# Representation Learning for Cloud Classification

David Bernecker, Christian Riess, Vincent Christlein, Elli Angelopoulou, and Joachim Hornegger

Pattern Recognition Lab, Department of Computer Science,
Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
{david.bernecker, christian.riess, vincent.christlein, elli.angelopoulou,
joachim.hornegger}@cs.fau.de

**Abstract** Proper cloud segmentation can serve as an important precursor to predicting the output of solar power plants. However, due to the high variability of cloud appearance, and the high dynamic range between different sky regions, cloud segmentation is a surprisingly difficult task. In this paper, we present an approach to cloud segmentation and classification that is based on representation learning. Texture primitives of cloud regions are represented within a restricted Boltzmann Machine. Quantitative results are encouraging. Experimental results yield a relative improvement of the unweighted average (pixelwise) precision on a three-class problem by 11% to 94% in comparison to prior work.

## 1 Introduction

Solar power plants suffer from sudden, sharp drop-offs in the energy output when a single, thick cloud moves in front of the sun. This forces suppliers to combine each power plant with expensive batteries for backup. If it is possible to reliably predict such power drop-offs, slower backup systems can be set up, which lowers costs considerably. For predicting such sun occlusions, several groups investigate the exploitation of ground-based camera systems that observe the sky, e.g. [2, 4, 10]. Such systems aim to track the overall cloud motion in the sky.

However, to assess the impact of an occluding cloud on the generated power profile, it is very reasonable to setup predictors that incorporate the consistency and exact shape of the cloud. Towards this goal, we present a vision-based method that distinguishes sky, thin clouds and dense clouds.

Cloud segmentation is a challenging problem. Its main difficulty lies in the varying appearance and non-rigidity of clouds, as well as the high brightness close to the sun. Explicit feature design is difficult: structural features face the fact that there is barely any distinct “cloud structure”. Using color also yields severe ambiguities due to the color similarity of denser clouds with non-cloud regions [11]: no sunlight passes through dense clouds, and these clouds are only illuminated by the light reflected from the earth's surface. This leads to a bluish-grey color that is hard to distinguish from the color of the sky. The opposite situation, blue sky with the sun clearly visible, is also difficult to segment correctly. Around the sun lens flares occur, or a large glowing region is visible. Lastly, contextual information

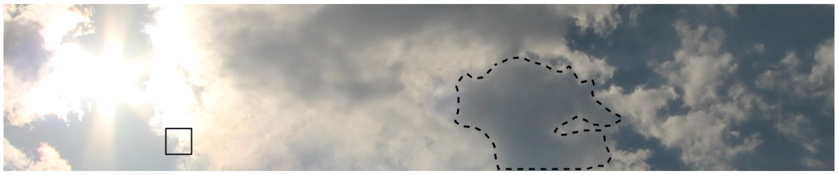

**Figure 1.** Part of a sky image showing the difficulties when segmenting or classifying clouds. The dark parts of clouds (example region dashed) and sky have a similar color. For scale reference a patch of 40 × 40 pixels is marked.

is difficult to incorporate correctly. For less overcast days, one may model dark clouds to be small dark-blue patches surrounded by white. However, with more clouds in the sky, the sky itself will only be visible as small dark-blue patches surrounded by white clouds. We are facing several such ambiguous situations.

We address these challenges by learning a representation of cloud structure from unlabeled data rather than designing features by hand. As training data, we use small image patches extracted from sky-images. The learning of the representation amounts to the training of a Restricted Boltzmann Machine (RBM) or Deep Belief Network (DBN). Since training is unsupervised, a large number of image patches can be used, as manual annotation is not required. To classify the clouds, the learned representation is treated as regular filters, i. e. sky-images are convolved with the learned 'filters' and classification is done pixel by pixel using the result of the convolution combined with the red/blue ratio as features.

The rest of the paper is organized as follows. Section 2 gives an overview over related work in cloud classification and representation learning. The concept of representation learning with Restricted Boltzmann Machines and Deep Belief Networks is introduced in Section 3. For comparison, a classical approach using handcrafted features is briefly illustrated in Section 4. The proposed use of a learned representation for cloud classification is evaluated in Section 5 and some examples of learned representations are shown and discussed in Section 5.1. This is followed by the evaluation of the actual cloud classification in Section 5.3. A short conclusion is given in Section 6.

## 2 Related Work

Recently, cloud segmentation or classification raised attention by meteorologists and researchers working on the integration of renewable energies into the existing power grid. The baseline method employs fixed thresholding on the ratio of the red and the blue color channel [12]. In a more recent work, Chow *et al.* [4] expand this approach by using a reference model trained on clear blue sky. Their model incorporates the effects of aerosols and airmass, which influence the perceived color of the sky near the horizon. Other approaches, e. g. the one by Richards *et al.* [11], investigate the use of texture as an additional feature. These texture features consist of combinations of the weighted average and

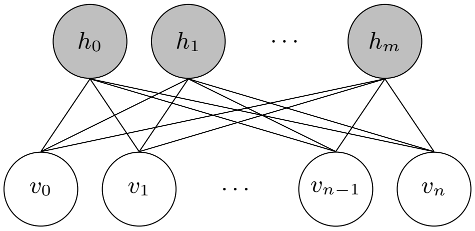

**Figure 2.** Restricted Boltzmann Machine: a neural net consisting of $n$ visible and $m$ hidden units. Connections between nodes are undirected.

basic spot and edge detectors over regions of the size of 3 × 3 pixels. In the same work, Richards *et al.* also extended the segmentation of the clouds to a classification into different cloud types. Unfortunately, the work does not provide a quantitative performance evaluation. We compare our results to the baseline method of color thresholding and a combination of color and texture features. Instead of the texture features used by Richards *et al.*, we are using the slightly more sophisticated Gabor filters.

Since Hinton *et al.* [6] presented an efficient approach for learning representations with Deep Belief Networks by pretraining each layer of the network individually, representation learning has become one of the most active research topics in image classification. For example, methods from representation learning have been used to improve the classification rates on several large-scale classification problems like MNIST [8]. For a recent survey article over various methods for representation learning and their applications one may refer to [1]. In our work, we focus on using Restricted Boltzmann Machines and Deep Belief Networks for learning representations. Note that Restricted Boltzmann Machines, which form a building block for DBNs, were originally invented under the name of "Harmonium" by Smolensky *et al.* [13] in 1986.

## 3 Representation Learning

Representation learning aims to learn a suitable representation from large amounts of unlabeled data. We present two variations of artificial neural networks, the Restricted Boltzmann Machine and the Deep Belief Network, and the fast training algorithm Contrastive Divergence introduced by Hinton *et al.* [6]. These are used to learn a representation of cloud structure. For a more extensive introduction to their principles and training, please refer to Hinton *et al.* [6, 7].

### 3.1 Restricted Boltzmann Machines (RBM)

RBMs are neural networks, where the neurons form a bipartite graph (cf. Fig. 2) and connections between nodes are undirected. The two layers of the net are referred to as visible and hidden, their nodes are denoted as $v_i$ and $h_j$, respectively.

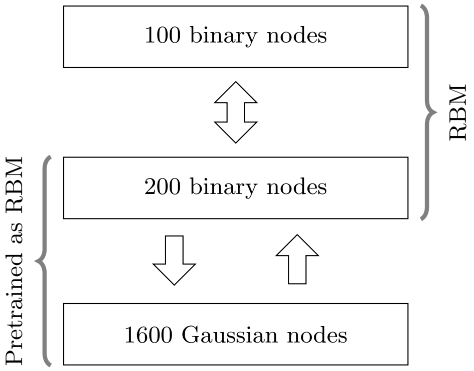

**Figure 3.** Schematic of the DBN configuration that is used. The two lower levels are pretrained as an RBM. Afterwards the weights are fixed and the layers are treated as a regular neural net. The top two layers form an RBM with undirected connections.

The number of visible nodes is determined by the size of the input patch, while the number of hidden nodes can be chosen freely. We are choosing binary nodes in the hidden layer, and Gaussian nodes [15] for the visible layer.

The activation probability for the binary hidden nodes is given by:

$$P(h_j = 1|v) = \mathrm{LS}\left(b_j + \sum_{i=1}^{n} w_{i,j} v_i\right) \ , \tag{1}$$

where $n = |v|$ is the number of visible nodes, LS denotes the logistic sigmoid function, $b_j$ denotes the bias for the visible node $h_j$ and $w_{i,j}$ are the individual weights.

In contrast, Gaussian nodes are linear nodes with an independent Gaussian noise. Thus, for input patches coming from natural images this type of nodes is more suitable. Besides the bias $a_i$, each node has a standard deviation $\sigma_i$ associated with it. Instead of learning the standard deviation during training, it is easier to first normalize the training data to have zero mean and unit variance. Given the values of the hidden nodes, the value of a Gaussian visible node is then

$$v_i = a_i + \sum_{j=1}^{m} w_{i,j} h_j + \hat{n}_i \tag{2}$$

where $m$ denotes the number of hidden nodes and $\hat{n}_i$ is Gaussian noise, i. e. $\hat{n} \sim N(0, 1)$ .

### 3.2 Contrastive Divergence (CD-k)

Hinton *et al.* [6] introduced *Contrastive Divergence* as an efficient learning algorithm for RBMs and DBNs. The general learning rule for RBMs is

$$\frac{\partial \log p(v)}{\partial w_{ij}} = \langle v_i h_j \rangle_{\mathrm{data}} - \langle v_i h_j \rangle_{\mathrm{model}} \ , \tag{3}$$

where for $\langle v_i h_j \rangle_{\text{data}}$ the values of the visible nodes are clamped to the current training data, and $\langle v_i h_j \rangle_{\text{model}}$ represents the current learned model of the data. To compute this value, in theory a Markov chain with the current training sample needs to be started and then Gibbs sampling (i. e. sampling the hidden nodes and then reconstructing the visible nodes) is performed until the chain has reached its equilibrium state. As this is not feasible for training, Hinton *et al.* [6] introduced Contrastive Divergence, where $\langle v_i h_j \rangle_{\text{model}}$ is replaced with $\langle v_i h_j \rangle_{\text{recon}}$. The latter is calculated by performing $k$ steps of Gibbs sampling, starting from the current training sample. The calculated updates for the weight matrix are then multiplied by a factor $\epsilon \ll 1$, the learning rate.

We use a slight modification of this algorithm, where the Markov chain is persistent and is only restarted after each full iteration over all training samples. Furthermore, we use momentum to speed up the training and we encourage a sparse activation of the hidden nodes by introducing a penalty term if the average activation probability of a hidden node is above or below the target activation probability (sparsity target) of 10%. A comprehensive description of the variants of the Contrastive Divergence, and practical hints for training RBMs can be found in the work by Hinton [7].

### 3.3 Deep Belief Networks (DBN)

An extension of RBMs are DBNs, neural networks consisting of more than two layers. While the top two layers have undirected connections and can be interpreted as an RBM, the lower levels function as a regular neural network. Similar to the RBMs, our DBNs consist of a visible layer of Gaussian nodes, while all hidden layers use binary nodes. For training, the greedy, layer-wise strategy is used [6], where starting from the bottom, each pair of layers is seen as an RBM and is trained using Contrastive Divergence. After the training of such an RBM has finished, its weights are set fixed. Next, the activation probabilities for the current hidden nodes for all training samples are calculated and this is used for training the RBMs of the next two layers.

## 4 Handcrafted Features using Gabor Filters

For comparison, we briefly describe an algorithm to cloud classification using handcrafted features. We follow the idea of Richards *et al.* [11], but use Gabor filters [5] to describe the cloud texture. With a total of 32 filters, this filter bank spans over different scales ($\sigma_x$, $\sigma_y$), frequencies ($F$) and orientations ($\theta$).

$$G(x, y) = N \cdot \exp\left(\frac{x_r^2}{2\sigma_x^2} + \frac{y_r^2}{2\sigma_y^2}\right) \cdot \cos(2\pi\, F\, x_r) \tag{4}$$

where $x_r = x \sin\theta + y \cos\theta$ and $y_r = x \cos\theta - y \sin\theta$ denote rotated coordinates of $x$ and $y$. Fig. 4 shows an example of different Gabor filters, generated by using different orientations and frequencies.

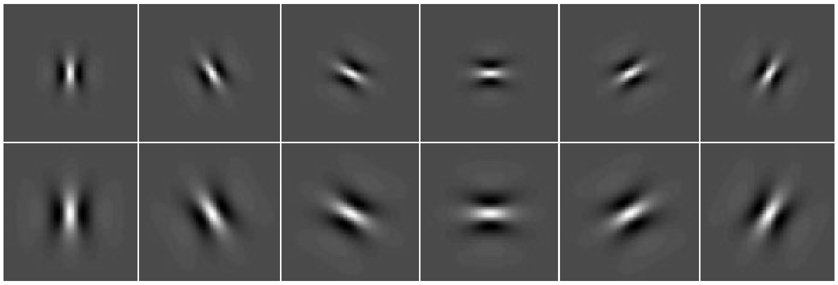

**Figure 4.** Gabor Filters in different orientations and frequencies.

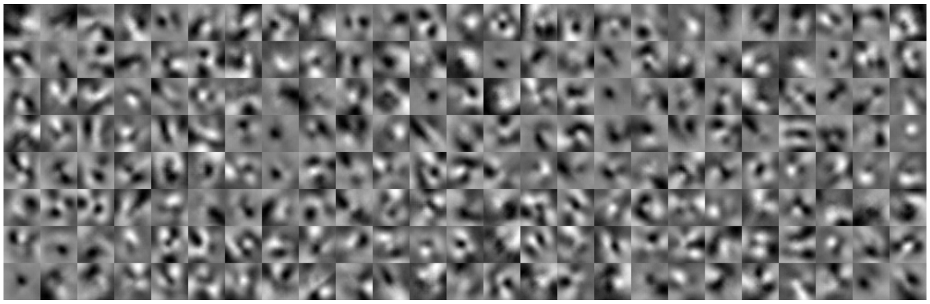

**Figure 5.** Representation of cloud structure, obtained by training a RBM with 200 hidden nodes with CD-1.

## 5 Evaluation

For representation learning, the usual pattern recognition pipeline of feature extraction, supervised training and subsequent testing of the classifier has to be slightly extended. First, the RBM or DBN is trained in an unsupervised manner (see Sec. 5.1). Then, the weight matrices of the RBM or DBN are interpreted as filters. The filter output is used as a feature vector within a classical pattern recognition pipeline. The evaluation protocol and error metrics are presented in Section 5.2, and quantitative results in Section 5.3.

### 5.1 Training of Learned Representations

We are considering two learned representations, RBM and DBN. The representations learned by an RBM with 200 hidden nodes is shown in Fig. 5. Fig. 6 depicts the learned representation using a three layer DBN that extends the RBM and has 100 hidden nodes in the top layer. The figures show that cloud structures are well represented and encourage to use these learned representations as discriminative features for cloud structures.

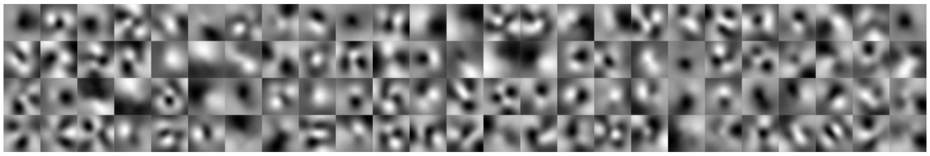

**Figure 6.** Representation of cloud structure, obtained by training a three layer DBN. The RBM weights shown in Fig. 5 were used for the two lower levels, while the uppermost layer with 100 hidden nodes was trained with CD-1.

**Training Data** For training the RBM or DBN, we use small patches of sky images of 40 × 40 pixels (cf. Fig. 1). These are extracted from 320 sky images that were captured over the course of one day. Out of these images, a total of about 160 000 non-overlapping patches are extracted and used for training.

**Normalization** Two normalization steps are applied. Since our aim is to learn a representation of cloud structure, the absolute brightness is not important. Hence, we subtract from each patch its mean intensity. As a second step, pixel variation needs to be normalized. This comes from the fact that Gaussian visible nodes are required to model patches of natural images (cf. Section 3.1). To avoid learning the standard deviation of each node during training, each pixel of the patches is normalized over all training samples to have zero mean and unit variance.

**RBM and DBN Parameters** The RBM is trained with 200 binary hidden nodes using the Contrastive Divergence algorithm with one step of Gibbs sampling (CD-1). The learning rate $\epsilon$ is set to 0.00001, momentum $m$ to 0.9 and the sparsity target to 0.1 with a penalty factor of 1.0. Training consists of 750 epochs, and the weight updates are calculated for mini-batches of size 100.

This RBM is also used as the lowest layer of the three layer DBN. For training the latter, the following parameters are used: $\epsilon = 0.001$, $m = 0.9$, sparsity target 0.1 and sparsity penalty 0.01. Mini-batch size is 100, and the training algorithm runs for 500 epochs.

### 5.2 Evaluation Protocol and Error Metrics

We treat cloud classification as a three class problem, with the classes *sky*, *white cloud* and *thick cloud*. This classification task exhibits some particular characteristics. A large number of training samples is available when classifying each pixel of the data. However, the distribution of the label occurrences is strongly skewed among the large number of samples. The number of sky pixels is several orders of magnitude higher than the number of thick cloud pixels. For training, it is important that this spread between label occurrences is not too large, since our goal is a classifier that can safely distinguish between the individual classes. Thus, we set the maximum spread between the number of class occurrences for training to two, i. e. the number of samples in one class is at most by a factor two higher than the number of the other classes. For the classes with a higher number of occurrences, samples are randomly drawn from the total number of samples of the class.

Since we are interested in a performance measure that is independent of the imbalances in class occurrence, we are using the unweighted average recall (UAR) and unweighted average precision (UAP). These measures are commonly used in speech recognition [14]. UAR and UAP are simply the unweighted averages of the *recall* and *precision* values of all classes. Precision and recall are defined as:

$$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}\,, \qquad \mathrm{RE} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}\ \ , \tag{5}$$

where TP denotes the number of true positives, FP the number of false positives and FN the number of false negatives. Precision (PR) relates to the classifier's ability to identify positive results, while recall (RE) shows the probability that a class is identified correctly. Consequently, UAP and UAR are computed as

$$\mathrm{UAP} = \frac{1}{K}\sum_{k=1}^{K}\mathrm{PR}_k\,, \qquad \mathrm{UAR} = \frac{1}{K}\sum_{k=1}^{K}\mathrm{RE}_k\ \ , \tag{6}$$

where $K$ denotes the number of classes, i. e. in our case $K = 3$.

### 5.3 Results and Discussion

Overall, we compare four different types of features for cloud classification. As a baseline method we are using color information as a single feature. This color information is the ratio of the red and blue channel and widely used in cloud segmentation [4,12]. In this ratio, the distinction between white clouds and blue sky is the greatest. The second feature set is a combination of the color feature with Gabor filters. This set is closely related to the work of Richards *et al.* [11], who also use color and texture features. The third feature set contains the color feature and in addition the filters learned with an RBM that are shown in Fig. 5. The last feature set combines the color feature and filters learned with a three layer DBN.

For the classification a Random Forest$^{\mathrm{TM}}$ [3] classifier is used. A total of about 500 000 samples are extracted from several sky images and labelled manually. 350 000 samples are used for training the Random Forest. During training cross-validation is used to find suitable parameters. The remaining 150 000 unseen samples are used for the evaluation. Particular care was taken that samples from one image were taken either for training, validation or testing, to avoid overfitting. A different set of sky images that was acquired at a different time and place, is used for learning the representation of cloud structures (i. e. training the RBM). That ensures that the learned representation is not directly fitted to the occurring clouds in the test set.

All classification results are summarized in Table 1. The upper part contains results on the features evaluated individually, the lower part on the combinations

**Table 1.** Overall classification results for the three class cloud classification problem using the unweighted average precision (UAP) and unweighted average recall (UAR) for the test set. The tables show the results using one single feature set (top) and the color ratio feature in combination with Gabor filters and the learned representations (bottom).

| Single features | UAP | UAR |
|---|---|---|
| color (baseline) | 0.85 | 0.83 |
| Gabor | 0.69 | 0.63 |
| RBM | 0.77 | 0.75 |
| DBN | 0.81 | 0.80 |

| Feature combinations | UAP | UAR |
|---|---|---|
| color + Gabor | 0.89 | 0.86 |
| color + RBM | 0.92 | 0.88 |
| color + DBN | 0.94 | 0.90 |

of features. We set color features as a baseline [12]. Gabor features, RBM and DBN in the top of Table 1 are computed on grayscale images. It turns out, that the baseline color features yield high UAP and UAR of 85% and 83%, resp., justifying its common use for cloud classification. Gabor filters perform worst, indicating that structural information alone does not solve this task well. RBM and DBN perform slightly better than Gabor. The slight increase between RBM and DBN meets the expectation that increasing the number of layers leads to a more descriptive learned representation.

In Table 1 (bottom) evaluation results are shown on combinations of color and texture (or structure) features. Interestingly, the combination of color and Gabor filters only slightly increases the performance over using solely color. This is in agreement with the texture classification work by Mäenpää *et al.* [9], who state that color and texture filters are not able to exploit their complementary information to full extend. Overall, the best result is obtained by combining the color feature and the representation learned with a three layer DBN. Both UAP and UAR are improved considerably. The relative improvement over the baseline is 11% and 8%, respectively.

## 6 Conclusion

For local, short-term predictions of solar power production, researchers investigate ground-based camera systems to monitor cloud movement in the sky. Cloud segmentation and thickness classification can act as important components in such monitoring systems.

We propose features from representation learning for a novel method for cloud segmentation and classification. In contrast to traditional, fixed texture filters, the proposed descriptors for cloud structure are learned from unlabeled sky images. We evaluate these learned features by posing the segmentation of clouds as a three-class classification problem. Quantitative results demonstrate that representation learning outperforms classical texture features like Gabor filters by a large margin. The best performance is achieved using a Deep Belief Network, which yields a relative improvement over the baseline by 11%.

## References

1. Bengio, Y., Courville, A., Vincent, P.: Representation Learning: A Review and New Perspectives. ArXiv e-prints, arXiv:1206.5538 (Jun 2012)
2. Bernecker, D., Riess, C., Angelopoulou, E., Hornegger, J.: Towards Improving Solar Irradiance Forecasts with Methods from Computer Vision. In: Computer Vision in Applications Workshop (2012)
3. Breiman, L.: Random forests. Machine learning 45, 5–32 (2001)
4. Chow, C.W., Urquhart, B., Lave, M., Dominguez, A., Kleissl, J., Shields, J., Washom, B.: Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed. Solar Energy 85(11), 2881–2893 (Nov 2011)
5. Daugman, J.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. Optical Society of America, Journal, A: Optics and Image Science 2(7), 1160–1169 (1985)
6. Hinton, G., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. Neural computation 1554, 1527–1554 (2006)
7. Hinton, G.: A Practical Guide to Training Restricted Boltzmann Machines. Tech. rep., UTML TR 2010–003 (2010)
8. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
9. Mäenpää, T., Pietikäinen, M.: Classification with color and texture: jointly or separately? Pattern Recognition 37(8), 1629–1640 (Aug 2004)
10. Marquez, R., Coimbra, C.F.: Intra-hour DNI forecasting based on cloud tracking image analysis. Solar Energy 91, 327–336 (Oct 2013)
11. Richards, K., Sullivan, G.: Estimation of cloud cover using colour and texture. British Machine Vision Conference (1992)
12. Shields, J.E., Karr, M.E., Burden, A.R., Johnson, R.W., Mikuls, V.W., Streeter, J.R., Hodgkiss, W.S.: Research toward Multi-Site Characterization of Sky Obscuration by Clouds, Final Report for Grant N00244-07-1-009, Marine Physical Laboratory, Scripps Institution of Oceanography, University of California San Diego. Tech. rep., Technical Note 274, DTIS (Stinet) File ADA126296 (2009)
13. Smolensky, P.: Information processing in dynamical systems: Foundations of harmony theory. In: Rumelhart, D., McClelland, J. (eds.) Parallel distributed processing: explorations in the microstructure of cognition, vol. 1, pp. 194–281. MIT Press Cambridge, MA, USA ©1986 (1986)
14. Vinciarelli, A., Burkhardt, F., Son, R.V., Weninger, F., Eyben, F., Bocklet, T., Mohammadi, G., Weiss, B., Telekom, D., Laboratories, A.G.: The INTERSPEECH 2012 Speaker Trait Challenge. Proc Interspeech (2012)
15. Welling, M., Rosen-Zvi, M., Hinton, G.: Exponential family harmoniums with an application to information retrieval. Advances in neural information processing systems 17, 1481–1488 (2005)