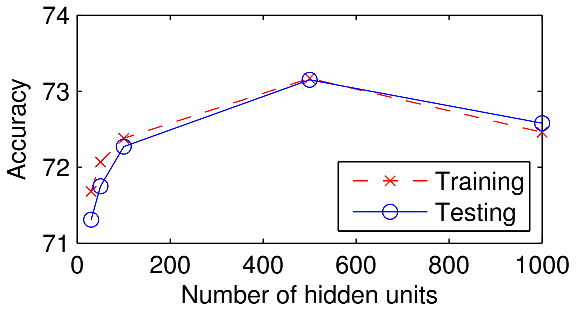

Fig. 3: Test set performances of Gradient Ascent with different dimensionality of RBM features.

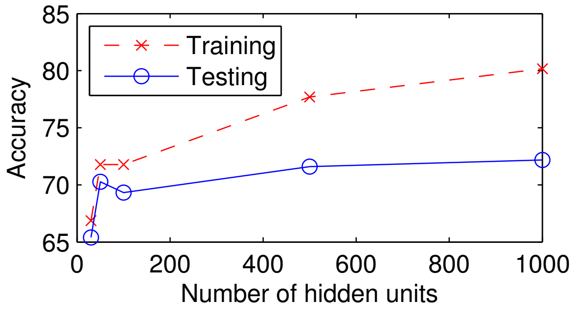

Fig. 4: Test set performances of SVM-Light with different dimensionality of RBM features.

rameter which is varied according to the values in Table 1. SVM-Light reaches its highest performance with the maximal number of 1000 hidden units. The performance at 30 units very low at 65.40%. Using an output feature dimension of 1000 gains 6.78% to a maximal performance of 73.30%. This trend cannot be found for Gradient Descent, which reaches its maximum performance, also the best performance reported in this paper, at 500 hidden units.

6. DISCUSSION

The results in Table 3 show a general performance gain when using Restricted Boltzmann Machines for feature preprocessing in similarity learning. Apart from the gains through transformation of the feature space, the large dimensionality of the resulting features may improve results as well: Different feature dimensions lead to a different number of parameters a_i. For SVM-Light, the presented results were achieved with $hidNum = 1000$, allowing the model over 5 times more parameters than for the original features, which have a dimensionality of 197. On the other hand, learning models with many parameters is also complex and requires a larger number of training examples. Across algorithms, we found no clear trend for the number of RBM features, i.e. RBM hidden layer units. [25] compared the impact of parameter reduction using PCA, showing slightly higher performance of SVM-Light for the models with reduced feature dimensionality. But in [26], the authors found that performance does not change significantly when reducing feature dimensionality.

In this study, the method of RBM preprocessing together with selection of the best RBM features the on grounds of training performance provides an effective way boost classification performance. When available, an additional validation set might allow for an even better selection of promising features. Our experiments show that in this way basic learning algorithms such as gradient ascent can adapt better to complex data such as the presented similarity ratings.

7. CONCLUSION AND FUTURE WORK

Our experiments show that transforming features using RBMs can improve both results of similarity learning with Gradient Ascent and Support Vector Machines on music audio.

For Gradient Ascent, the model achieved competitive performance to the other approaches, increasing the testing accuracy from 70.47% to 73.14% For Support Vector Machines, the RBM features allowed for a smaller but still significant rise from 71.20% to 72.18%. Comparing to the features extracted using PCA, the features processed using RBM show better performance and more consistent improvement.

For future work we are interested in discovering the similarity relations by comparing subspace distances built by combining different feature dimensions. By using validation sets, we expect to select RBMs with even better generalisation. Furthermore, we would like to apply a knowledge extraction method [21] to understand why the similarity relation can be captured by only a single layers of hidden units in the RBM.

References

[1] Sander Dieleman, Philémon Brakel, and Benjamin Schrauwen. Audio-based music classification with a pretrained convolutional network. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 669–674, Miami (Florida), USA, October 24-28 2011.

[2] Daniel P. W. Ellis and Brian Whitman. The quest for ground truth in musical artist similarity. In *Proceedings of ISMIR 2002*, pages 170–177, 2002.

[3] Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 339–344, Utrecht, The Netherlands, August 9-13 2010.

[4] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, August 2002.

[5] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.

[6] Navdeep Jaitly and Geoffrey E. Hinton. Learning a better representation of speech soundwaves using restricted boltzmann machines. In *ICASSP*, pages 5884–5887, 2011.

[7] Edith Law and Luis Von Ahn. Input-agreement: A new mechanism for collecting data using human computation games. In *Proc. of CHI*. ACM Press, 2009.

[8] Quoc Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg Corrado, Jeff Dean, and Andrew Ng. Building high-level features using large scale unsupervised learning. In *Proceedings of ICML 2012*, 2012.

[9] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of ICML 2009*, ICML '09, page 609–616, New York, NY, USA, 2009. ACM.

[10] Daryl Lim, Brian Mcfee, and Gert R. Lanckriet. Robust structural metric learning. In Sanjoy Dasgupta and David Mcallester, editors, *International Conference on Machine Learning ICML WS-13*, volume 28, pages 615–623. JMLR Workshop and Conference Proceedings, 2013.

[11] P C Mahalanobis. On the generalised distance in statistics. In *Proceedings of the National Institute of Sciences of India 2*, page 49–55. MIT Press, 1936.

[12] B. McFee, L. Barrington, and G. Lanckriet. Learning similarity from collaborative filters. In *Proceedings of ISMIR 2010*, pages 345–350, 2010.

[13] Juhan Nam, Jiquan Ngiam, Honglak Lee, and Malcolm Slaney. A classification-based polyphonic piano transcription approach using learned feature representations. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 175–180, Miami (Florida), USA, October 24-28 2011.

[14] Juhan Nam, Jorge Herrera, Malcolm Slaney, and Julius Smith. Learning sparse feature representations for music annotation and retrieval. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, October 8-12 2012.

[15] Abdel rahman Mohamed and Geoffrey E. Hinton. Phone recognition using restricted boltzmann machines. In *ICASSP*, pages 4354–4357, 2010.

[16] Ruslan Salakhutdinov and Geoffrey E. Hinton. Deep boltzmann machines. *Journal of Machine Learning Research - Proceedings Track*, 5:448–455, 2009.

[17] Jan Schlüter and Christian Osendorfer. Music Similarity Estimation with the Mean-Covariance Restricted Boltzmann Machine. In *Proceedings of ICMLA 2011*, Honolulu, USA, 2011.

[18] Erik Schmidt, Jeffrey Scott, and Youngmoo Kim. Feature learning in dynamic environments: Modeling the acoustic structure of musical emotion. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, October 8-12 2012.

[19] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2003.

[20] Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In *Rumelhart, D. E. and McClelland, J. L., editors, Parallel Distributed Processing: Volume 1: Foundations*, pages 194–281. MIT Press, Cambridge, 1986.

[21] Son Tran and Artur Garcez. Extraction from deep belief networks. In *ICML 2012 Representation Learning Workshop*, Edinburgh, July 2012.

[22] Sebastian Stober and Andreas Nürnberger. Similarity adaptation in an exploratory retrieval scenario. In *Proceedings of AMR 2010*, Linz, Austria, Aug 2010. to appear.

[23] Sebastian Stober and Andreas Nürnberger. An experimental comparison of similarity adaptation approaches. In *Proceedings of AMR*, Barcelona, Spain, Jul 2011. To appear.

[24] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. 11:3371–3408, December 2010.

[25] D. Wolff and T. Weyde. Adapting metrics for music similarity using comparative judgements. In *Proceedings of ISMIR*, 2011.

[26] Daniel Wolff and Tillman Weyde. Learning Music Similarity from Relative User Ratings. *Information Retrieval*, 2013.

[27] Daniel Wolff, Sebastian Stober, Andreas Nürnberger, and Tillman Weyde. A systematic comparison of music similarity adaptation approaches. In *Proceedings of ISMIR*, 2012.