Learning Features for Tissue Classification with the Classification Restricted Boltzmann Machine

Gijs van Tulder[1] and Marleen de Bruijne[1,2]

[1] Biomedical Imaging Group Rotterdam
Erasmus MC University Medical Center, the Netherlands
[2] Image Group, Department of Computer Science
University of Copenhagen, Denmark

Abstract. Performance of automated tissue classification in medical imaging depends on the choice of descriptive features. In this paper, we show how restricted Boltzmann machines (RBMs) can be used to learn features that are especially suited for texture-based tissue classification. We introduce the convolutional classification RBM, a combination of the existing convolutional RBM and classification RBM, and use it for discriminative feature learning. We evaluate the classification accuracy of convolutional and non-convolutional classification RBMs on two lung CT problems. We find that RBM-learned features outperform conventional RBM-based feature learning, which is unsupervised and uses only a generative learning objective, as well as often-used filter banks. We show that a mixture of generative and discriminative learning can produce filters that give a higher classification accuracy.

1 Introduction

Most machine learning applications – for classification and other tasks, in medical image analysis and elsewhere – do not work directly on the input data, but use a higher-level representation instead. For example, when training a classifier for images, the pixel values are mapped to features that simplify classification. Most conventional approaches used in medical image analysis, such as filter banks of Gaussian derivatives, wavelets or SIFT, are predesigned, general methods that are not tuned for a specific problem or dataset.

Feature learning or representation learning [1] provides an alternative to predesigned filters, because it learns a new representation from the data. Ideally, this data-derived representation discards irrelevant information and preserves only those details that are useful for the intended task. By varying the objective function of the feature learning method, it might be possible to tailor the features to a specific application, such as classification. Because these features have been optimized for a specific classification problem, they may provide a classification result that is better than that of predesigned filter banks.

In this paper, we discuss the restricted Boltzmann machine (RBM), a representation learning method that is popular in computer vision but still little-used

B. Menze et al. (Eds.): MCV 2014, LNCS 8848, pp. 47–58, 2014.
The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-13972-2_5

in medical image analysis. An RBM is a probabilistic graphical model that learns the probability distribution of the input data and a latent representation of that data. Because the size of this latent representation is limited, the RBM learns a concise representation that still captures most of the important information.

There are two ways to use RBMs in a classification problem. The standard RBM is an unsupervised model without label information. It learns a representation that can be used as the input for an external classifier. This approach has previously been used to model lung CT [2] and brain MR [3] images. The second option is a classification RBM [4], an extension of the standard RBM that includes labels and can be used for classification. Like standard RBMs, classification RBMs can be trained with a purely generative objective that optimizes the joint probability distribution of inputs and labels. Classification RBMs, however, can also be trained with a discriminative objective that optimizes the posterior probability of the label. A discriminative objective can improve classification results, because it helps the RBM to focus on modeling the inter-class variation that is relevant for classification. An RBM trained with only a generative learning objective might waste effort on modeling the intra-class variation in the data, which does not improve the classification. For example, in tissue classification in medical images, the model should represent the subtle differences between tissue types, rather than the more obvious differences between patients.

Specifically designed to model images, convolutional RBMs [5–8] use a weight-sharing approach borrowed from convolutional networks. Convolutional weight-sharing reduces the number of weights and adds some translational invariance to the model. Like standard RBMs, convolutional RBMs are unsupervised models. We introduce the convolutional classification RBM, a combination of the convolutional RBM with a classification RBM. We use this convolutional classification RBM in our texture classification experiments. To our knowledge, the combination of convolution and classification RBMs has not been investigated before, and the application of classification RBMs is new within medical imaging.

In this paper, we evaluate convolutional and non-convolutional RBMs as classifiers and as feature learners, on two lung CT classification problems. In particular, we are interested in the influence of the label information and the learning objective on the classification performance. We test if the classification RBM learns better features than the standard RBM. To do this, we compare the results of the standard RBM and the classification RBM, for different mixtures of discriminative and generative learning.

2 Restricted Boltzmann Machines

Standard RBM. A restricted Boltzmann machine (RBM) models the probability distribution over a set of hidden nodes $\mathbf{h}$ and visible nodes $\mathbf{v}$. We use binary hidden nodes, $h_j \in \{0, 1\}$, and real-valued visible nodes with a Gaussian distribution [9]. The joint probability distribution $P(\mathbf{v}, \mathbf{h})$ is determined by a set of weights and biases. Each visible node v_i has an undirected connection with weight W_{ij} to each hidden node h_j. Each visible node v_i has a bias b_i and each hidden node h_j has a bias c_j. These parameters define the energy function

$$E\left(\mathbf{v},\mathbf{h}\right) = \sum_{j} \frac{\left(v_i - b_i\right)^2}{2\sigma_i^2} - \sum_{i,\,j} \frac{v_i}{\sigma_i} W_{ij} h_j - \sum_{j} c_j h_j \ , \tag{1}$$

where σ_i is the standard deviation of the Gaussian noise of visible node i. The joint distribution of the input $\mathbf{v}$ and hidden representation $\mathbf{h}$ is defined as

$$P\left(\mathbf{v},\mathbf{h}\right) = \frac{\exp\left(-E\left(\mathbf{v},\mathbf{h}\right)\right)}{Z}, \tag{2}$$

where Z is a normalization constant. The conditional probabilities for the hidden nodes given the visible nodes and vice versa are

$$P\left(h_j \left| \mathbf{v} \right.\right) = \mathrm{sigm}(\sum_{i} W_{ij} v_i + c_j) \text{ and} \tag{3}$$

$$P\left(v_i \left| \mathbf{h} \right.\right) = \mathcal{N}(\sum_{j} W_{ij} h_j + b_i, \ \ \sigma_i), \tag{4}$$

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

Classification RBM. The classification RBM [4] extends the standard RBM with binary nodes that encode the label of the input. Each label node represents one class, with $y_k = 1$ if the sample belongs to class k and 0 otherwise. Like the visible nodes, the label nodes have a bias d_k and are connected to each of the hidden nodes, with weight U_{kj} connecting label node y_k to hidden node h_j. The energy function of a classification RBM with Gaussian visible nodes is

$$E\left(\mathbf{v},\mathbf{h},\mathbf{y}\right) = \sum_{j} \frac{\left(v_i - b_i\right)^2}{2\sigma_i^2} - \sum_{i,\,j} \frac{v_i}{\sigma_i} W_{ij} h_j - \sum_{j} c_j h_j - \sum_{k,\,j} y_k U_{kj} h_j - \sum_{k} d_k y_k \ . \tag{5}$$

The energy function defines the distribution

$$P\left(\mathbf{v},\mathbf{h},\mathbf{y}\right) = \frac{\exp\left(-E\left(\mathbf{v},\mathbf{h},\mathbf{y}\right)\right)}{Z} \tag{6}$$

and the conditional probabilities

$$P\left(h_j \left| \mathbf{v},\mathbf{y} \right.\right) = \mathrm{sigm}(\sum_{i} W_{ij} v_i + \sum_{k} U_{kj} y_k + c_j) \text{ and} \tag{7}$$

$$P\left(y_k \left| \mathbf{h} \right.\right) = \mathrm{sigm}(\sum_{j} U_{kj} h_j + c_k). \tag{8}$$

The visible and label nodes are not connected, so $P\left(v_i \left| \mathbf{h} \right.\right)$ is unchanged from the standard RBM. The posterior probability for classification is

$$P\left(y \left| \mathbf{v} \right.\right) = \frac{\exp\Big(d_y + \sum_j \mathrm{softplus}\left(c_j + U_{jy} + \sum_i W_{ij} v_i\right)\Big)}{\sum_{y^*} \exp\Big(d_{y^*} + \sum_j \mathrm{softplus}\left(c_j + U_{y^*j} + \sum_i W_{ij} v_i\right)\Big)}, \tag{9}$$

where $\mathrm{softplus}\left(x\right) = \log\left(1 + \exp\left(x\right)\right)$.

Learning Objectives. The standard RBM optimizes the generative learning objective $\log P(\mathbf{v}_t)$, the probability distribution of each input image t. The classification RBM can be trained with the generative learning objective $\log P(\mathbf{v}_t, y_t)$, which optimizes the joint probability distribution of the input image and the label. A classification RBM can also be trained with the discriminative objective $\log P(y_t \,|\mathbf{v}_t\,)$, which only optimizes the classification and does not optimize the representation of the input image. Larochelle et al. [4] suggest a hybrid objective

$$\beta \log P(\mathbf{v}_t, y_t) + (1-\beta) \log P(y_t \,|\mathbf{v}_t\,), \tag{10}$$

where $\beta \in [0,1]$ is the amount of generative learning. We will use this objective with different values for β in our feature learning experiments.

Learning the Weights. RBMs are probabilistic models, so when given an input, it is necessary to sample the activation of all nodes to find the new state of the model. Gibbs sampling provides a way to make this sampling more efficient. Given the visible and label nodes, the new state of the hidden nodes can be sampled using the distribution $p(\mathbf{h}_t \,|\mathbf{v}_t, y_t\,)$. Then, keeping the hidden nodes fixed, the new activation of the visible and label nodes can be sampled from $p(\mathbf{v}_t, y_t \,|\mathbf{h}_t\,)$. This can be repeated for several iterations, until the model converges to a stable state. Gibbs sampling forms the basis for contrastive divergence [9], a method that provides an efficient approximation for the gradient-based updates to the weights. We use contrastive divergence and stochastic gradient descent to optimize the weights in our RBMs.

Convolutional RBM. Designed to model images, convolutional RBMs [5–8] use the weight-sharing approach from convolutional neural networks. Unlike convolutional neural networks, convolutional RBMs are generative models and can be trained in the same way as standard RBMs. In a convolutional RBM, the connections share weights in a pattern that resembles convolution, with M convolutional filters $\mathbf{W}_m$ that connect hidden nodes arranged in M feature maps $\mathbf{h}_m$ (Fig. 1). The connections between the visible nodes and the hidden nodes in map m use the weights from convolution filter $\mathbf{W}_m$, such that each hidden node is connected to the visible nodes in its receptive field. The visible nodes share one bias b; all hidden nodes in map m share the bias c_m. With the convolution operator $*$ we define the probabilities

$$P\left(h_{ij}^m \,|\mathbf{v}\,\right) = \operatorname{sigm}\big((\tilde{\mathbf{W}}_m * \mathbf{v})_{ij} + c_m\big) \text{ and} \tag{11}$$

$$P(v_{ij} \,|\mathbf{h}\,) = \mathcal{N}\Big(\big(\sum_m \mathbf{W}_m * \mathbf{h}_m\big)_{ij} + b,\;\; 1\Big), \tag{12}$$

where $\tilde{\mathbf{W}}_m$ is the horizontally and vertically flipped filter $\mathbf{W}_m$, and $\cdot_{ij}$ denotes the pixel on location $(i,\, j)$.

Convolutional RBMs can produce unwanted border effects when reconstructing the visible layer, because the visible units near the borders are only connected to a few hidden nodes. We pad our patches with pixels from neighboring patches, and keep the padding pixels fixed during the iterations of Gibbs sampling.

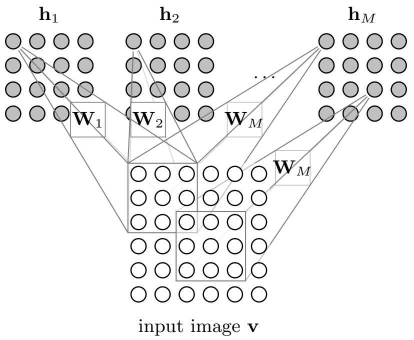

Fig. 1. Schematic view of weight sharing in a convolutional RBM. The hidden nodes are grouped in feature maps $\mathbf{h}_1, \ldots, \mathbf{h}_M$. The weights of the connections are shared using the convolutional filters $\mathbf{W}_1, \ldots, \mathbf{W}_M$.

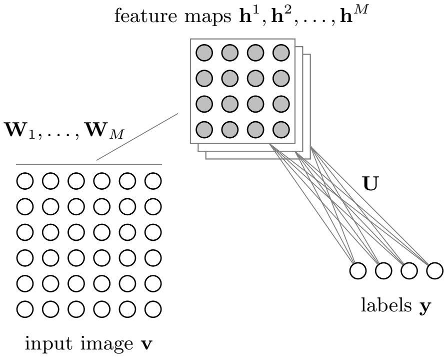
feature maps $\mathbf{h}^1, \mathbf{h}^2, \ldots, \mathbf{h}^M$
$\mathbf{W}_1, \ldots, \mathbf{W}_M$
$\mathbf{U}$
labels $\mathbf{y}$
input image $\mathbf{v}$

Fig. 2. Schematic view of a convolutional classification RBM. Convolution of the input image $\mathbf{v}$ with the filter $\mathbf{W}_m$ gives the hidden feature map $\mathbf{h}^m$. The activation of each feature map is summed and connected with the label nodes $\mathbf{y}$ with weights $\mathbf{U}$.

Convolutional Classification RBM. We introduce a convolutional classification RBM that includes visible, hidden and label nodes (Fig. 2) and can be trained in a discriminative way. The visible nodes are connected to the hidden nodes using convolutional weight-sharing, as in the convolutional RBM, and the hidden nodes are connected to the label nodes, as in the classification RBM. In our patch-based texture classification problem, the exact location of a feature inside the patch is not relevant, so we use shared weights to connect the hidden nodes and the label nodes. All connections from a label node y_k to a hidden node h_{ij}^m in map m share the weight U_{km}. The activation probabilities are

$$P\left(y_k \,|\mathbf{h}\right) = \mathrm{sigm}\Big(\sum_m U_{ym} \sum_{i,j} h_{ij}^m + d_k\Big) \text{ and } \tag{13}$$

$$P\left(h_{ij}^m \,|\mathbf{y}\right) = \mathrm{sigm}\big(\big(\tilde{\mathbf{W}}_m * \mathbf{v}\big)_{ij} + \sum_k U_{km} y_k + c_m\big). \tag{14}$$

The probability for the visible nodes is unchanged from the convolutional RBM.

3 Experiments

3.1 Datasets and Problems

Airways. We show results for two classification problems on lung CT images. In the first dataset we classify airway and non-airway patches, to detect airway center points as a preprocessing step for airway extraction algorithms. We use 40 scans of 20 patients from the Danish Lung Cancer Screening Trial [10]. The voxel size is approximately 0.78 × 0.78 × 1 mm. Using the output of an existing

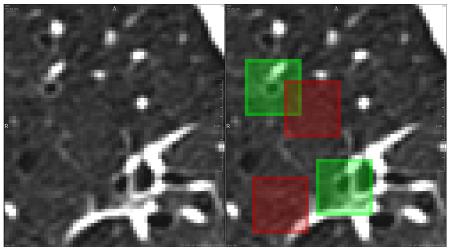

Fig. 3. First dataset. In the airway dataset, we extract patches at the airway centerline (green) and non-airway samples (red) close to the airway.

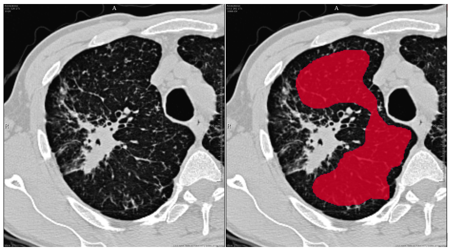

Fig. 4. Second dataset. Example from the interstitial lung disease scans. The annotation (right) shows an ROI (red) marked as micronodules.

segmentation algorithm [11] to find the airways (Fig. 3), we extracted patches of 16×16 pixels at the center point of airways with a diameter of 16 pixels or less. For each airway patch, we create a non-airway sample by extracting a patch at a random point just outside the outer airway wall. We selected a random subset of 500 patches per scan. We use 15 subjects (30 scans, 15 000 patches) as our training set and 5 subjects (10 scans, 5 000 patches) for testing.

Lung Tissue. The second dataset is a larger, publicly available dataset on interstitial lung diseases (see [12] for a description). In this texture classification problem with scans from 73 patients, we do patch-wise classification of five types of lung tissue (healthy tissue: 22%, emphysema: 3%, ground glass: 16%, fibrosis: 15%, micronodules: 44%). The resolution varies between $0.4 - 1$ mm, with a slice thickness of $1 - 2$ mm and inter-slice spacing of $10 - 15$ mm. The dataset provides hand-drawn 2D ROIs with labels (Fig. 4) for a subset of slices in each scan. Following other work on this dataset [2,13], we extracted patches of 32×32 pixels along a grid with a 16-pixel overlap. We include a patch if at least 75% of the pixels belong to the same class. We use 48 patients (8 165 patches) for training and 25 others (4 265 patches) for testing.

3.2 Evaluation Procedure

We trained the RBMs with various learning rates until convergence. We report the test accuracy of the RBM with the best accuracy on the training set. For the airway dataset, the patches are centered at the center point of the airways. This means that the features do not have to be translation-invariant, so we can use a non-convolutional RBM. For the lung tissue classification, translation-invariance is required, so we use convolutional RBMs on this dataset. After learning the filters, we convolve the input patches with the filters and use an adaptive binning method to generate one histogram per filter. We trained support vector machines (SVMs) with linear and radial basis function (RBF) kernels on the concatenated histograms, with 5-fold cross-validation on the training sets to optimize the SVM parameters and number of histogram bins.

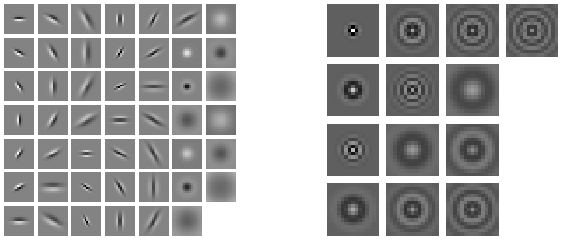

Fig. 5. Two filter banks: Leung-Malik (left) and Schmid (right), generated with the code from http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html.

3.3 Baselines

We compare the results of RBM-learned filters with several baselines (Table 1). On the airway data, we compare with SVMs trained on the raw input pixels. Raw input pixels do not work for the lung tissue classification problem, because the texture classification needs translation-invariant features. We therefore use convolution with random filters as a baseline for the tissue set. As the second set of baseline results, we use two of the standard filter banks discussed by Varma and Zisserman [14] (Fig. 5). The filter bank of Leung and Malik [15] is a set of Gaussian filters and derivatives, with 48 filters of 16 × 16 pixels. The filter bank of Schmid [16] has 13 filters of 16 × 16 pixels with rotation-invariant Gabor-like patterns. For the airway dataset, we multiply the image patch with each filter to get the feature vector. For the lung tissue data, we apply the filters using the same procedure with convolution and adaptive histograms that we also use for the RBM-learned filters.

Table 1 summarizes the baseline results for both datasets. On the airway dataset, an RBF SVM trained with Leung-Malik filters gives a test accuracy of 90.5%, against 91.3% for an RBF SVM trained on the raw intensity values. On the lung tissue dataset, Leung-Malik filters have the best performance at 64.1%.

For the lung tissue dataset, two earlier publications also give an indication of expected classification results, although the differences in patch extraction and train/test splits make a direct comparison with our results impossible. Depeursinge et al. [13] use a set of near-affine-invariant feature descriptors based on isotropic wavelet filters. Li et al. [2] use unsupervised, non-convolutional RBMs at three scales to learn convolution filters. Both studies show a multi-class classification accuracy of approximately 77%.

3.4 Results

Airways. Fig. 8 shows the results of the RBM classification and of the SVM classification on airway data, with non-convolutional RBMs with different numbers of hidden nodes (example filters shown in Fig. 6). The best RBM reaches an accuracy on the test set of a little more than 89%. Discriminative learning improves the RBM classification accuracy, although the difference becomes smaller if the number of hidden nodes is larger.

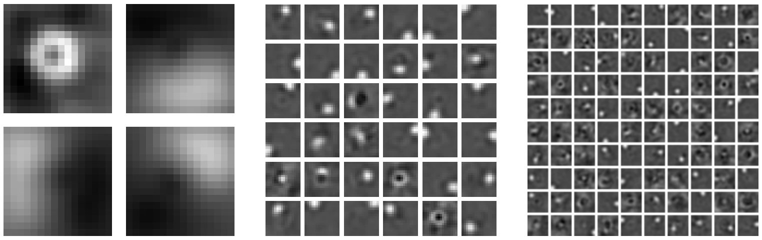

Fig. 6. Three filter sets learned from the airway data: 4, 36 or 100 filters of 16 × 16 pixels, learned with a mix of discriminative and generative learning ($\beta = 0.01$).

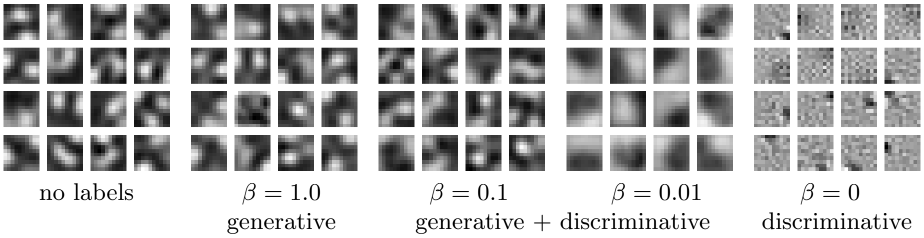

Fig. 7. Filters learned on the lung tissue data. Sets of 16 filters of 10 × 10 pixels, learned by non-convolutional RBMs on subpatches. RBM without labels (leftmost) and classification RBMs for various mixtures of generative and discriminative learning.

Table 1. SVM classification accuracy on the test set with baseline features.

Features	raw pixels		Leung-Malik		Schmid	
SVM kernel	linear	RBF	linear	RBF	linear	RBF
Airways	89.30	91.28	90.10	90.50	83.88	85.40
Lung tissue	not applicable		64.88	64.06	56.90	54.81

The results of SVMs using the RBM-learned features show a similar pattern. At best, an RBF SVM with RBM-learned features achieves an accuracy of 90.3% (36 filters, pure discriminative learning), comparable to that of the best filter bank. In general, more discriminative learning produces features that give a higher accuracy in the SVM. This difference is strongest if the number of hidden nodes is small, when it is most important to specialize.

Lung Tissue. We trained non-convolutional RBMs and convolutional RBMs on the lung tissue dataset, with different filter sizes and numbers of filters. (Training a large convolutional RBM took up to three days.) On the lung tissue dataset, we found the best SVM classification accuracy ($77-78\%$) with filters learned with a mixture of generative and discriminative learning (Fig. 9). The best learned filters outperformed both the standard filter banks and the random filters. Adding label information, even with pure generative learning, often improved the results.

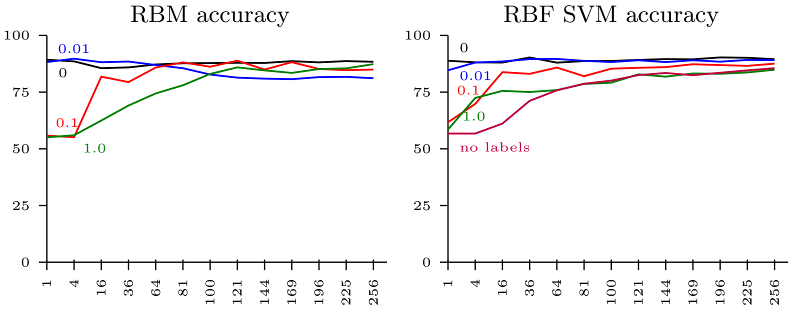

Fig. 8. Airway classification accuracy of the non-convolutional RBM (left) and the RBF SVM (right) trained on the RBM-learned filters. The plot shows the effect of different values for β on the classification accuracy (vertical axis).

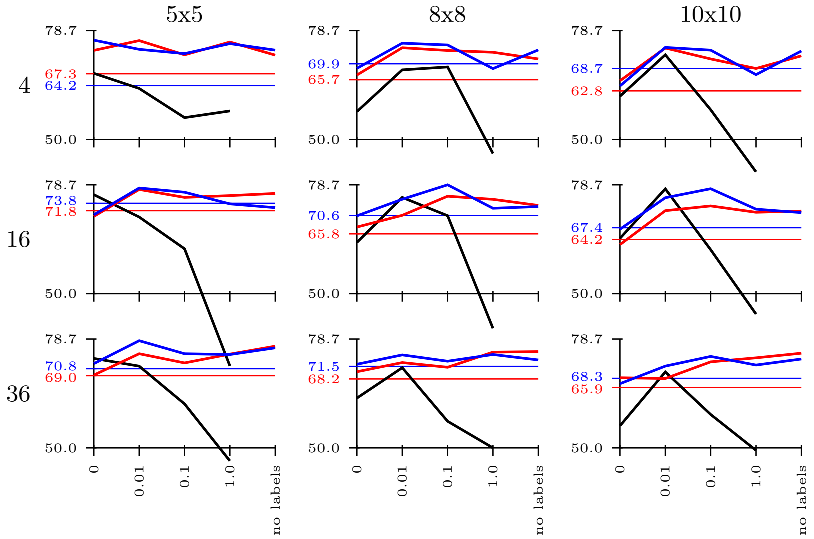

Fig. 9. Lung tissue classification results with the convolutional RBM, for different filter sizes (columns) and numbers of filters (rows). Each subplot shows the RBM classification accuracy (black) and linear SVM accuracy (blue) and RBF SVM accuracy (red) for different β (horizontal axis). Horizontal blue and red lines show the SVM accuracy with random filters.

Filters learned with pure discriminative or pure generative learning generally showed a lower performance than those learned with a mix. In a visual inspection, we found that filters learned with a bit of generative learning seem to have more visible structure (see, for example, the filters in Fig. 7).

We can also evaluate the classification accuracy of the RBM itself. The convolutional RBM had a maximum accuracy on the test set of 77.6%, with 16 filters of 10 × 10 pixels. In most cases, pure discriminative learning gave the RBM with the best classification result.

Table 2. Results of McNemar's and Durkalski's tests comparing the classification accuracy on the lung tissue dataset, for linear SVMs with RBM-learned filters versus random filters. 'RBM' and 'random' indicate the method with the highest mean accuracy. The p-values indicate the significance of the difference, according to both tests.

Filter size	5×5 pixels			8×8 pixels			10×10 pixels		
Filters	4	16	36	4	16	36	4	16	36
$\beta = 0$	RBM	random	RBM	random	random	RBM	random	random	random
McNemar	0.0000	0.0000	0.0434	0.0921	0.9202	0.4242	0.0000	0.5826	0.0671
Durkalski	0.1778	0.0867	0.6809	0.6273	0.1614	0.0864	0.2242	0.1848	0.2135
$\beta = 0.01$	RBM	RBM	RBM	RBM	RBM	RBM	RBM	RBM	RBM
McNemar	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Durkalski	0.0014	0.2411	0.1680	0.2087	0.4775	0.5432	0.2614	0.6208	0.4958
$\beta = 0.1$	RBM	RBM	RBM	RBM	RBM	RBM	RBM	RBM	RBM
McNemar	0.0000	0.0000	0.0000	0.0000	0.0000	0.0413	0.0000	0.0000	0.0000
Durkalski	0.2340	0.5124	0.3770	0.4374	0.9676	0.1917	0.2661	0.2302	0.8062
$\beta = 1.0$	RBM	random	RBM	random	RBM	RBM	random	RBM	RBM
McNemar	0.0000	0.8027	0.0000	0.0826	0.0049	0.0000	0.0195	0.0000	0.0000
Durkalski	0.1222	0.1732	0.9608	0.7709	0.5714	0.3185	0.7284	0.7086	0.6697
no labels	RBM	random	RBM	RBM	RBM	RBM	RBM	RBM	RBM
McNemar	0.0000	0.0995	0.0000	0.0000	0.0005	0.0154	0.0000	0.0000	0.0000
Durkalski	0.0576	0.1390	0.5584	0.7621	0.4975	0.0985	0.1520	0.5676	0.5046

For a statistical evaluation of the RBM-learned filters, we compare the mean accuracy of SVMs with RBM-learned features with the results of SVMs with random features. We use two statistical tests to determine the significance of the results. McNemar's test is the usual test to compare two classifiers [17]. McNemar's test, however, assumes that the samples are independent, and in our case this assumption might not hold: our samples are patches from 25 scans, and there might be correlations between samples from the same scan. There are several extensions of McNemar's test that allow for within-cluster correlations. We follow the advice by Yang et al. [18] and use Durkalski's test [19].

Table 2 shows the results of McNemar's test and Durkalski's test for all RBM configurations. In most cases, the average performance with RBM-learned features was higher than the performance of the best-performing baseline, the random filters. According to McNemar's test (which is too optimistic), many of these differences would be significant. In cases where random filters had a higher mean accuracy than RBM-learned filters, the p-value is generally higher and not significant. Durkalski's test is far more conservative: almost none of the differences are significant. But although the individual differences are not significant according to these tests, the differences are mostly in favor of the RBM-learned features. For example, features learned with mixed strategies ($\beta = 0.1$ and $\beta = 0.01$) had the highest mean accuracy for all configurations. This promising result suggests that RBM-based feature learning may provide an advantage.

4 Discussion and Conclusion

We have shown how the classification RBM can be used to learn useful features for medical image analysis, with a mean classification accuracy that is better than or close to that of other methods. Using label information and discriminative learning led to an improved performance. Although no significant differences were found in these relatively small datasets, the results suggest that RBM-learned filters have an advantage over random filters. It is interesting to see that random filters performed better than two standard filter banks. The surprising performance of random filters has been noted in the literature before [20].

Because the lung tissue data that we used in our experiments is highly anisotropic and has 2D annotations only in selected slices, we chose to do feature learning and classification in 2D. However, given the right training data, the methods discussed in this paper could easily be extended to 3D.

The accuracy of the classification RBM in most of our experiments came close to that of an SVM classifier, but the SVMs usually achieved better results. We have two possible explanations for this difference. One, there may be an imbalance in the energy function, because there are many more connections between the visible and hidden nodes than between the hidden and label nodes. This might lead the model to over-optimize the representation at the cost of a higher classification error. Using a discriminative learning objective might help to prevent this. Two, although our results suggest that combining representation and classification can help to learn better features, this combination might make it harder to optimize the classification. Since the representation changes in each update step, the classification weights need to change as well. It might help to add a final training phase that only optimizes the classification weights, while keeping the representation fixed.

Feature learning is usually done with a purely generative learning objective, which favors a representation that gives the most faithful description of the data. But this is not always the representation that is best for the goal of the system. For example, classification may need more features that model inter-class variation than features that model intra-class variation. We therefore argue that it is important to use the right learning objective for feature learning. In this paper, we found that the classification accuracy of SVMs using RBM-learned features can be improved by including label information in the model and by adding discriminative learning to the learning objective of the RBM.

Acknowledgment. This research is financed by the Netherlands Organization for Scientific Research (NWO).

References

1. Bengio, Y., Courville, A., Vincent, P.: Representation Learning: A Review and New Perspectives. Technical report, Université de Montréal (2012)

2. Li, Q., Cai, W., Feng, D.D.: Lung Image Patch Classification with Automatic Feature Learning. In: EMBC. (2013)
3. Brosch, T., Tam, R.: Manifold Learning of Brain MRIs by Deep Learning. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013, Part II. LNCS, vol. 8150, pp. 663–640. Springer, Heidelberg (2013)
4. Larochelle, H., Mandel, M., Pascanu, R., Bengio, Y.: Learning Algorithms for the Classification Restricted Boltzmann Machine. Journal of Machine Learning Research **13** (2012) 643–669
5. Desjardins, G., Bengio, Y.: Empirical Evaluation of Convolutional RBMs for Vision. Technical report, Université de Montréal (2008)
6. Norouzi, M., Ranjbar, M., Mori, G.: Stacks of Convolutional Restricted Boltzmann Machines for Shift-Invariant Feature Learning. In: CVPR. (2009)
7. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: ICML. (2009)
8. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Unsupervised learning of hierarchical representations with convolutional deep belief networks. Communications of the ACM **54**(10) (October 2011)
9. Hinton, G.E.: A Practical Guide to Training Restricted Boltzmann Machines. Technical report, University of Toronto (2010)
10. Pedersen, J.H., Ashraf, H., Dirksen, A. et al.: The Danish Randomized Lung Cancer CT Screening Trial - Overall Design and Results of the Prevalence Round. Journal of Thoracic Oncology **4**(5) (May 2009)
11. Petersen, J., Nielsen, M., Lo, P., Saghir, Z., Dirksen, A., de Bruijne, M.: Optimal graph based segmentation using flow lines with application to airway wall segmentation. In: IPMI. (2011)
12. Depeursinge, A., Vargas, A., Platon, A., Geissbuhler, A., Poletti, P.A., Müller, H.: Building a reference multimedia database for interstitial lung diseases. Computerized medical imaging and graphics **36**(3) (April 2012)
13. Depeursinge, A., Van de Ville, D., Platon, A., Geissbuhler, A., Poletti, P.A., Müller, H.: Near-affine-invariant texture learning for lung tissue analysis using isotropic wavelet frames. Transactions on Information Technology in Biomedicine (July 2012)
14. Varma, M., Zisserman, A.: A statistical approach to material classification using image patch exemplars. Transactions on Pattern Analysis and Machine Intelligence **31**(11) (November 2009)
15. Leung, T., Malik, J.: Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. International Journal of Computer Vision **43**(1) (2001)
16. Schmid, C.: Constructing models for content-based image retrieval. In: CVPR. (2001)
17. Dietterich, T.G.: Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural computation **10**(7) (September 1998) 1895–1923
18. Yang, Z., Sun, X., Hardin, J.W.: A note on the tests for clustered matched-pair binary data. Biometrical journal **52**(5) (October 2010) 638–52
19. Durkalski, V.L., Palesch, Y.Y., Lipsitz, S.R., Rust, P.F.: Analysis of clustered matched-pair data. Statistics in medicine **22**(15) (August 2003) 2417–2428
20. Saxe, A.M., Koh, P.W., Chen, Z., Bhand, M., Suresh, B., Ng, A.Y.: On Random Weights and Unsupervised Feature Learning. In: ICML. (2011)