Jointly Learning and Selecting Features via Conditional Point-wise Mixture RBMs

Kihyuk Sohn **Guanyu Zhou** **Honglak Lee**
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109, USA

Abstract

Feature selection is an important technique for finding relevant features from high-dimensional data. However, the performance of feature selection methods is often limited by the *raw* feature representation. On the other hand, unsupervised feature learning has recently emerged as a promising tool for extracting useful features from data. Although supervised information can be exploited in the process of supervised fine-tuning (preceded by unsupervised pre-training), the training becomes challenging when the unlabeled data contain significant amounts of irrelevant information. To address these issues, we propose a new generative model, the *conditional point-wise mixture restricted Boltzmann machine*, which attempts to perform feature grouping while learning the features. Our model represents each input coordinate as a mixture model when conditioned on the hidden units, where each group of hidden units can generate the corresponding mixture component. Furthermore, we present an extension of our method that combines bottom-up feature learning and top-down feature selection in a unified way, which can effectively handle irrelevant input patterns by *focusing on relevant signals* and thus learn more informative features. Our experiments show that our model is effective in learning separate groups of hidden units (e.g., that correspond to informative signals vs. irrelevant patterns) from complex, noisy data.

1 Introduction

Over the years, feature selection [28, 8, 26, 12] has been an important tool for finding relevant features from high-dimensional data. However, feature selection methods typically assume that there exists a fixed set of *raw features* (e.g., input coordinates) that *readily* provides relevant information about the tasks of interest. However, this assumption does not hold when there does not exist good domain knowledge or hand-crafted features.

Recently, representation learning algorithms (e.g., [11, 3, 19, 14]; also see [2] for a survey) have emerged as promising tools for learning useful feature representations from unlabeled and labeled data. The strength of these methods is that they do not require much domain-specific knowledge, and thus can be easily adapted to other domains. Among these methods, restricted Boltzmann machines (RBMs) [22] have shown great promise in learning features from complex data. Despite the promise, the RBM is an unsupervised learning algorithm and therefore lacks ability to distinguish relevant information from noisy data. Although supervised information can be exploited in the process of supervised fine-tuning (preceded by unsupervised pre-training), training becomes challenging when the unlabeled data contains significant irrelevant information.

To address this issue, the paper presents the *conditional point-wise mixture restricted Boltzmann machine (pmRBM)* which can *learn* features and *group* the learned features at the same time. In other words, our model is able to separate between different groups of hidden units (where each group of hidden units model semantically distinct patterns). Specifically, the proposed model assumes that each visible unit (i.e., each coordinate in the input) is represented as a mixture model when conditioned over the hidden units, and it learns groups of hidden units that correspond to each mixture component. Our method can be naturally combined with a supervised setting, and we present a generative model that can *jointly perform feature learning and feature selection*.

We apply our method in two scenarios: recognizing foreground digits from noisy background and classifying objects from natural images. For the first scenario, our method shows strong performance in learning features and separating the task-relevant features from task-irrelevant features, achieving state-of-the-art classification results. In the second scenario, our model can successfully distinguish the foreground objects and detect their bounding boxes in a weakly-supervised way (i.e., without supervision about bounding boxes), which improves object recognition performance.

2 Preliminaries

Since our model is built upon the RBM, we briefly review it in this section. The RBM is an undirected generative model that defines the distribution of visible units (i.e., input data) using binary hidden units. Assuming that input data is binary-valued, the joint distribution of an RBM is defined as follows:

$$\begin{aligned} P(\mathbf{v},\mathbf{h}) &= \frac{1}{Z}\exp(-E(\mathbf{v},\mathbf{h})), \\ E(\mathbf{v},\mathbf{h}) &= -\mathbf{c}^T\mathbf{v} - \mathbf{b}^T\mathbf{h} - \mathbf{v}^T W\mathbf{h}, \end{aligned}$$

where $\mathbf{v} \in \{0,1\}^D$ are visible units, $\mathbf{h} \in \{0,1\}^K$ are hidden units, Z is the normalization constant, $W \in \mathbb{R}^{D\times K}$ is the weight matrix, $\mathbf{b} \in \mathbb{R}^K$ is the hidden bias vector, and $\mathbf{c} \in \mathbb{R}^D$ is the visible bias vector. We can also write the energy function as follows:

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{D}\sum_{j=1}^{K} v_i W_{ij} h_j - \sum_{j=1}^{K} b_j h_j - \sum_{i=1}^{D} c_i v_i.$$

Since there are no connections between units in the same layer, visible units are conditionally independent given the hidden units (and vice versa). The conditional probabilities of individual v_i and h_j can be explicitly written as follows:

$$P(v_i = 1 \mid \mathbf{h}) = \sigma(\sum_{j=1}^{K} W_{ij} h_j + c_i),$$

$$P(h_j = 1 \mid \mathbf{v}) = \sigma(\sum_{i=1}^{D} W_{ij} v_i + b_j),$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$. For training, computing the gradient of the log-likelihood for RBMs is intractable. Instead, we can use the contrastive divergence approximation [10] through Gibbs sampling for optimizing the parameters in the RBMs.

3 Proposed Model

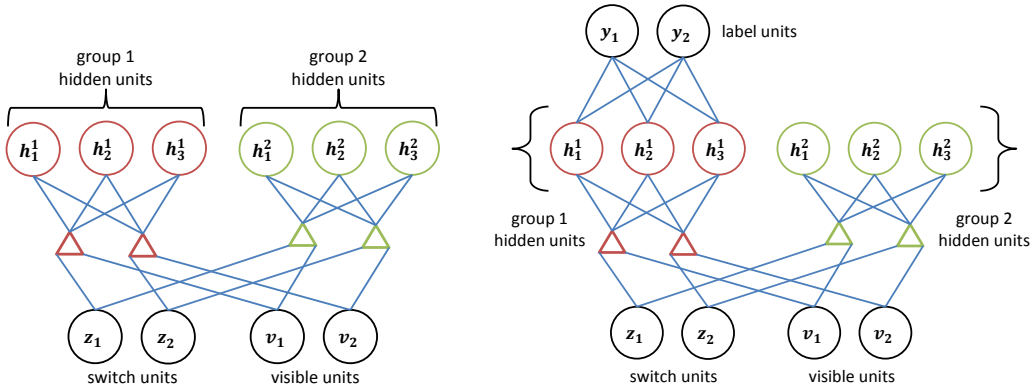
3.1 Conditional point-wise mixture restricted Boltzmann machine

In this paper, we consider highly complex data where each example can be decomposed into many semantically distinct patterns. In such cases, we assume that the data is generated from a mixture model, where each mixture component models some portion of the example using distributed representations. Specifically, our proposed model, the conditional point-wise mixture restricted Boltzmann machine (pmRBM), represents each visible unit as a mixture model conditioned over the hidden units.

In a generative modeling perspective, this can be interpreted as follows: (1) we first assume implicit prior on groups of binary hidden units, where each group of hidden units defines a distinct distribution over visible units. (2) After conditioning over the hidden units, we can sample the switching unit for each visible unit. (3) Finally, the switching unit determines which group of hidden units will generate the corresponding visible unit. A schematic diagram is shown in Figure 1(a), and we further specify this generative process using an undirected model.

To construct the conditional point-wise mixture RBM with C mixture components, we first introduce a switch unit $z_i \in \{1,\cdots,C\}$[1] that represents the mixture component assignment for each

[1]For notational convenience, we also use the boldfaced, vector representation of switch unit, $\mathbf{z}_i = [z_i^{(1)},\cdots,z_i^{(C)}] \in \{0,1\}^C$ where $\sum_{r=1}^{C} z_i^{(r)} = 1$.



(a) unsupervised pmRBM (b) supervised pmRBM

Figure 1: Graphical model representation of the (a) unsupervised pmRBM and (b) supervised pmRBM. In this example, two groups of hidden units form the mixture components, and the Bernoulli switch units z_i specify under which of the two components the visible unit v_i is modeled. For each i, when $z_i = 1$, v_i is generated from the hidden units in group 1 (shown in red); when $z_i = 2$, v_i is generated from the hidden units in group 2 (shown in green). See text for more details.

visible unit v_i, and impose an element-wise multiplicative interaction between the switch unit and the corresponding visible unit, as shown in Figure 1(a). Then, the energy function can be defined as follows:

$$E^{\text{Unsup}}(\mathbf{v}, \mathbf{z}, \mathbf{h}) = -\sum_{i=1}^{D}\sum_{r=1}^{C}\left(z_i^{(r)} v_i\right)\left(\sum_{j=1}^{K_r} W_{ij}^{(r)} h_j^{(r)} + c_i^{(r)}\right) - \sum_{r=1}^{C}\sum_{j=1}^{K_r} b_j^{(r)} h_j^{(r)} \quad (1)$$

$$\text{s.t.} \quad \sum_{r=1}^{C} z_i^{(r)} = 1,\ i = 1, \cdots, D,$$

where $\mathbf{v}$, $\mathbf{z}$ and $\mathbf{h}$ are the visible, switch and the hidden units, respectively. Here, the switch unit $\mathbf{z}_i$ is a stochastic variable specifying the component assignment that is distributed under a Bernoulli (when $C = 2$) or a Categorical (when $C \geq 2$) distributions given the hidden units. The model parameters $W_{ij}^{(r)}$, $b_j^{(r)}$, $c_j^{(r)}$ are the weights, hidden biases, and the visible biases corresponding to the r-th component, respectively.

The visible, hidden, and switch units are conditionally independent given the other two types of units. Therefore, the conditional probabilities can be computed as follows:

$$P\left(h_j^{(r)} = 1 \mid \mathbf{z}, \mathbf{v}\right) = \sigma\left(\sum_{i=1}^{D}\left(z_i^{(r)} v_i\right) W_{ij}^{(r)} + b_j^{(r)}\right), \quad (2)$$

$$P\left(v_i = 1 \mid \mathbf{z}, \mathbf{h}\right) = \sigma\left(\sum_{r=1}^{C} z_i^{(r)}\left(\sum_{j=1}^{K_r} W_{ij}^{(r)} h_j^{(r)} + c_i^{(r)}\right)\right), \quad (3)$$

$$P\left(z_i^{(r)} = 1 \mid \mathbf{v}, \mathbf{h}\right) = \frac{\exp\left(v_i\left(\sum_{j=1}^{K_r} W_{ij}^{(r)} h_j^{(r)} + c_i^{r}\right)\right)}{\sum_{s=1}^{C}\exp\left(v_i\left(\sum_{j=1}^{K_s} W_{ij}^{(s)} h_j^{(s)} + c_i^{(s)}\right)\right)}. \quad (4)$$

One important aspect is that, while inferring the hidden units, our model re-weighs each input visible unit according to the corresponding component weights provided by z_i (Equation (2)). In other words, the point-wise multiplicative interaction between the switch units and the visible units allows the hidden units in each component to *focus* on a specific portion of the example, and therefore the hidden units can be robust to irrelevant patterns in the data. Moreover, while inferring the switch units, the top-down signal from the hidden units encourages assigning the same mixture component to semantically correlated input coordinates, and this makes easy to prune out the irrelevant raw features more accurately for each example. We will further discuss this aspect in Section 3.2.

It is worth noting that when we tie all the switch units (i.e., $z_i = z, \forall i$), the pmRBM becomes equivalent to the implicit mixture of restricted Boltzmann machine (imRBM) [18]. Therefore, our model can be viewed as an extension of the imRBM, but it has several advantages over the imRBM, as will be discussed in detail in Section 4.

We train the pmRBM with stochastic gradient descent based on contrastive divergence. Due to the three-way interaction, however, exact inference is intractable. Instead, we use alternate block Gibbs sampling (i.e., iteratively sampling one type of variables given the other two based on Equations (2),(3), and (4)).

In principle, we can train pmRBM in a fully unsupervised way. However, we can learn a better model when we provide a good initialization of the weight matrices for different mixture components. For example, we can initialize the pmRBM with two components with regular RBM parameters; specifically, we can divide the RBM features into two groups by sorting them with the scores of simple feature selection algorithms, such as t-test [1].

3.2 Generative feature selection via supervised pmRBM

Although the pmRBM can learn distinct features for different mixture components, a generative training is done in unsupervised way and therefore it may not necessarily learn discriminative features. In this section, we propose a *supervised pmRBM*, which explicitly assigns which of C components to learn task-relevant features by connecting the hidden nodes of specific mixture components to the label nodes. The graphical model representation of supervised pmRBM is shown in Figure 1(b). As we discuss later, by transferring the supervision information to the raw features through the task-relevant hidden units, the supervised pmRBM can perform generative feature selection both at the high-level (i.e., using only a subset of hidden units for classification) and the low-level features (e.g., dynamically turning off the influence of the task-irrelevant visible units) in a unified way.

In this section, we present supervised pmRBM with two mixture components only, where the first component contains task-relevant hidden units. The energy function can be defined as follows:

$$\begin{aligned} E^{\text{Sup}}(\mathbf{v}, \mathbf{z}, \mathbf{h}, \mathbf{y}) &= E^{\text{Unsup}}(\mathbf{v}, \mathbf{z}, \mathbf{h}) - \sum_{l=1}^{L} h_j^{(1)} U_{lj} y_l - \sum_{l=1}^{L} d_l y_l \qquad (5) \\ \text{s.t.} \quad & z_i^{(1)} + z_i^{(2)} = 1, \ i = 1, \cdots, D, \end{aligned}$$

where $\mathbf{y} \in \{0, 1\}^L$ is a label vector in the 1-of-L representation, U_{lj} is a weight connection between the hidden units in task-relevant component and the labels, and d_l is a label bias. The conditional probabilities can be computed as follows:

$$P\left(h_j^{(1)} = 1 \mid \mathbf{z}, \mathbf{v}, \mathbf{y}\right) = \sigma\left(\sum_{i=1}^{D}\left(z_i^{(1)} v_i\right) W_{ij}^{(1)} + b_j^{(1)} + \sum_{l=1}^{L} y_l U_{lj}\right), \qquad (6)$$

$$P\left(h_j^{(2)} = 1 \mid \mathbf{z}, \mathbf{v}\right) = \sigma\left(\sum_{i=1}^{D}\left(z_i^{(2)} v_i\right) W_{ij}^{(2)} + b_j^{(2)}\right), \qquad (7)$$

$$P\left(z_i^{(1)} = 1 \mid \mathbf{v}, \mathbf{h}\right) = \frac{\exp\left(v_i\left(\sum_{j=1}^{K_1} W_{ij}^{(1)} h_j^{(1)} + c_i^{(1)}\right)\right)}{\sum_{s=1}^{2} \exp\left(v_i\left(\sum_{j=1}^{K_s} W_{ij}^{(s)} h_j^{(s)} + c_i^{(s)}\right)\right)}, \qquad (8)$$

$$P\left(y_l = 1 \mid \mathbf{h}^{(1)}\right) = \frac{\exp\left(\sum_{j=1}^{K_1} U_{lj} h_j^{(1)} + d_l\right)}{\sum_{s=1}^{L} \exp\left(\sum_{j=1}^{K_1} U_{sj} h_j^{(1)} + d_s\right)}. \qquad (9)$$

The conditional probability for the visible units is the same as Equations (3). As we can see in Equation (6), the label information is used to infer the hidden units in the first component, and this encourages the first component switch units $z_i^{(1)}$ to activate at the task-relevant input coordinates.

We can train the supervised pmRBM in generative criteria whose objective is to maximize the joint log-likelihood of the visible units and the labels [13]. Similar to the unsupervised pmRBM, we can do inference using alternate block Gibbs sampling, following Equations (6), (7) and (8).

3.3 Variations of the model

Deep networks The proposed model can be used as a building block to build a deep network (DN). Specifically, we can use pmRBM to model the first layer representation and stack layers of neural networks or deep belief networks on top of the hidden units belonging to the *task-relevant* component, which can be determined by measuring the validation performance. As we show in Section 5, the pmRBM can distinguish the groups of hidden units, which provides an implicit feature selection mechanism (i.e., selecting hidden units) that improves the performance when the input data inherently contains significant amounts of irrelevant patterns.

Convolutional pmRBM Convolutional models can be useful in representing spatially or temporally correlated data (e.g., images, speech, and video). Similar to RBMs, our patch-based model can be extended to a convolutional setting [17], where the filter weights are shared over different locations in a large image. We will present experimental results showing that the convolutional pmRBM can be used in segmenting the foreground object without bounding box information. Furthermore, we will show the improved object recognition performance by accurately detecting the bounding box of the foreground object with the convolutional pmRBM.

4 Related Work

As mentioned in Section 3, the pmRBM can be viewed as an extension of the imRBM by allowing *per-visible-unit switching*. Although the two models appear similar, the pmRBM has several important advantages over the imRBM. First, the pmRBM allows a specific subset of hidden units to focus on the relevant patterns in the data (via switch units), which can be useful when the data inherently contains significant amounts of irrelevant patterns in each sample. However, the imRBM represents the entire data with the hidden units in a single mixture component and therefore it cannot distinguish between relevant and irrelevant patterns. Second, per-visible-unit switching allows grouping of the hidden units, which can be useful in unsupervised or supervised feature selection, as discussed in Section 3.2. Third, our model is flexible and provides a new perspective that could lead to interesting extensions. For example, when specific visible units are correlated, it is plausible to apply the *local sharing* of switch units, which is left as a future work.

In the context of learning and selecting features simultaneously, our work is also related to the discriminative RBM (discRBM) [13]. However, the pmRBM can be more robust to the input data that contains significant amount of noise since the switch units can block the irrelevant input coordinates dynamically for each input data, whereas, due to the static switch units, the discRBM accumulates the energy from the noisy coordinates unless the weight connection between the hidden unit and the visible layer is very sparse. Moreover, as we have discussed in Section 3.2, we can do generative training of pmRBM with label nodes on top of task-relevant group of hidden units to incorporate both bottom-up and top-down feature selection mechanisms (e.g., supervised pmRBM). We empirically show that the pmRBM results in much better classification performance than the discRBM.

Our model is also related to several recently proposed models. For example, the robust Boltzmann machine (RoBM) [24] shares similar motivation to our work. However, there are several differences. First, the RoBM models each background coordinate with unimodal Gaussian distribution, whereas the pmRBM models the background coordinates with multimodal distribution with a group of hidden units. Furthermore, the pmRBM can directly learn from the noisy data with proper initialization using labels, which is arguably much easier information to obtain than the "clean" data that is required to pre-train the GRBM part of the RoBM.

The (unsupervised) pmRBM also looks similar to the masked restricted Boltzmann machines [16, 9] in terms of energy function. However, our main motivation is to perform joint feature selection both at the low- and high-level, where the low-level features (e.g., raw pixels) are dynamically selected during the switch unit inference with the top-down signal conditioned on the groups of hidden units, and the high-level features can be grouped in an unsupervised manner with the bottom-up signal given switch units. The difference becomes much clearer comparing with the supervised pmRBM in Section 3.2 since we made it possible to perform generative feature selection at the raw feature level as well using the top-down signal from the hidden units that are grouped according to their relevance to the task. Finally, we achieved state-of-the-art classification performance on several challenging datasets.

5 Experiments

5.1 Recognizing handwritten digits in the presence of irrelevant patterns

We evaluated the capability of our model on learning and separating task-relevant features from the task-irrelevant features. In this experiment, we first tested a single-layer pmRBM on MNIST variation datasets [14], *mnist-back-rand* and *mnist-back-image*, which consist of digits as the foregrounds and uniform random noise or natural images as the backgrounds, respectively. Furthermore, we tested on the rotated version of above datasets, such as *mnist-rot-back-rand*[2] and *mnist-rot-back-image*. We used the pmRBM with two components, each of which contains 500 hidden units.

In Figure 2, we visualize the filters and the activation of switch units for *mnist-back-image* dataset. It is clear that most of the filters corresponding to one component represent the patterns in the foreground (e.g., "pen-strokes", Figure 2(a)). In contrast, the group of filters that correspond to the background (Figure 2(b)) are indeed noisy due to the natural images in the background. Furthermore, the activations of the switch units (i.e., the posterior probabilities of the input pixel belonging to the *foreground component*, Figure 2(c)) show a good distinction between the pixels belonging to the digits (colored in white) and the background pixels (colored in gray). This suggests that our model has a good potential to distinguish and group the features into relevant and irrelevant patterns (e.g., foreground and background patterns).

For quantitative evaluation, we show test classification errors in Tables 1 and 2. In our experiments, we used support vector machine [5] as a classifier for single-layer models, and softmax classifier for the deep networks to perform fine-tuning. Since our model divides the hidden units into two groups, we only used the "task-relevant hidden unit" activations as the input for the classifier, which can be easily chosen by computing the validation performance for each component separately. Compared to the regular RBM, the single-layer pmRBM achieved significantly lower classification errors for all datasets. For comparison, we also evaluated performance of the imRBM [18][3] and the discRBM [13][4] after careful cross-validation. The classification errors for both models were much higher than those of pmRBM, and this suggests that our point-wise mixture hypothesis is effective in learning task-relevant features from complex data which contain highly irrelevant patterns.

To evaluate the advantage of joint feature learning and selection, we compared the results of our model to that of a two-step process in which we first learn features with an RBM and apply feature selection on the RBM features. We denote this as an "RBM-FS" model. As we can see in Table 2, the RBM-FS model shows reasonable improvement over the baseline RBM in most cases. However, our pmRBM model can improve the quality of task-relevant features through joint feature learning and feature selection, which results in significant additional improvement in classification performance.

Algorithm	RBM [25]	imRBM	discRBM	RBM-FS	pmRBM (Unsup)	pmRBM (Sup)
mnist-back-rand	9.80	9.94	9.64	10.18	6.29	**6.20**
mnist-back-image	16.15	15.96	15.35	13.84	13.29	**13.00**
mnist-rot-back-rand	51.05	52.16	47.75	49.87	45.36	**41.44**
mnist-rot-back-image	52.21	50.10	49.12	47.71	44.23	**44.10**

Table 1: Test classification errors of single-layer models on MNIST variation datasets. We refer "RBM-FS" to an RBM with feature selection. We denote supervised pmRBM as "pmRBM (Sup)".

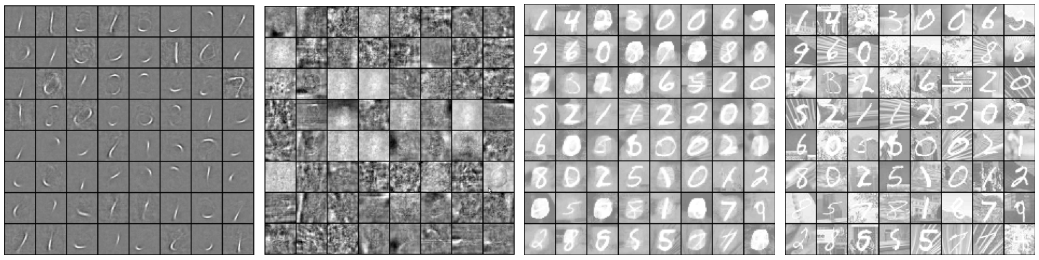
Algorithm	pmRBM	pmRBM + DN-1	DBN-3 [25]	CAE-2 [21]	CAE-H-2 [20]
mnist-back-rand	6.29	**5.05**	6.73	10.90	-
mnist-back-image	13.29	**12.30**	16.31	15.50	14.8
mnist-rot-back-rand	45.36	**29.67**	-	-	-
mnist-rot-back-image	44.23	**35.02**	47.39	45.23	-

Table 2: Test classification errors of deep networks on MNIST variation datasets.

[2]We generated *mnist-rot-back-rand* dataset in a similar way of generating *mnist-rot-back-image* and *mnist-back-rand* datasets. All the results for *mnist-rot-back-rand* dataset are produced by ourselves.

[3]For the imRBM, we report the results after cross-validating over the total number of hidden units, the number of mixture components, and other hyperparameters.

[4]We used "hybrid" (discriminative-generative) RBM whose objective is defined as a combination of discriminative objective (classification loss) and generative objective (log-likelihood), as defined in [13]. We cross validated the generative weight α from a range between 0.01 and 0.5.



(a) task-relevant filters (b) task-irrelevant filters (c) switch unit activation (d) original images

Figure 2: (a, b) Visualization of filters corresponding to two components learned from the pmRBM, (c) visualization of the activation of switch units, and (d) corresponding original images on *mnist-back-image* dataset. Specifically, (a) represents the group of hidden units that activates for the foreground digits (task-relevant), and (b) represents the group of hidden units that activates for the background images (task-irrelvant). See text for details.

Finally, we constructed a deep network by stacking a single-layer neural network with 1000 hidden units on the task-relevant components of a first-layer pmRBM. Table 2 shows that the deep network variant of the pmRBM (referred to as "pmRBM+DN-1") outperformed the DBN-3 or stacked contractive autoencoders with a large margin. In particular, the result of DBN-3 on *mnist-back-image* implies that stacking up unsupervised feature learning modules (e.g., RBM) does not necessarily improve the performance even after fine-tuning, potentially due to the presence of highly-irrelevant patterns in the data. In contrast, our model can selectively propagate task-relevant information to the higher layers, which explains its superior performance over other baseline models. These results suggest that the pmRBM can be a useful building block for deep networks, especially in dealing with highly irrelevant patterns.

5.2 Unsupervised object segmentation, with application to object recognition

Unsupervised object segmentation In this section, we evaluated the capability of our model on learning relevant features (e.g., foreground patterns) from the large images. To extract features from images with higher resolution, we extend our proposed model to a convolutional pmRBM and applied it to build a deep convolutional network. Specifically, we formed two-layer convolutional network that is similar to the convolutional deep belief network (CDBN) proposed in [17]. While the original CDBN is built using the convolutional RBM (with probabilistic max-pooling) as a building block in all layers, we departed from this model by applying convolutional pmRBM only to the second layer (on top of the first layer convolutional RBM) rather than stacking up the convolutional pmRBM modules from the first layer. This strategy makes sense given that the convolutional RBM learns generic oriented edge filters in the first layer.

To provide an appropriate initialization for the convolutional pmRBM, we first trained a set of second-layer CRBMs composed of a small number of hidden units (e.g., 30) for each object category[5] from Caltech 101 dataset [6], and performed a top-down supervised feature selection from the union of category-specific CRBM features from all object classes. Once initialized, we used the training images from all object categories to train convolutional pmRBM.

We visualize the second-layer task-relevant and task-irrelevant features learned from the two-layer convolutional pmRBM network in Figure 3. As we can see, our model can learn object parts (e.g., face parts, wheels, etc.) for the foreground component and learn more generic patterns like contours or corners for the background component. In Figure 4 (top row), we also visualize the activation map of switch units, which shows that the switch units can select the most informative parts of an object. Interestingly, although our model is not designed for unsupervised image segmentation, using the activation of switch units, we can still segment out the object region from the background image reasonably well.

Object recognition Motivated by the convolutional pmRBM's ability in distinguishing the foreground object from the background image, we propose the novel object recognition pipeline that we first detect the bounding box for the object in each image using the two-layer convolutional pmRBM

[5] We trained CRBM on each class to learn more diverse patterns that can capture all object classes.

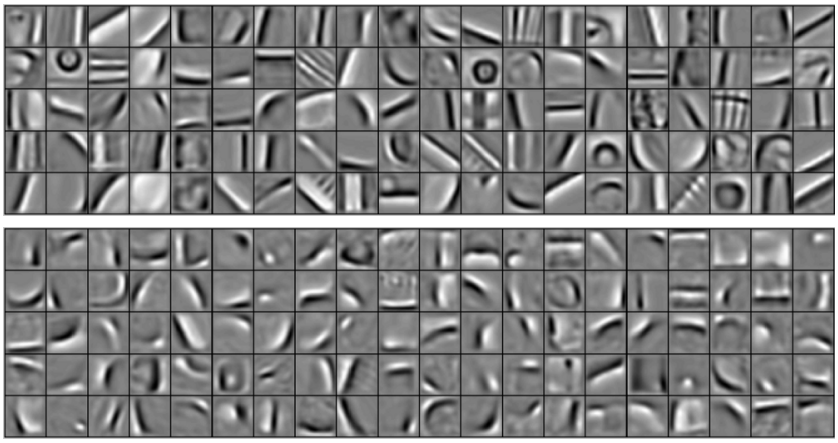


Figure 3: Visualization of second-layer features learned from convolutional pmRBM network on Caltech 101 dataset. Specifically, we show visualizations of all features in the task-relevant component (top) and all features in the task-irrelevant component (bottom).

network as described above, and perform the classification on the Caltech 101 dataset augmented with the detected bounding boxes. To detect the bounding box, we used convolutional pmRBM with two groups that are composed of 100 hidden units for each.[6] For classification, we followed the classification pipeline used in [23], which uses the gaussian RBMs on the SIFT features.

Before the classification tasks, we evaluated the bounding box detection accuracy, where the overlap was measured as the ratio between the area of the intersection and the union of predicted and the ground truth bounding boxes, and it is declared as a correct detection if the overlap is greater than 0.5. We obtained a mean overlap of 70.2% and a detection accuracy of 88.3%. We visualize some examples of detected bounding boxes (red) and ground truth bounding boxes (green) at the bottom of Figure 4. As we can see, the convolutional pmRBM can be used to localize the bounding box around an object. Finally, we report the classification accuracy in Table 3. Using 4096 hidden units, we were able to achieve improved classification accuracies upon the baseline model [23], such as 70.2% and 76.8% with RBM and 72.4% and 78.9% with CRBM, using 15 and 30 training images per class, respectively. As a baseline comparison, we also report the classification accuracy on the augmented dataset where we simply crop the center region uniformly across all the images with a fixed ratio. We used RBM with 4096 hidden units, and after cross-validating with different ratios, we obtain a worse accuracy of 75.8% using 30 training images per class. This suggests that our method is doing more than cropping the center region, but localizing and tightening the bounding box around the object well.

Training images	15	30
Lazebnik et al. [15]	56.4%	64.6%
Griffin et al. [7]	59.0%	67.6%
Yang et al. [27]	67.0%	73.2%
Boureau et al. [4]	-	75.7%
RBM [23]	68.6%	74.9%
Our method + RBM	70.2%	76.8%
CRBM [23]	71.3%	77.8%
Our method + CRBM	**72.4%**	**78.9%**

Table 3: Test classification accuracy on the Caltech-101 dataset.

6 Conclusion

In this paper, we proposed a conditional point-wise mixture restricted Boltzmann machine that can effectively learn useful feature representations from data containing highly irrelevant patterns. Our model can selectively learn distinct groups of features (e.g., foreground and background patterns). This property of our model naturally enables unsupervised bottom-up feature selection by dynamically activating switching variables in the input data. Furthermore, we proposed a generative model for joint top-down and bottom-up feature selection. Our experimental results suggest that: (1) the proposed method is effective in distinguishing different groups of features (e.g., task-relevant pat-

[6] Implementation detail: Specifically, for each image, we first estimate the posterior (activation) of the switching units (arranged in 2d). Then, we simply computed the row-wise or column-wise cumulative sum of the switch unit activations and estimated the bounding box by picking the range that contains $(5, 95)$ percentiles of the total activations in both row-wise and column-wise.

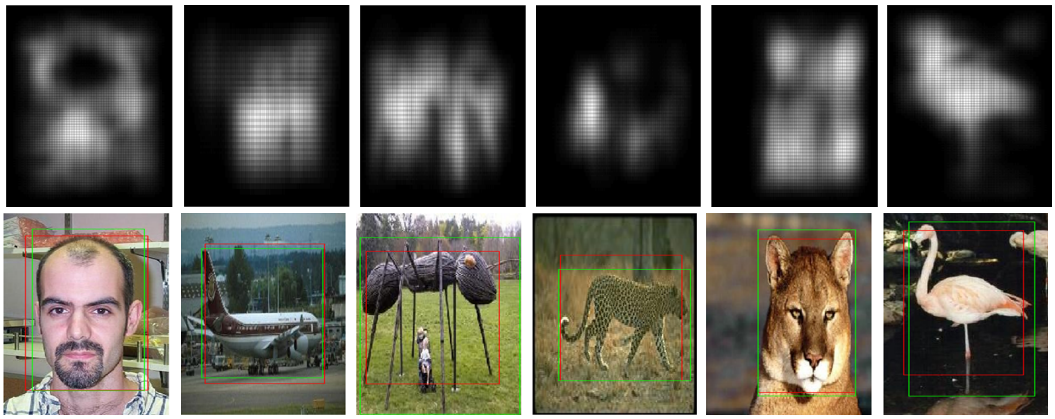


Figure 4: (Top) Visualization of activation map of switch units corresponding to each image below. (Bottom) images overlayed with the estimated bounding boxes (red) and ground truth bounding boxes (green).

terns and task-irrelevant patterns); (2) our model can *filter out* groups of hidden units that represent irrelevant patterns, which then enables itself to *focus on* a subset of raw input coordinates dynamically; (3) our model provides significant improvement in classification performance compared to other strong baseline models on the challenging benchmark datasets; and (4) our convolutional extension of the pmRBM is effective in detecting objects without any ground truth bounding box information, which leads to significantly improved object classification performance.

Acknowledgments

This work was supported in part by NSF IIS 1247414 and a Google Faculty research award. We also thank Chansoo Lee, Ali Fadlallah, Scott Reed, and Min-Yian Su for helpful comments.

References

[1] http://featureselection.asu.edu/documentation/ttest.htm. 4

[2] Y. Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, Jan. 2009. 1

[3] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *NIPS*, 2007. 1

[4] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *CVPR*, 2010. 8

[5] 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*, 2008. 6

[6] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative Model Based Vision*, 2004. 7

[7] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007. 8

[8] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR*, 3:1157–1182, 2003. 1

[9] N. Heess, N. Le Roux, and J. Winn. Weakly supervised learning of foreground-background segmentation using masked rbms. In *ICANN*, 2011. 5

[10] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002. 2

[11] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006. 1

[12] A. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE TPAMI*, 19(2):153–158, 1997. 1

[13] H. Larochelle and Y. Bengio. Classification using discriminative restricted Boltzmann machines. In *Proceedings of the International Conference on Machine Learning*, 2008. 4, 5, 6

[14] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *ICML*, 2007. 1, 6

[15] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 8

[16] N. Le Roux, N. Heess, J. Shotton, and J. Winn. Learning a generative model of images by factoring appearance and shape. *Neural computation*, 23(3):593–650, 2011. 5

[17] 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. 5, 7

[18] V. Nair and G. Hinton. Implicit mixtures of restricted boltzmann machines. *NIPS*, 2008. 4, 6

[19] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun. Efficient learning of sparse representations with an energy-based model. In *NIPS*, 2006. 1

[20] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin, and X. Glorot. Higher order contractive auto-encoder. In *ECML*, 2011. 6

[21] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *ICML*, 2011. 6

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

[23] K. Sohn, D. Y. Jung, H. Lee, and A. Hero III. Efficient learning of sparse, distributed, convolutional feature representations for object recognition. In *Proceedings of 13th International Conference on Computer Vision*, 2011. 8

[24] Y. Tang, R. Salakhutdinov, and G. E. Hinton. Robust boltzmann machines for recognition and denoising. In *CVPR*, 2012. 5

[25] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008. 6

[26] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svms. *NIPS*, pages 668–674, 2001. 1

[27] J. Yang, K. Yu, Y. Gong, and T. S. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801, 2009. 8

[28] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, pages 412–420, 1997. 1