Weakly Supervised Learning of Mid-Level Features with Beta-Bernoulli Process Restricted Boltzmann Machines

Roni Mittelman, Honglak Lee, Benjamin Kuipers, Silvio Savarese
Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor
{rmittelm,honglak,kuipers,silvio}@umich.edu

Abstract

The use of semantic attributes in computer vision problems has been gaining increased popularity in recent years. Attributes provide an intermediate feature representation in between low-level features and the class categories, leading to improved learning on novel categories from few examples. However, a major caveat is that learning semantic attributes is a laborious task, requiring a significant amount of time and human intervention to provide labels. In order to address this issue, we propose a weakly supervised approach to learn mid-level features, where only class-level supervision is provided during training. We develop a novel extension of the restricted Boltzmann machine (RBM) by incorporating a Beta-Bernoulli process factor potential for hidden units. Unlike the standard RBM, our model uses the class labels to promote category-dependent sharing of learned features, which tends to improve the generalization performance. By using semantic attributes for which annotations are available, we show that we can find correspondences between the learned mid-level features and the labeled attributes. Therefore, the mid-level features have distinct semantic characterization which is similar to that given by the semantic attributes, even though their labeling was not provided during training. Our experimental results on object recognition tasks show significant performance gains, outperforming existing methods which rely on manually labeled semantic attributes.

1. Introduction

Modern low-level feature representations, such as SIFT and HOG, have had great success in visual recognition problems, yet there has been a growing body of work suggesting that the traditional approach of using only low-level features may be insufficient. Instead, significant performance gains can be achieved by introducing an intermediate set of features that capture higher-level semantic concepts beyond the plain visual cues that low-level features offer [29, 27, 13]. One popular approach to introducing such mid-level features is to use semantic attributes [7, 16, 9]. Specifically, each category can be represented by a set of semantic attributes, where some of these attributes can be shared by other categories. This facilitates the transfer of information between different categories and allows for improved generalization performance.

Typically, the attribute representation is obtained using the following process. First, a set of concepts is defined by the designer, and each instance in the training set has to be labeled with the presence or absence of each attribute. Subsequently, a classifier is trained for each of the attributes using the constructed training set. Furthermore, as was reported in [7], some additional feature selection schemes which utilize the attribute labels may be necessary in order to achieve satisfactory performance. Obtaining the semantic attribute representation is clearly a highly labor-intensive process. Furthermore, it is not clear how to choose the constituent semantic concepts for problems in which the shared semantic content is less intuitive (e.g., activity recognition in videos [22]).

One approach to learning a semantic mid-level feature representation is based on latent Dirichlet allocation (LDA) [2], which uses a set of topics to describe the semantic content. LDA has been very successful in text analysis and information retrieval, and has been applied to several computer vision problems [3, 20]. However, unlike linguistic words, visual words often do not carry much semantic interpretation beyond basic appearance cues. Therefore, the LDA has not been very successful in identifying mid-level feature representations [21].

Another line of work is the deep learning approach (see [1] for a survey), such as deep belief networks (DBNs) [12], which tries to learn a hierarchical set of features from unlabeled and labeled data. It has been shown that features in the upper levels of the hierarchy capture distinct semantic concepts, such as object parts [19]. The DBNs can be effectively trained in a greedy layer-wise procedure using the restricted Boltzmann machine [25] as a building block. The RBM is a bi-partite undirected graphical model that is

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.68

capable of learning a dictionary of patterns from the unlabeled data. By expanding the RBM into a hierarchical representation, relevant semantic concepts can be revealed at the higher levels. RBMs and their extension to deeper architectures have been shown to achieve state-of-the-art results on image classification tasks (e.g., [26, 14]).

In this work, we propose to combine the powers of topic models and DBNs into a single framework. We propose to learn mid-level features using the *replicated softmax* RBM (RS-RBM), which is an undirected topic model applied to bag-of-words data [24]. Unlike other topic models, such as LDA, the RS-RBM can be expanded into a DBN hierarchy by stacking additional RBM layers with binary inputs on-top of the first RS-RBM layer. Therefore, we expect that features in higher levels can capture important semantic concepts that could not be captured by standard topic models with only a single layer (e.g., LDA). To our knowledge, this work is the first application of the RS-RBM to object recognition problems.

As another contribution, we propose a new approach to include class labels in training an RBM-like model. Although unsupervised learning can be effective in learning useful features, there is a lot to be gained by allowing some degree of supervision. To this end, we develop a new extension of the RBM which promotes a class-dependent use of dictionary elements. This can be viewed as a form of multi-task learning [4], and as such tends to improve the generalization performance.

The idea underlying our approach is to define an undirected graphical model using a factor graph with two kinds of factors; the first is an RBM-like type, and the second is related to a Beta-Bernoulli process (BBP) prior [28, 23]. The BBP is a Bayesian prior that is closely related to the Indian buffet process [10], and it defines a prior for binary vectors where each coordinate can be viewed as a feature for describing the data. The BBP has been used to allow for multi-task learning under a Bayesian formulation of sparse coding [30]. Our approach, which we refer to as the *Beta-Bernoulli Process Restricted Boltzmann Machine (BBP-RBM)*, permits an efficient inference scheme using Gibbs sampling, akin to the inference in the RBM. Parameter estimation can also be effectively performed using Contrastive Divergence. Our experimental results on object recognition show that the proposed model outperforms other baseline methods, such as LDA, RBMs, and previous state-of-the-art methods using attribute labels.

In order to analyze the semantic content that is captured by the mid-level features learned with the BBP-RBM, we used the datasets from [7] which include annotations of manually specified semantic attributes. By using the learned features to predict each of the labeled attributes in the training set, we found the correspondences between the learned mid-level features and the labeled attributes. We performed localization experiments where we try to predict the bounding boxes of the mid-level features in the image and compare them to their corresponding attributes. We demonstrate that our method can localize semantic concepts like snout, skin, and furry, even though no information about these attributes was used during the training process.

The rest of the paper is organized as follows. In Section 2, we provide background on the RBM and the BBP. In Section 3, we describe the DBN architecture for object recognition. In Section 4, we formulate the BBP-RBM model, and in Section 5, we evaluate the BBP-RBM experimentally. Section 6 concludes the paper.

2. Preliminaries

In this section, we provide background on RBMs and the BBP. We review two forms of the RBM which are both used in this work: the first assumes binary observations, and the second is the RS-RBM which uses word count observations.

2.1. RBM with binary observations

The RBM [25] defines a joint probability distribution over a hidden layer $\mathbf{h} = [h_1, \ldots, h_K]^T$, where $h_k \in \{0, 1\}$, and a visible layer $\mathbf{v} = [v_1, \ldots, v_N]^T$, where $v_i \in \{0, 1\}$. The joint probability distribution can be written as

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h})). \tag{1}$$

Here, the energy function of $\mathbf{v}, \mathbf{h}$ is defined as

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{h}^T \mathbf{W} \mathbf{v} - \mathbf{b}^T \mathbf{h} - \mathbf{c}^T \mathbf{v}, \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{K \times N}$, $\mathbf{b} \in \mathbb{R}^K$, $\mathbf{c} \in \mathbb{R}^N$ are parameters.

It is straightforward to show that the conditional probability distributions take the form

$$\begin{aligned} p(h_k = 1|\mathbf{v}) &= \sigma\big(\sum_i w_{k,i} v_i + b_k\big), & (3) \\ p(v_i = 1|\mathbf{h}) &= \sigma\big(\sum_k w_{k,i} h_k + c_i\big), & (4) \end{aligned}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function. Inference can be performed using Gibbs sampling, alternating between sampling the hidden and visible layers. Although computing the gradient of the log-likelihood of training data is intractable, the Contrastive Divergence [11] approximation can be used to approximate the gradient.

2.2. The Replicated Softmax RBM

The RBM can be extended to the case where the observations are word counts in a document [24]. The word counts are transformed into a vector of binary digits, where the number of 1's for each word in the document equals its word count. A single hidden layer of a binary RBM then connects to each of these binary observation vectors (with weight sharing), which allows for modeling of the word counts. The model can be further simplified such that it deals with the word count observations directly, rather than with the intermediate binary vectors. Specifically, let N denote the number of words in the dictionary, and let v_i ($i = 1, \ldots, N$) denote the number of times word i appears in the document, then the joint probability distributions of the binary hidden layer $\mathbf{h}$ and the observed word counts $\mathbf{v}$ is of the same form as in Equations (1) & (2), where the energy of $\mathbf{v}, \mathbf{h}$ is defined as

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{h}^T \mathbf{W} \mathbf{v} - D \mathbf{b}^T \mathbf{h} - \mathbf{c}^T \mathbf{v}, \tag{5}$$

and $D = \sum_{i=1}^N v_i$ is the total word count in a document.

Inference is performed using Gibbs sampling, where the posterior for the hidden layer takes the form

$$p(h_k = 1|\mathbf{v}) = \sigma\big(\sum_i w_{k,i} v_i + D b_k\big). \tag{6}$$

Sampling from the posterior of the visible layer is performed by sampling D times from the following multinomial distribution:

$$p_i = \frac{\exp(\sum_{k=1}^K h_k w_{k,i} + c_i)}{\sum_{i=1}^N \exp(\sum_{k=1}^K h_k w_{k,i} + c_i)}, \; i = 1, ..., N, \tag{7}$$

and setting v_i to the number of times the index i appears in the D samples.

Parameter estimation is performed in the same manner as the case of the RBM with binary observations.

2.3. Beta-Bernoulli process

BBP is a Bayesian generative model for binary vectors, where each coordinate can be viewed as a feature for describing the data. In this work, we use a finite approximation to the BBP [23] which can be described using the following generative model. Let $f_1, \ldots, f_K \in \{0, 1\}$ denote the elements of a binary vector, then the BBP generates f_k according to

$$\begin{aligned} \pi_k &\sim \text{Beta}(\alpha/K, \beta(K-1)/K), \\ f_k &\sim \text{Bernoulli}(\pi_k), \end{aligned} \tag{8}$$

where α, β are positive constants (hyperparameters), and we use the notation $\pi = [\pi_1, \ldots, \pi_K]^T$. Equation (8) implies that if π_k is close to 1 then f_k is more likely to be 1, and vice versa. Since the Beta and Bernoulli distributions are conjugate, the posterior distribution for π_k also follows a Beta distribution. In addition, for a sufficiently large K and reasonable choices of α and β, most π_k will be close to zero, which implies a sparsity constraint on f_k.

Furthermore, by drawing a different π_k for each class, we can impose a unique class-specific sparsity structure, and such a prior allows for multi-task learning. The BBP has been used to allow for multi-task learning under a Bayesian formulation of sparse coding [30, 5]. The multi-task paradigm promotes sharing of information between related groups, and therefore can lead to improved generalization performance. Motivated by this observation, we propose an extension of the RBM that incorporates a BBP-like factor and extend to a deeper architecture.

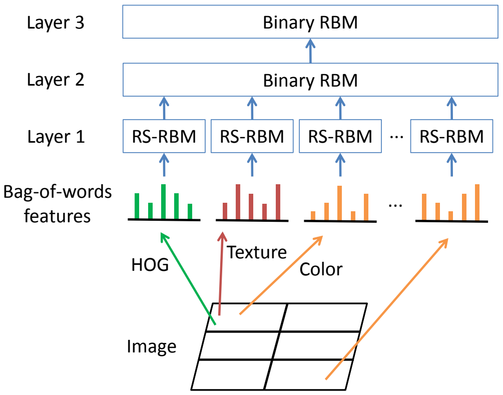


Figure 1. A pipeline for constructing mid-level features. For both the RS-RBM and the binary RBM, we propose their extensions by incorporating the Beta-Bernoulli process factor potentials.

3. The object recognition scheme

Our mid-level feature extraction scheme is described in Figure 1. We use a low-level feature extraction method following [7], where the image is first partitioned into a 3 × 2 grid, and HOG, texture, and color features are extracted from each of the cells, as well as from the entire image. In order to obtain the bag-of-words representation, we first compute the histogram over the visual words, and then obtain the word counts by multiplying each histogram with a constant (we used the constant 200 throughout this work) and rounding the numbers to the nearest integer values.

The word counts are used as the inputs to RS-RBMs (or *BBP-RS-RBMs* which we describe in Section 4), where different RS-RBM units are used for each of the histograms. The binary outputs of all the RS-RBM units are concatenated and fed into a binary RBM (or a binary BBP-RBM) at the second layer. The outputs of the hidden units of the second layer are then used as input to the third layer binary RBM, and similarly to any higher layers. Training the DBN is performed in a greedy layer-wise fashion, starting with the first layer and proceeding in the upward direction [12].

Each of the RS-RBM units independently captures important patterns which are observed within its defined feature type and spatial extent. The binary RBM in the second layer captures higher-order dependencies between the different histograms in the first layer. The binary RBMs in higher levels could model further high-order dependencies, which we hypothesize to be related to some semantic concepts. In Section 5, we find associations between the learned features and manually specified semantic attributes.

The feature vector which is used for classification is obtained by concatenating the outputs of all the hidden units from all the layers of the learned DBN. Given a training set, we compute the feature vector for every instance and train a multi-class classifier. Similarly, for every previously unseen test instance, we compute its feature vector and classify it using the trained classifier.

4. The BBP-RBM

In this section, we develop the BBP-RBM, both when assuming that all the training examples are unlabeled, and also when each example belongs to one of C classes. We refer to these two versions as *single-task BBP-RBM* and *multi-task BBP-RBM*, respectively. The single-task version can be considered as a new approach to introduce sparsity into the RBM formulation, which is an alternative to the common approach of promoting sparsity through regularization [18]. It is also related to “dropout”, which randomly sets individual hidden units to zeros during training and has been reported to reduce overfitting when training deep convolutional neural networks [15]. The BBP-RBM uses a factor graph formulation to combine two different types of factors: the first factor is related to the RBM, and the second factor is related to the BBP. Combining these factors together leads to an undirected graphical model for which we develop efficient inference and parameter estimation schemes.

4.1. Proposed Model

We define a binary selection vector $\mathbf{f} = [f_1, \ldots, f_K]^T$ that is used to choose which of the K hidden units to activate. Our approach is to define an undirected graphical model in the form of a factor graph with two types of factors, as shown in Figure 2(a) for the single-task case and Figure 2(b) for the multi-task cases. The first factor is obtained as an unnormalized RBM-like probability distribution which includes the binary selection variables $\mathbf{f}$:

$$g_a(\mathbf{v}, \mathbf{h}, \mathbf{f}) = \exp(-E(\mathbf{v}, \mathbf{h}, \mathbf{f})), \tag{9}$$

where the energy term takes the form

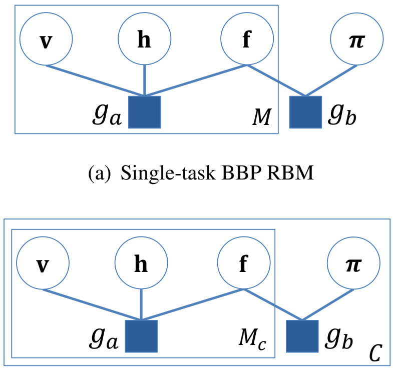
$$E(\mathbf{v}, \mathbf{h}, \mathbf{f}) = -(\mathbf{f} \odot \mathbf{h})^T \mathbf{W}\mathbf{v} - \mathbf{b}^T(\mathbf{f} \odot \mathbf{h}) - \mathbf{c}^T\mathbf{v}, \tag{10}$$

and $\odot$ denotes element-wise vector multiplication.

The second factor is obtained from the BBP generative model (described in Equation (8)) as follows:

$$g_b(\{\mathbf{f}^{(j)}\}_{j=1}^M, \pi) = \prod_{k=1}^K \pi_k^{\sum_{j=1}^M f_k^{(j)}} (1-\pi_k)^{\sum_{j=1}^M (1-f_k^{(j)})} \times \pi_k^{\alpha/K-1}(1-\pi_k)^{\beta(K-1)/K-1}, \tag{11}$$

where j denotes the index of the training sample, and M denotes the number of training samples.



(a) Single-task BBP RBM

(b) Multi-task BBP RBM

Figure 2. The factor graphs for the BBP-RBM. g_a and g_b are the two factor types, and M denotes the total number of training samples. C denotes the number of classes in the training set, and M_c denotes the number of training instances belonging to class c.

Using the factor graph description in Figure 2(a), the probability distribution for the single-task BBP-RBM takes the form

$$p(\{\mathbf{v}^{(j)}, \mathbf{h}^{(j)}, \mathbf{f}^{(j)}\}_{j=1}^M, \pi) \propto g_b(\{\mathbf{f}^{(j)}\}_{j=1}^M, \pi) \times \prod_{j=1}^M g_a(\mathbf{v}^{(j)}, \mathbf{h}^{(j)}, \mathbf{f}^{(j)}). \tag{12}$$

Using the factor graph description in Figure 2(b), we have that the joint probability distribution for the multi-task case takes the form

$$p(\{\mathbf{v}^{(j)}, \mathbf{h}^{(j)}, \mathbf{f}^{(j)}\}_{j=1}^M, \{\pi^{(c)}\}_{c=1}^C) \propto \prod_{c=1}^C \left(g_b(\{\mathbf{f}^{(j_c)}\}_{j=1}^{M_c}, \pi^{(c)}) \prod_{j_c=1}^{M_c} g_a(\mathbf{v}^{(j_c)}, \mathbf{h}^{(j_c)}, \mathbf{f}^{(j_c)}) \right), \tag{13}$$

where C denotes the number of different classes in the training set, and we use the notation j_c to denote the unique index of the training instance which belongs to class c, and M_c denotes the number of training instances which belong to class c.

4.2. Inference

Similarly to the RBM, inference in the BBP-RBM can be performed using Gibbs sampling. We only provide the posterior probability distributions for the multi-task case, since the single-task can be obtained as a special case by setting $C = 1$. Sampling from the joint posterior probability distribution of $\mathbf{h}$ and $\mathbf{f}$ can be performed using

$$p(h_k^{(j_c)} = x, f_k^{(j_c)} = y | -) \propto \begin{cases} \pi_k^{(c)} e^{\delta_k^{(j_c)}}, & x = 1, y = 1 \\ \pi_k^{(c)}, & x = 0, y = 1 \\ 1 - \pi_k^{(c)}, & x = 0, y = 0 \\ 1 - \pi_k^{(c)}, & x = 1, y = 0 \end{cases} \tag{14}$$

where “$-$” denotes all other variables, and we define $\delta_k^{(j_c)} = \sum_i w_{k,i} v_i^{(j_c)} + b_k$ for binary inputs, or $\delta_k^{(j_c)} = \sum_i w_{k,i} v_i^{(j_c)} + Db_k$ for word count observations.

The posterior probability for $\pi^{(c)}$ takes the form

$$p(\pi_k^{(c)}|-) = \tag{15}$$
$$\mathrm{Beta}(\alpha/K + \sum_{j_c=1}^{M_c} f_k^{(j_c)}, \beta(K-1)/K + \sum_{j_c=1}^{M_c} (1 - f_k^{(j_c)})).$$

Sampling from the posterior of the visible layer is performed in a similar way that was discussed in Section 2 for the RBM with either binary or word count observations, where the only difference is that $\mathbf{h}$ is replaced by $\mathbf{f} \odot \mathbf{h}$.

From Equation (14), we observe that if $\pi_k^{(j_c)} = 1$ then the BBP-RBM reduces to the standard RBM, since the posterior probability distribution for $h_k^{(j_c)}$ becomes $p(h_k^{(j_c)} = 1|-) = \sigma(\delta_k^{(j_c)})$ (i.e., the standard RBM has the same posterior probability for $h_k^{(j_c)}$).

4.3. Parameter estimation

Using the property of conditional expectation, we can show that the gradient of the log-likelihood of $\mathbf{v}^{(j_c)}$ with respect to the parameter $\theta \in \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ takes the form

$$\frac{\partial \log p(\mathbf{v}^{(j_c)})}{\partial \theta} = \mathbb{E}_{\pi^{(c)}} \Bigg[- \mathbb{E}_{\mathbf{h},\mathbf{f},|\pi^{(c)},\mathbf{v}^{(j_c)}} \Bigg[\frac{\partial}{\partial \theta} E(\mathbf{v}^{(j_c)}, \mathbf{h}, \mathbf{f}) \Bigg] + \mathbb{E}_{\mathbf{h},\mathbf{f},\mathbf{v}|\pi^{(c)}} \Bigg[\frac{\partial}{\partial \theta} E(\mathbf{v}, \mathbf{h}, \mathbf{f}) \Bigg] \Bigg]. \tag{16}$$

The expression cannot be evaluated analytically; however, we note that the first inner expectation does admit an analytical expression, whereas the second inner expectation is intractable. We propose to use an approach similar to Contrastive Divergence to approximate Equation (16). First, we sample $\pi^{(c)}$ using Gibbs sampling, and then use a Markov chain Monte-Carlo approach to approximate the second inner expectation. The batch version of our approach is summarized in Algorithm 1. In practice, we use an online version where we update the parameters incrementally using mini-batches. We also re-sample the parameters $\{\pi^{(c)}\}_{c=1}^{C}$ only after a full sweep over the training set is finished.

4.4. Object recognition using the BBP-RBM

When using the BBP-RBM in the DBN architecture described in Figure 1, there is an added complication of dealing with the variable π since it cannot be marginalized efficiently. Our solution is to train each layer of a BBP-RBM as described in the previous section. However, when computing the output of the hidden units to be fed into the consecutive layer, we choose $\pi_k^{(c)} = 1, \forall c = 1, ..., C, k = 1, ..., K$, which corresponds to the output of a standard RBM (as explained in Section 4.2). Using this approach, we avoid the issues which would otherwise arise during the recognition stage (i.e., class labels are unknown for test examples).

Algorithm 1 Batch Contrastive Divergence training for the multi-task BBP-RBM.

```
Input: Previous samples of $\{\pi^{(c)}\}_{c=1}^{C}$, training samples $\{\mathbf{v}^{(j)}\}_{j=1}^{M}$, and learning rate $\lambda$.
• For $c = 1, \ldots, C$, sample $\pi_{new}^{(c)}$ as follows
    1. Sample $\mathbf{h}^{(j_c)}, \mathbf{f}^{(j_c)} | \pi^{(c)}, \mathbf{v}^{(j_c)}, \forall\, j_c = 1, \ldots, M_c$ using Equation (14).
    2. Sample $\pi_{new}^{(c)}$ using Equation (15).
• For $c = 1, \ldots, C$, $j_c = 1, \ldots, M_c$
    1. Sample $\mathbf{h}^{(j_c,0)}, \mathbf{f}^{(j_c,0)} | \pi_{new}^{(c)}, \mathbf{v}^{(j_c)}$.
    2. Sample $\mathbf{v}^{(j_c,1)} | \pi_{new}^{(c)}, \mathbf{h}^{(j_c,0)}, \mathbf{f}^{(j_c,0)}$.
    3. Sample $\mathbf{h}^{(j_c,1)}, \mathbf{f}^{(j_c,1)} | \pi_{new}^{(c)}, \mathbf{v}^{(j_c,1)}$.
• Update each of the parameters $\theta \in \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ using
    $\theta \leftarrow \theta - \lambda \sum_{c=1}^{C} \sum_{j_c=1}^{M_c} \big( \frac{\partial}{\partial \theta} E(\mathbf{v}^{(j_c)}, \mathbf{h}^{(j_c,0)}, \mathbf{f}^{(j_c,0)}) - \frac{\partial}{\partial \theta} E(\mathbf{v}^{(j_c,1)}, \mathbf{h}^{(j_c,1)}, \mathbf{f}^{(j_c,1)}) \big)$
```

5. Experimental results

We evaluated the features learned by the BBP-RBM using two datasets that were developed in [7], which include annotation for labeled attributes. We refer to the two datasets as the PASCAL and Yahoo datasets. We performed object classification experiments within the PASCAL dataset and also across the two datasets (i.e., learning the BBP-RBM features using the PASCAL training set, and performing classification on the Yahoo dataset). Finally, we examined the semantic content of the features by finding correspondences between the learned features and the manually labeled attributes available for the PASCAL dataset. We also used these correspondences to perform attribute localization experiments, by predicting the bounding boxes for several of the learned mid-level features.

5.1. PASCAL and Yahoo datasets

The PASCAL dataset is comprised of instances corresponding to 20 different categories, with pre-existing splits into training and testing sets, each containing over 6400 images. The categories are: person, bird, cat, cow, dog, horse, sheep, airplane, bicycle, boat, bus, car, motorcycle, train, bottle, chair, dining-table, potted-plant, sofa, and tv/monitor. The Yahoo dataset contains 2644 images with 12 categories which are not included in the PASCAL dataset. Additionally, there are annotations for 64 attributes

	Overall			Mean per-class		
# Layers	1	2	3	1	2	3
LDA	54.0	-	-	32.1	-	-
RBM	55.5	55.9	56.5	32.6	33.6	33.8
sparse RBM	60.0	60.8	61.0	40.5	41.1	41.8
single-task BBP-RBM	61.7	62.0	61.7	42.3	42.3	41.7
multi-task BBP-RBM	62.5	63.2	63.2	42.7	45.5	46.1

Table 1. Test classification accuracy for the dataset proposed in [7] using LDA, baseline RBMs, and BBP-RBMs.

which are available for all the instances in the PASCAL and Yahoo datasets. We used the same low-level features (referred to as *base features*) which were employed in [7] and are available online. The feature types that we used are: 1000 dimensional HOG histogram, 128 dimensional color histogram, and 256 dimensional texture histogram. In [7], an eight dimensional edge histogram was used as well; however, we did not use it in our RBM and BBP-RBM based experiments since the code to extract the edge features and the corresponding descriptors were not available online. Note that not using the edge features in our methods may give an unfair disadvantage when comparing to the results in [7] and [29] that used all the base features. The HOG, color, and texture descriptors which we used are identical to [7]. When learning an RBM based model, we used 800 hidden units for the HOG histogram, 200 hidden units for the color histogram, and 300 units for the texture histogram. The number of hidden units for the upper layers was 4000 for the second layer, and 2000 for the third layer.

5.1.1 Recognition on the PASCAL dataset

In Table 1, we compare the test classification accuracy for the PASCAL dataset using features that were learned with the following methods: LDA, the standard RBM, the RBM with sparsity regularization (sparse RBM) [18], the single-task BBP-RBM, and the multi-task BBP-RBM. The LDA features were the topic proportions learned for each of the histograms (see Section 3), and we used 50 topics for each histogram. For evaluating features, we used the multi-class linear SVM [6] in all the experiments. When performing cross validation, the training set was partitioned into two sets. The first was used to learn the BBP-RBM features, and the second was used as a validation set. For both the overall classification accuracy and the mean per-class classification accuracy, the sparse RBM outperformed the standard RBM and LDA, but it performed slightly worse than the single-task BBP-RBM. This could suggest that the single-task BBP-RBM is an alternative approach to inducing sparsity in the RBM. Furthermore, the multi-task BBP-RBM outperformed all other methods, particularly for the mean per-class classification rate.[1] Adding more layers generally improved the classification performance; however, the improvement reached saturation at approximately 2-3 layers.

	Using attributes		Without attributes	
Method	Overall	Per-class	Overall	Per-class
[7]	59.4	37.7	58.5	34.3
[29] w/loss-1	62.16	46.25	58.77	38.52
[29] w/loss-2	59.15	50.84	53.74	44.04
BBP-RBM	-	-	63.2	46.1

Table 2. Comparison of test accuracy between several methods using the same dataset and low-level features used in [7]

In Table 2, we compare the classification results obtained using the multi-task BBP-RBM to the results reported in [7] and [29] for the same task. Note that the baseline methods were adapted to exploit the information from the labeled attributes (which the BBP-RBM did not use). In [7], scores from attribute classifiers were used as input for a multi-class linear SVM. In [29], the attribute classifier scores were used in a latent SVM [8] formulation, using two different loss functions (referred to as “loss-1” and “loss-2” in the table). Note that attribute annotations are very expensive to obtain, and for many visual recognition problems, such as activity recognition in videos [22], it is even harder to identify and label the semantic content that is shared by different types of classes. The results show that, even though our method did not use the attribute annotation, it significantly improved both the overall classification accuracy and the mean per-class accuracy in comparison to the baseline methods.

5.1.2 Learning new categories in the Yahoo dataset

An important aspect of evaluating the features is the degree to which they generalize well across different datasets. To this end, we used the PASCAL training set to learn the features and evaluated their performance on the Yahoo dataset. We partitioned the Yahoo dataset into different proportions of training samples and compared the performance when using the multi-task BBP-RBM and base features, respectively. Table 3 summarizes the test accuracy averaged over 10 random trials for several training set sizes. The results suggest that our method using the BBP-RBM features can recognize new categories from the Yahoo dataset with fewer training samples, as compared to using the base features.

[1] We note that a supervised version of the RBM (referred to as “discRBM” here) which regards the class label as an observation was introduced in [17]. In our experiments, the discRBM’s classification performance was not significantly better than that of the sparse RBM.

	Base features		BBP-RBM features	
Training %	Overall	Per-class	Overall	Per-class
10%	56.3	44.2	65.8	54.1
20%	61.0	48.2	70.6	60.0
30%	64.5	49.9	73.6	62.4
40%	68.3	51.4	75.3	64.8
50%	71.0	52.2	77.1	66.6
60%	71.0	52.5	78.4	68.4

Table 3. Average test classification accuracy on the Yahoo dataset when using the base features and when using the BBP-RBM features learned from the PASCAL training set.

For example, the overall classification performance with the BBP-RBM features using only 20% of the dataset for training is comparable to or better than that with the base features using 60% of the dataset for training.

5.2. Correspondence between mid-level features and semantic attributes

In this experiment, we evaluated the degree to which the features learned using the BBP-RBM demonstrate identifiable semantic concepts. For each feature and labeled attribute pair, we used the score given by Equation (3) to predict the presence of manually labeled semantic attributes in each training example and computed the area under the ROC curve over the PASCAL training data. The feature corresponding to each attribute is determined as that which has the largest area under the ROC curve. Figure 3 shows the corresponding area under the ROC curve for every attribute on the PASCAL test data (i.e., using the training set to determine the correspondences, and the test set to compute the ROC area). The area under the ROC curve obtained using attribute classifiers (linear SVMs trained using the attribute labels and the base features [7]) is also shown together. The figure shows that the learned features without using attribute labels performed reasonably well, and some learned features performed comparably to the attribute classifiers that were trained using the attribute labels. We note that all the semantic attributes were associated to features in either the second layer or the third layer in Figure 1, which supports our hypothesis that the higher levels of the DBN can capture semantic concepts.

5.2.1 Predicting attribute bounding boxes

We also performed experiments where the mid-level features corresponding to the attributes "snout", "skin", and "furry" were used to predict the bounding boxes of these attributes. For fine-grained localization, we ran simple sliding-window detection using bounding boxes of different aspect ratios on each image, and show only the first few non-overlapping windows that achieved the best scores. As shown in Figure 4, although there were some missdetections in the "skin" case, we were able to identify appropriate bounding boxes. Note that there were no bounding boxes available for these attributes in the training set (i.e., the bounding boxes were provided only for the entire objects); yet in some cases the BBP-RBM could localize the subparts of the categories which the attributes describe.

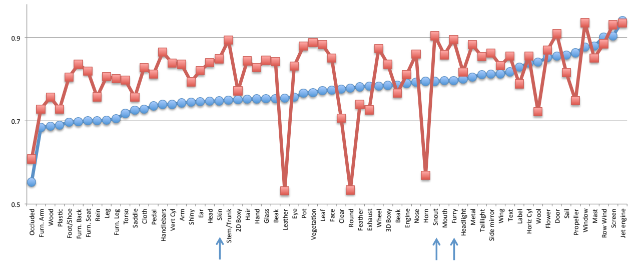


Figure 3. The area under the ROC curve of each of the 64 attributes for (1) the BBP-RBM features corresponding to labeled attributes (circles) and (2) the attribute classifiers trained using the base features (squares). The attributes that are used to predict bounding boxes in Figure 4 are marked with arrows. See text for details.

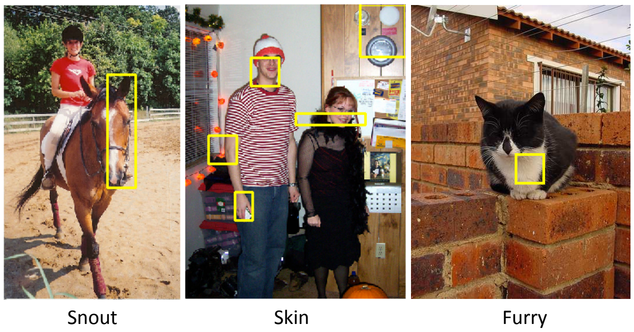


Figure 4. Predicting the bounding boxes for features corresponding to the attributes "snout", "skin", and "furry".

5.3. Choice of hyperparameters

In our experiments, we used the hyperparameter values $\alpha = 1$, $\beta = 5$ for the BBP-RBM. We observed that the exact choice of these hyperparameter had very little effect on the performance. The parameters $\mathbf{W}$, $\mathbf{b}$, and $\mathbf{c}$ were initialized by drawing from a zero-mean isotropic Gaussian with standard deviation 0.001. We also added ℓ_2 regularization for the elements of $\mathbf{W}$, and used the regularization hyperparameter 0.001 for the first layer and 0.01 for the second and third layers. We used a target sparsity of 0.2 for the sparse RBM. These hyperparameters were determined by cross validation.

6. Conclusion

In this work, we proposed the BBP-RBM as a new method to learn mid-level feature representations. The BBP-RBM is based on a factor graph representation that combines the properties of the RBM and the Beta-Bernoulli process. Our method can induce category-dependent sharing of learned features, which can be helpful in improving the generalization performance. We evaluated our model in object recognition experiments, and showed superior performance compared to recent state-of-the-art results, even though our model does not use any attribute labels. We also performed qualitative analysis on the semantic content of the learned features. Our results suggest that the learned mid-level features can capture distinct semantic concepts, and we believe that our method holds promise in advancing attribute-based recognition methods.

Acknowledgements

We acknowledge the support of the NSF Grant CPS-0931474 and a Google Faculty Research Award.

References

[1] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003.

[3] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. In *ICCV*, 2007.

[4] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[5] B. Chen, G. Polatkan, G. Sapiro, D. B. Dunson, and L. Carin. The hierarchical Beta process for convolutional factor analysis and deep learning. In *ICML*, 2011.

[6] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[7] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[9] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007.

[10] T. L. Griffiths and Z. Ghahramani. The indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224, 2011.

[11] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[12] G. E. Hinton and S. Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006, 2006.

[13] S. J. Hwang, F. Sha, and K. Grauman. Sharing features between objects and their attributes. In *CVPR*, 2011.

[14] A. Krizhevsky. Convolutional deep belief networks on cifar-10. Technical report, 2010.

[15] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012.

[16] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[17] H. Larochelle and Y. Bengio. Classification using discriminative restricted Boltzmann machines. In *ICML*, 2008.

[18] H. Lee, C. Ekanadham, and A. Y. Ng. Sparse deep belief net model for visual area V2. 2008.

[19] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, 54(10):95–103, 2011.

[20] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding:classification, annotation and segmentation in an automatic framework. In *CVPR*, 2009.

[21] L.-J. Li, C. Wang, Y. Lim, D. M. Blei, and L. Fei-Fei. Building and using a semantivisual image hierarchy. *CVPR*, 2010.

[22] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *CVPR*, 2011.

[23] J. W. Paisley and L. Carin. Nonparametric factor analysis with Beta process priors. In *ICML*, 2009.

[24] R. Salakhutdinov and G. Hinton. Replicated softmax: an undirected topic model. In *NIPS*, 2010.

[25] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1:194–281, 1986.

[26] K. Sohn, D. Y. Jung, H. Lee, and A. Hero III. Efficient learning of sparse, distributed, convolutional feature representations for object recognition. In *CVPR*, 2011.

[27] Y. Su and F. Jurie. Improving image classification using semantic attributes. *International Journal of Computer Vision*, 100(1):59–77, 2012.

[28] R. Thibaux and M. I. Jordan. Hierarchical Beta processes and the indian buffet process. In *AISTATS*, 2007.

[29] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. In *ECCV*, 2010.

[30] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin. Non-parametric bayesian dictionary learning for sparse image representations. In *NIPS*. 2009.