# Classification of Sets using Restricted Boltzmann Machines

**Jérôme Louradour**
A2iA SA
40 bis, rue Fabert
75007 Paris, France

**Hugo Larochelle** *
Département d'informatique
Université de Sherbrooke
Sherbrooke, Canada

## Abstract

We consider the problem of classification when inputs correspond to sets of vectors with the same size. This setting occurs in many problems such as the classification of pieces of mail containing several pages, of web sites with several sections or of images that have been pre-segmented into smaller regions. We propose generalizations of the restricted Boltzmann machine (RBM) that are appropriate in this context and explore how to incorporate different assumptions about the relationship between the input sets and the target class within the RBM. In experiments on standard multiple-instance learning datasets, we demonstrate the competitiveness of approaches based on RBMs and apply the proposed variants to the problem of incoming mail classification.

## 1 Introduction

The vast majority of machine learning algorithms are developed in the context where each input can be assumed to take the form of a fixed-size vector $\mathbf{x}$. In some applications however, inputs cannot easily be converted into this form. In this paper, we consider one such setting where the input consists in an unordered and variable-length set of vectors $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(|\mathbf{X}|)}\}$ where each vector is of the same size. For instance, $\mathbf{X}$ could be the set of text documents $\mathbf{x}^{(s)}$ found in some incoming piece of mail, where each document is represented as a bag of words. In this particular example, a simple approach to converting the set $\mathbf{X}$ into a single vector $\mathbf{x}$ would consist in computing the global bag of word representation of all documents in $\mathbf{X}$, as if all documents had been concatenated into a single one. This would however correspond to throwing away all the information about the structure of the incoming mail, which could be useful to solve the task at hand. This problem setting is not specific to text data either: $\mathbf{X}$ could correspond to a collection of images or to a single image that has been pre-segmented, and some recognition tasks in computer vision have previously been formulated in terms of classification of sets (Kondor & Jebara, 2003; Wallraven et al., 2003). Another example is text-independent speaker recognition (Reynolds, 1995), where inputs are sequences of acoustic vectors but for which the order is not relevant: relevant short-term dynamics are taken into account in the vector features themselves (e.g. spectral coefficients and their derivatives) and long-term dynamics are not useful for classification (the succession of these features is informative of the speech content, not the speaker identity).

A popular approach to classifying sets has been that of multiple-instance learning (MIL). In MIL, binary classification of sets of vectors is performed by assuming that a set belongs to the positive class if at least one element of the set belongs to that positive class. Otherwise the set belongs to the negative class (i.e. all elements of the set are from the negative class). This problem was originally motivated in the context of drug activity prediction (Dietterich et al., 1997), where a drug molecule can take several shapes but only some of them might allow the molecule to bind with some given protein associated with a disease. A drug molecule can then be represented as the set of its potential shapes and this set will have a positive label only if at least one of its shapes allow binding.

The MIL approach makes the implicit assumption that the presence of just a single positive example is sufficient to recognize the whole set as positive. However, this assumption is not always appropriate. For instance, each vector could only provide *partial* class information, such that the observation of only a single informative vector is not enough to label the whole set.

* This work was done while Hugo was at University of Toronto

In this paper, we describe extensions of the restricted Boltzmann machine that perform multiclass classification of sets and do not assume that sufficient discriminative information is present in a single element of the set. By learning a latent representation of its input, these extensions can deal with cases where only partial evidence of class membership is present in only a few set vectors. We report competitive results on some common MIL datasets and present an application of these models to a mail classification problem.

## 2 Classification with Restricted Boltzmann Machines

In this work, we build on a specific restricted Boltzmann machine (RBM) that can be used to perform classification (Larochelle & Bengio, 2008; Tieleman, 2008). We will refer to this RBM as a classification RBM (ClassRBM).

The ClassRBM is an energy-based probabilistic model where a layer $\mathbf{h}$ of $H$ binary hidden units are used to model the joint distribution of a vector of $D$ inputs $\mathbf{x}$ and a target vector $\mathbf{y}$ of size $C$. The target $\mathbf{y}$ corresponds to a class label and takes the "one out of $C$" representation, meaning that if $\mathbf{x}$ belongs to class $c$, then $\mathbf{y} = \mathbf{e}_c$ where $\mathbf{e}_c$ is a vector with all values set to 0 except at position $c$, which is set to 1. For simplicity, we will also assume that $\mathbf{x}$ is a binary vector, though generalizations to other types of vectors are possible (Welling et al., 2005).

Using the energy function

$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}) = -\mathbf{d}^\top \mathbf{y} - \mathbf{c}^\top \mathbf{h} - \mathbf{b}^\top \mathbf{x} - \mathbf{h}^\top \mathbf{W}\mathbf{x} - \mathbf{h}^\top \mathbf{U}\mathbf{y},$$

the probability for some configuration of $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{h}$ is defined as

$$p(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \exp(-E(\mathbf{x}, \mathbf{y}, \mathbf{h}))/Z \tag{1}$$

where $Z$ is a normalizing constant that ensures $p(\mathbf{x}, \mathbf{y}, \mathbf{h})$ defines a valid distribution. Figure 1 shows an illustration of a ClassRBM.

Though $Z$ (and hence $p(\mathbf{x}, \mathbf{y}, \mathbf{h})$) is usually intractable to compute, the following conditional distributions of the model are themselves tractable:

$$\begin{aligned}
p(\mathbf{h}|\mathbf{x}, \mathbf{y}) &= \prod_{j=1}^{H} p(h_j|\mathbf{x}, \mathbf{y}) \\
p(h_j = 1|\mathbf{x}, \mathbf{y}) &= \mathrm{sigm}(c_j + \mathbf{W}_{j\cdot}\mathbf{x} + \mathbf{U}_{j\cdot}\mathbf{y}) \\
p(\mathbf{x}, \mathbf{y}|\mathbf{h}) &= p(\mathbf{y}|\mathbf{h}) \prod_{i=1}^{D} p(x_i|\mathbf{h}) \\
p(x_i = 1|\mathbf{h}) &= \mathrm{sigm}(b_i + \mathbf{h}^\top \mathbf{W}_{\cdot i}) \\
p(\mathbf{y} = \mathbf{e}_c|\mathbf{h}) &= \frac{\exp(\mathbf{d}^\top \mathbf{e}_c + \mathbf{h}^\top \mathbf{U}\mathbf{e}_c)}{\sum_{c'=1}^{C} \exp(\mathbf{d}^\top \mathbf{e}_{c'} + \mathbf{h}^\top \mathbf{U}\mathbf{e}_{c'})}
\end{aligned}$$

where $\mathrm{sigm}(v) = 1/(1 + \exp(-v))$ and we use the notation $\mathbf{A}_{j\cdot}$ to refer to the $j^{\text{th}}$ row of matrix $\mathbf{A}$ and $\mathbf{A}_{\cdot i}$ for it's $i^{\text{th}}$ column.

Given (1), it can also be shown that the posterior class probability distribution given some input $\mathbf{x}$ has the closed form

$$\begin{aligned}
p(\mathbf{y} = \mathbf{e}_c|\mathbf{x}) &= \sum_{\mathbf{h}} p(\mathbf{y} = \mathbf{e}_c, \mathbf{h}|\mathbf{x}) \\
&= \frac{\exp(-F(\mathbf{x}, \mathbf{e}_c))}{\sum_{c'=1\ldots C} \exp(-F(\mathbf{x}, \mathbf{e}_{c'}))}
\end{aligned} \tag{2}$$

where $F(\mathbf{x}, \mathbf{y})$ is referred to as the free-energy

$$F(\mathbf{x}, \mathbf{y}) = -\mathbf{d}^\top \mathbf{y} - \sum_{j=1}^{H} \mathrm{softplus}\left(c_j + \mathbf{W}_{j,\cdot}\mathbf{x} + \mathbf{U}_{j,\cdot}\mathbf{y}\right) \tag{3}$$

with $\mathrm{softplus}(v) = \log(1 + \exp(v))$.

In order to train the ClassRBM, different strategies can be followed. A first option is to train it discriminatively, by minimizing the average negative conditional log-likelihood $-\log p(\mathbf{y}_t|\mathbf{x}_t)$ of the parameters for the available training data $\{\mathbf{x}_t, \mathbf{y}_t\}$. This can be achieved by simple stochastic gradient descent.

A second option is to train the ClassRBM generatively, by minimizing the negative joint log-likelihood $-\log p(\mathbf{x}_t, \mathbf{y}_t)$. Unfortunately, the necessary gradients cannot be computed exactly. The Contrastive Divergence (CD) algorithm (Hinton et al., 2006) however provides a useful approximation

$$\begin{aligned}
\frac{\partial - \log p(\mathbf{x}_t, \mathbf{y}_t)}{\partial \theta} \approx\ & \mathrm{E}_{\mathbf{h}|\mathbf{x}_t, \mathbf{y}_t}\left[\frac{\partial E(\mathbf{x}_t, \mathbf{y}_t, \mathbf{h})}{\partial \theta}\right] \\
& - \mathrm{E}_{\mathbf{h}|\widetilde{\mathbf{x}}_t, \widetilde{\mathbf{y}}_t}\left[\frac{\partial E(\widetilde{\mathbf{x}}_t, \widetilde{\mathbf{y}}_t, \mathbf{h})}{\partial \theta}\right]
\end{aligned} \tag{4}$$

where $\theta$ is any parameter of the ClassRBM and where $\widetilde{\mathbf{x}}_t$ and $\widetilde{\mathbf{y}}_t$ is the result of a one-step Gibbs sampling chain, initialized at the training example $\mathbf{x}_t$ and $\mathbf{y}_t$. Noting $\widehat{\mathbf{h}}_t = \mathrm{sigm}(\mathbf{c} + \mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{y}_t)$ and $\widetilde{\mathbf{h}}_t = \mathrm{sigm}(\mathbf{c} + \mathbf{W}\widetilde{\mathbf{x}}_t + \mathbf{U}\widetilde{\mathbf{y}}_t)$, we get the following stochastic gradient updates from CD:

$$\begin{array}{lcllc}
\mathbf{b} & \leftarrow & \mathbf{b} & + & \lambda\,(\mathbf{x}_t - \widetilde{\mathbf{x}}_t) \\
\mathbf{c} & \leftarrow & \mathbf{c} & + & \lambda\,(\widehat{\mathbf{h}}_t - \widetilde{\mathbf{h}}_t) \\
\mathbf{d} & \leftarrow & \mathbf{d} & + & \lambda\,(\mathbf{y}_t - \widetilde{\mathbf{y}}_t) \\
\mathbf{W} & \leftarrow & \mathbf{W} & + & \lambda\,(\widehat{\mathbf{h}}_t \mathbf{x}_t^\top - \widetilde{\mathbf{h}}_t \widetilde{\mathbf{x}}_t^\top) \\
\mathbf{U} & \leftarrow & \mathbf{U} & + & \lambda\,(\widehat{\mathbf{h}}_t \mathbf{y}_t^\top - \widetilde{\mathbf{h}}_t \widetilde{\mathbf{y}}_t^\top)
\end{array}$$

where $\lambda$ is the stochastic gradient learning rate used for generative training.

As argued by Larochelle and Bengio (2008), in some situations, neither discriminative nor generative learning alone are optimal and better performance can be achieved by using a linear combination of both objectives. This is referred to as hybrid generative/discriminative learning and corresponds to performing both the discriminative and generative parameter updates with separate learning rates.

## 3 Generalization of the ClassRBM to handle sets (ClassSetRBM)

Now, we wish to generalize the ClassRBM so that it can model the distribution of a set $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(|\mathbf{X}|)}\}$ with target vector $\mathbf{y}$. The simplest approach would be to connect each vector $\mathbf{x}^{(s)}$ to some global hidden layer $\mathbf{h}$ with the same connection matrix $\mathbf{W}$. However, this approach is not appropriate because, not only do we expect sets to have varying sizes, but also the number of vectors $\mathbf{x}^{(s)}$ in $\mathbf{X}$ that actually contain predictive information about $\mathbf{y}$ will also vary. By having just a single global hidden layer, the activity of hidden units would thus tend to oversaturate for sets of large size.

To address this issue, we propose two generalizations where each vector $\mathbf{x}^{(s)}$ of a set will be connected to its own "copy" of the hidden layer. The total number of hidden units will then depend on the size of the input[1]. All hidden units will be connected to its corresponding input vector by the same matrix $\mathbf{W}$.

Given this approach, there are still different design choices to be made as to how these hidden layers should interact with the target units $\mathbf{y}$. We present here two such choices, which correspond to different assumptions about the nature of the interaction between the input sets and the target.

### 3.1 ClassSetRBM with Mutually Exclusive Hidden Units (XOR)

If we believe that the vectors in the input set $\mathbf{X}$ all contain information of a distinct nature, then a hidden feature detected within one vector $\mathbf{x}^{(s)}$ would be expected to be absent in the other vectors of the set. In this case, the set structure would convey very useful information about how to perform classification.

To exploit such information, we could impose that the activity of hidden units be mutually exclusive across the vectors of the set. Noting $\mathbf{h}^{(s)}$ the hidden layer to which $\mathbf{x}^{(s)}$ is connected and $\mathbf{H} = \{\mathbf{h}^{(1)}, \ldots, \mathbf{h}^{(|\mathbf{X}|)}\}$ the set of hidden layer vectors, this would translate into requiring the constraint that

$$\textstyle\sum_{s=1}^{|\mathbf{X}|} h_j^{(s)} \in \{0, 1\} \qquad \forall j = 1 \ldots H \tag{5}$$

i.e. for all hidden unit position $j$, at most one hidden unit $h_j^{(s)}$ should be active across all vectors $\mathbf{x}^{(s)}$. With that, we define the energy function

$$E(\mathbf{X}, \mathbf{y}, \mathbf{H}) = -\mathbf{d}^\top \mathbf{y} - \sum_s \mathbf{b}^\top \mathbf{x}^{(s)} - \sum_s \mathbf{c}^\top \mathbf{h}^{(s)} - \sum_s \left( \mathbf{h}^{(s)\top} \mathbf{W} \mathbf{x}^{(s)} + \mathbf{h}^{(s)\top} \mathbf{U} \mathbf{y} \right)$$

where the target $\mathbf{y}$ is connected to all hidden layers through the same connection matrix $\mathbf{U}$. We will refer to this variant of the ClassRBM for sets as ClassSetRBM$^{\text{XOR}}$. See Figure 1 for an illustration of ClassSetRBM$^{\text{XOR}}$.

While more complicated than the ClassRBM for single vectors, it can be shown that ClassSetRBM$^{\text{XOR}}$ has simple conditional distributions as well. The hidden layers conditional distributions become

$$\begin{aligned} p(\mathbf{H}|\mathbf{X}, \mathbf{y}) &= \textstyle\prod_{j=1}^{H} p\big(\{h_j^{(s)}\}_{s=1}^{|\mathbf{X}|} | \mathbf{X}, \mathbf{y}\big) \\ p(h_j^{(s)} = 1|\mathbf{X}, \mathbf{y}) &= \frac{\exp\big(\text{act}_j(\mathbf{x}^{(s)}, \mathbf{y})\big)}{1 + \sum_{s'=1}^{|\mathbf{X}|} \exp\big(\text{act}_j(\mathbf{x}^{(s')}, \mathbf{y})\big)} \\ p(h_j^{(\cdot)} = 0|\mathbf{X}, \mathbf{y}) &= \frac{1}{1 + \sum_{s'=1}^{|\mathbf{X}|} \exp\big(\text{act}_j(\mathbf{x}^{(s')}, \mathbf{y})\big)} \end{aligned}$$

where $\text{act}_j(\mathbf{x}^{(s)}, \mathbf{y}) = c_j + \mathbf{W}_{j\cdot}\mathbf{x}^{(s)} + \mathbf{U}_{j\cdot}\mathbf{y}$, the statement $h_j^{(s)} = 1$ also implies $h_j^{(s')} = 0 \ \forall s' \neq s$ and the statement $h_j^{(\cdot)} = 0$ is a shorthand for $h_j^{(s)} = 0 \ \forall s = 1, \ldots, |\mathbf{X}|$. The input and target conditional distributions are

$$\begin{aligned} p(\mathbf{X}, \mathbf{y}|\mathbf{H}) &= p(\mathbf{y}|\mathbf{H}) \textstyle\prod_{s=1}^{|\mathbf{X}|} \prod_{i=1}^{D} p(x_i^{(s)}|\mathbf{h}^{(s)}) \\ p(x_i^{(s)} = 1|\mathbf{h}^{(s)}) &= \text{sigm}(b_i + \mathbf{h}^{(s)\top} \mathbf{W}_{\cdot i}) \\ p(\mathbf{y} = \mathbf{e}_c|\mathbf{H}) &= \frac{\exp(\mathbf{d}^\top \mathbf{e}_c + \sum_{s=1}^{|X|} \mathbf{h}^{(s)\top} \mathbf{U} \mathbf{e}_c)}{\sum_{c'=1}^{C} \exp(\mathbf{d}^\top \mathbf{e}_{c'} + \sum_{s=1}^{|X|} \mathbf{h}^{(s)\top} \mathbf{U} \mathbf{e}_{c'})}. \end{aligned}$$

These conditional distributions are simple enough that Gibbs sampling can be performed, by sampling each element of $\mathbf{H}$ given $\mathbf{X}$ and $\mathbf{y}$, and then sampling new values for the vectors in $\mathbf{X}$ and for $\mathbf{y}$.

The target posterior $p(\mathbf{y}|\mathbf{X})$ can also be computed efficiently. It can be shown (see supplementary material[2]) that it has the following form:

$$p(\mathbf{y} = \mathbf{e}_c|\mathbf{X}) = \frac{\exp(-F^{\text{XOR}}(\mathbf{X}, \mathbf{e}_c))}{\sum_{c'=1 \ldots C} \exp(-F^{\text{XOR}}(\mathbf{X}, \mathbf{e}_{c'}))} \tag{6}$$

where the free-energy $F^{\text{XOR}}(\mathbf{X}, \mathbf{y})$ is now

$$F^{\text{XOR}}(\mathbf{X}, \mathbf{y}) = -\mathbf{d}^\top \mathbf{y} - \sum_{j=1}^{H} \text{softplus}\left(\text{softmax}_j(\mathbf{X}) + \mathbf{U}_{j\cdot}\mathbf{y}\right)$$

with $\text{softmax}_j(\mathbf{X}) = c_j + \log(\sum_{s=1}^{|\mathbf{X}|} \exp(\mathbf{W}_{j\cdot}\mathbf{x}^{(s)}))$ can be seen as a soft version of the max function of $c_j + \mathbf{W}_{j\cdot}\mathbf{x}^{(s)}$ over the set of input vectors.

[1] The implication of this is that we always condition on the size of the input set.

[2] see http://arxiv.org/abs/1103.4896

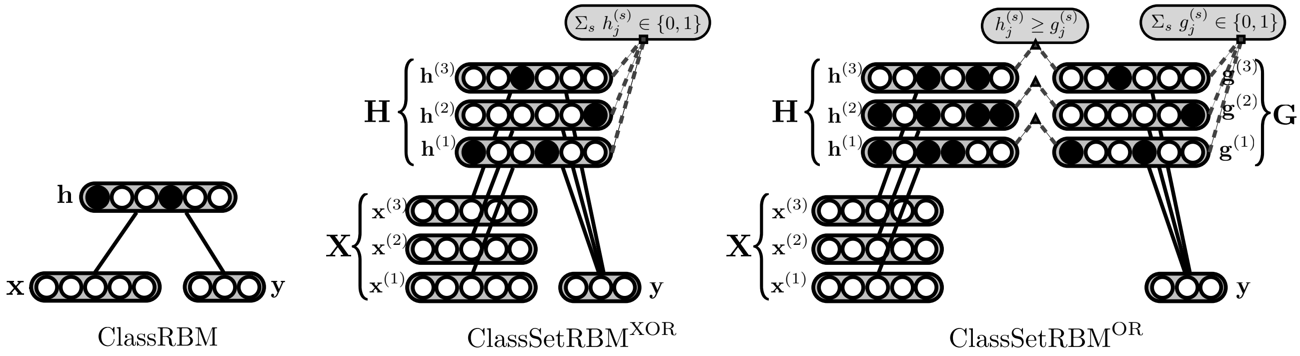


Figure 1: Illustration of the standard ClassRBM **(left)** and the two proposed variants for input sets: ClassSetRBM$^{\text{XOR}}$ **(middle)** and ClassSetRBM$^{\text{OR}}$ **(right)**. Dotted lines connect hidden layers whose activity is subject to joint constraints. An example of valid activations for the hidden layer units is given for each model (black means hidden unit is equal to 1).

As before, given a training set of pairs $\{\mathbf{X}_t, \mathbf{y}_t\}$, it is possible to train ClassSetRBM$^{\text{XOR}}$ discriminatively and generatively. The discriminative gradients are easily computed using the chain rule. CD approximations for the generative learning updates can also be obtained, since Gibbs sampling can be performed.

We note $\widetilde{\mathbf{X}}_t = \{\widetilde{\mathbf{x}}_t^{(1)}, \ldots, \widetilde{\mathbf{x}}_t^{(|\mathbf{X}_t|)}\}$ and $\widetilde{\mathbf{y}}_t$ as the result of one step of Gibbs sampling initialized at the training example pair $(\mathbf{X}_t, \mathbf{y}_t)$. Similarly to the standard ClassRBM case, we also note $\widehat{\mathbf{h}}_t^{(s)}$ and $\widetilde{\mathbf{h}}_t^{(s)}$ as the vector containing the conditional probabilities of the hidden units being equal to 1, conditioned on $(\mathbf{X}_t, \mathbf{y}_t)$ and $(\widetilde{\mathbf{X}}_t, \widetilde{\mathbf{y}}_t)$ respectively. Then, the Contrastive Divergence (CD) learning updates are computed as follows:

$$\begin{array}{lclcl}
\mathbf{b} & \leftarrow & \mathbf{b} & + & \lambda \sum_s \left(\mathbf{x}_t^{(s)} - \widetilde{\mathbf{x}}_t^{(s)}\right) \\
\mathbf{c} & \leftarrow & \mathbf{c} & + & \lambda \sum_s \left(\widehat{\mathbf{h}}_t^{(s)} - \widetilde{\mathbf{h}}_t^{(s)}\right) \\
\mathbf{d} & \leftarrow & \mathbf{d} & + & \lambda \left(\mathbf{y}_t - \widetilde{\mathbf{y}}_t\right) \\
\mathbf{W} & \leftarrow & \mathbf{W} & + & \lambda \sum_s \left(\widehat{\mathbf{h}}_t^{(s)} \mathbf{x}_t^{(s)\top} - \widetilde{\mathbf{h}}_t^{(s)} \widetilde{\mathbf{x}}_t^{(s)\top}\right) \\
\mathbf{U} & \leftarrow & \mathbf{U} & + & \lambda \sum_s \left(\widehat{\mathbf{h}}_t^{(s)} \mathbf{y}_t^\top - \widetilde{\mathbf{h}}_t^{(s)} \widetilde{\mathbf{y}}_t^\top\right).
\end{array}$$

### 3.2 ClassSetRBM for Sets with Redundant Evidence (OR)

There might be cases where the assumption of mutual exclusivity of the hidden units is too strong. One simple such case would be if additional copies of vectors were inserted in the set. More generally, it could be that the same useful hidden feature is present in several vectors within the input set. In this case, the actual number of vectors containing this evidence is not relevant, only the presence of that evidence in at least one set element is. It might then be desirable to have a model that is more robust to such situations.

To achieve this, we must somehow remove mutual exclusivity over the vectors $\mathbf{x}^{(s)}$ but maintain it for the connections with the target $\mathbf{y}$. This can be accomplished by having additional hidden layer "copies" $\mathbf{G} = \{\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(|\mathbf{X}|)}\}$ connected only to $\mathbf{y}$ and removing the direct connections of $\mathbf{H}$ to $\mathbf{y}$, yielding the new energy function

$$\begin{aligned}
E(\mathbf{X}, \mathbf{y}, \mathbf{H}, \mathbf{G}) = -\mathbf{d}^\top \mathbf{y} - \textstyle\sum_s \mathbf{b}^\top \mathbf{x}^{(s)} - \sum_s \mathbf{c}^\top \mathbf{h}^{(s)} \\
- \textstyle\sum_s \left( {\mathbf{h}^{(s)}}^\top \mathbf{W} \mathbf{x}^{(s)} + {\mathbf{g}^{(s)}}^\top \mathbf{U} \mathbf{y} \right) .
\end{aligned}$$

Then, dependencies between $\mathbf{X}$ and $\mathbf{y}$ are modeled by the hidden units through the imposition of the following constraints in the activities in $\mathbf{H}$ and $\mathbf{G}$:

$$\begin{array}{rl}
\sum_{s=1}^{|\mathbf{X}|} g_j^{(s)} \in \{0,1\} & \quad \forall j = 1 \ldots H \\
h_j^{(s)} \geq g_j^{(s)} & \quad \forall j = 1 \ldots H,\ \forall s = 1, \ldots, |\mathbf{X}| \ .
\end{array}$$

Hence, for a target hidden unit $g_j^{(s)}$ to be active, *at least* one input hidden units $h_j^{(s)}$ will need to be active. We call this second model ClassSetRBM$^{\text{OR}}$. Also see Figure 1 for an illustration of ClassSetRBM$^{\text{OR}}$.

It can be shown that ClassSetRBM$^{\text{OR}}$ has the following conditionals over $\mathbf{G}$ and $\mathbf{H}$:

$$\begin{array}{rcl}
p(\mathbf{G}|\mathbf{X}, \mathbf{y}) & = & \prod_{j=1}^{H} p(\{g_j^{(s)}\}_{s=1}^{|\mathbf{X}|} | \mathbf{X}, \mathbf{y}) \\
p(g_j^{(s)} = 1 | \mathbf{X}, \mathbf{y}) & = & \frac{\exp\left(\mathrm{act}_j(\mathbf{x}^{(s)}, \mathbf{y})\right)}{1 + \sum_{s'=1}^{|\mathbf{X}|} \exp\left(\mathrm{act}_j(\mathbf{x}^{(s')}, \mathbf{y})\right)} \\
p(g_j^{(\cdot)} = 0 | \mathbf{X}, \mathbf{y}) & = & \frac{1}{1 + \sum_{s'=1}^{|\mathbf{X}|} \exp\left(\mathrm{act}_j(\mathbf{x}^{(s')}, \mathbf{y})\right)}
\end{array}$$

$$\begin{array}{rcl}
p(\mathbf{H}|\mathbf{G}, \mathbf{X}, \mathbf{y}) & = & \prod_{s=1}^{|\mathbf{X}|} \prod_{j=1}^{H} p(h_j^{(s)} | g_j^{(s)}, \mathbf{x}^{(s)}) \\
p(h_j^{(s)} = 1 | g_j^{(s)}, \mathbf{x}^{(s)}) & = & \begin{cases} 1, \text{ if } g_j^{(s)} = 1 \\ \mathrm{sigm}(c_j + \mathbf{W}_{j\cdot} \mathbf{x}^{(s)}), \text{ else} \end{cases}
\end{array}$$

where we now have $\mathrm{act}_j(\mathbf{x}^{(s)}, \mathbf{y}) = \mathrm{softminus}(c_j + \mathbf{W}_{j\cdot} \mathbf{x}^{(s)}) + \mathbf{U}_{j\cdot} \mathbf{y}$, with $\mathrm{softminus}(a) = a - \mathrm{softplus}(a)$. The conditionals for $\mathbf{X}$ and $\mathbf{y}$ are the same as in Section 3.1, with the exception that $\mathbf{y}$ is conditioned on $\mathbf{G}$, not $\mathbf{H}$.

The class posterior is also tractable (see supplementary material). It is the same as in Section 3.1, but with a free-energy $F^{\mathrm{OR}}(\mathbf{X}, \mathbf{y})$ where the $\mathrm{softmax}_j(\mathbf{X})$ function is now

$$\mathrm{softmax}_j(\mathbf{X}) = \log\Big(\textstyle\sum_{s=1}^{|\mathbf{X}|} \exp(\mathrm{softminus}(c_j + \mathbf{W}_{j\cdot}\mathbf{x}^{(s)}))\Big).$$

Once again, discriminative and generative learning can be performed. When performing Gibbs sampling to compute the CD updates, the hidden layers $\mathbf{G}$ and $\mathbf{H}$ samples are obtained by first sampling from $p(\mathbf{G}|\mathbf{X}, \mathbf{y})$ and then sampling from $p(\mathbf{H}|\mathbf{G}, \mathbf{X}, \mathbf{y})$. Again denoting $(\widehat{\mathbf{h}}_t^{(s)}, \widehat{\mathbf{g}}_t^{(s)})$ and $(\widetilde{\mathbf{h}}_t^{(s)}, \widetilde{\mathbf{g}}_t^{(s)})$ the vectors of hidden probabilities from conditioning on $(\mathbf{X}_t, \mathbf{y}_t)$ and $(\widetilde{\mathbf{X}}_t, \widetilde{\mathbf{y}}_t)$ respectively, we obtain the following CD updates:

$$\begin{array}{lclcl}
\mathbf{b} & \leftarrow & \mathbf{b} & + & \lambda \sum_s \left(\mathbf{x}_t^{(s)} - \widetilde{\mathbf{x}}_t^{(s)}\right) \\
\mathbf{c} & \leftarrow & \mathbf{c} & + & \lambda \sum_s \left(\widehat{\mathbf{h}}_t^{(s)} - \widetilde{\mathbf{h}}_t^{(s)}\right) \\
\mathbf{d} & \leftarrow & \mathbf{d} & + & \lambda \left(\mathbf{y}_t - \widetilde{\mathbf{y}}_t\right) \\
\mathbf{W} & \leftarrow & \mathbf{W} & + & \lambda \sum_s \left(\widehat{\mathbf{h}}_t^{(s)} \mathbf{x}_t^{(s)\top} - \widetilde{\mathbf{h}}_t^{(s)} \widetilde{\mathbf{x}}_t^{(s)\top}\right) \\
\mathbf{U} & \leftarrow & \mathbf{U} & + & \lambda \sum_s \left(\widehat{\mathbf{g}}_t^{(s)} \mathbf{y}_t^{\top} - \widetilde{\mathbf{g}}_t^{(s)} \widetilde{\mathbf{y}}_t^{\top}\right) .
\end{array}$$

### 3.3 Variants with "hard" max pooling

In both proposed models, the class posterior $p(\mathbf{y}|\mathbf{X})$ require that some hidden units be implicitly "pooled" by taking a soft version of the maximum over the input set. Specifically, this is achieved through the $\log(\sum_{s=1}^{|\mathbf{X}} \exp(\cdot))$ operation in their definitions of $\mathrm{softmax}_j(\mathbf{X})$.

In practice, softmax pooling has the disadvantage that at the beginning of training, pooling essentially corresponds to summing the activations of all hidden units and does not select a single hidden unit. This potentially makes it harder for the hidden units to specialize.

Hence, we also consider "hard" max variants of ClassSetRBM$^{\mathrm{XOR}}$ and ClassSetRBM$^{\mathrm{OR}}$, referred to as ClassSetMaxRBM$^{\mathrm{XOR}}$ and ClassSetMaxRBM$^{\mathrm{OR}}$ respectively, where the $\log(\sum_{s=1}^{|\mathbf{X}} \exp(\cdot))$ operation is replaced by a $\max_{s=1,\ldots,|\mathbf{X}|}(\cdot)$ operation in their definition of $\mathrm{softmax}_j(\mathbf{X})$. This modification is only applied for computing $p(\mathbf{y}|\mathbf{X})$ and the discriminative gradients. This can be understood as approximating the true posterior $p(\mathbf{y}|\mathbf{X})$ by replacing parts of the sum over all the hidden units with a maximum, and optimizing the conditional log-likelihood of that approximation.

## 4 Related Work

As previously mentioned, the problem of classifying inputs corresponding to sets is closely related to that of multiple-instance learning (MIL). The standard case is binary classification, where a set of training vectors is labeled positive if and only if at least one instance in the set is positive. MIL has been studied to solve problems such as drug activity detection (Dietterich et al., 1997) and natural scene categorization. In the last fifteen years, several types of approaches have been proposed to address MIL, such as Learning Axis-Parallel Concepts (Dietterich et al., 1997), Diverse Density (Maron & Ratan, 1998) and its Expectation-Maximization version (Zhang & Goldman, 2001). Extensions of k-nearest neighbours (Citation kNN in Wang and Zucker (2000)), Support Vector Machines (MI-SVM in Andrews et al. (2002)), decisions trees (Chevaleyre & Zucker, 2001), perceptrons (Sabato et al., 2010) and neural networks (Zhou & Zhang, 2002) have also been explored. Classification within these approaches mainly consists of computing the maximum output over the vectors in the set, and a loss function to optimize is expressed accordingly. The approach presented in this paper rather consists in performing the (soft) maximum pooling over the sets in an intermediate latent representation, instead of on the output posterior probabilities as proposed by Zhou and Zhang (2002); Sabato et al. (2010).

Concerning classification of sets in general, several kernels between sets of vectors have been proposed to generalize kernel-based classifiers (SVM) without modifying the standard optimization problem. There are kernels defined between probabilistic density functions estimated on each set of vectors, such as Fisher kernels (Jaakkola & Haussler, 1998), Mutual Information kernels (Seeger, 2002), Probability product kernels (Jebara & Kondor, 2004; Lyu, 2005) and radial kernels based on a probabilistic distance such as Kullback-Leibler (Kondor & Jebara, 2003). These approaches have been developed with some particular families of density functions such as Gaussian distributions and Gaussian mixture models, and are not convenient when inputs are high-dimensional or sparse. There are also kernels based on combinations of kernels between inter-sets pairs of instances. These include several kinds of linear combination of kernels on inter-sets pairs of instances (Louradour et al., 2007; Zhou et al., 2009) as well as max kernels (Wallraven et al., 2003). The mi-Graph kernels of Zhou et al. (2009) actually achieves some of the best results reported on MIL tasks (Deselaers & Ferrari, 2010). The main disadvantage of such kernel-based SVM approaches is that they tend not to scale well with big datasets: the complexity of optimizing an SVM is quadratic in the number of training samples, and also the complexity of computing kernels between sets is quadratic in the number of vectors per set.

Finally, in the RBM literature, Lee et al. (2009) also explored the use of soft (probabilistic) pooling operations in a convolutional RBM. The two models proposed here can be seen as other pooling-based RBMs that are appropriate when the inputs are sets.

# 5 Experiments

We present here experiments on standard MIL datasets as well as on the problem of mail classification, which motivated this work. To evaluate the proposed RBMs for sets, we compare them with the following baseline models:

**ClassRBM-poolIn**: In this model, we simply apply ClassRBM described in section 2 on fixed-size vectors that are generated by pooling all the vectors in each input set using the maximum, minimum and average values of the vector features over that set.

**ClassRBM-maxOut**: This model is an implementation for a ClassRBM of the ideas in Zhou and Zhang (2002); Sabato et al. (2010). The model is trained by gradient descent to predict the target based only on the input vector that gives the maximal output response. We also apply the same strategy with logistic regression (Logit-maxOut) and a one hidden layer perceptron (MLP-maxOut). Note that these methods are only applicable in the case of binary classification, so we only use these baselines for MIL problems[3].

**SVM-miGraph**: This state-of-the-art SVM model based on a kernel between sets gave some of the best results on MIL tasks as reported by Deselaers and Ferrari (2010). The miGraph kernel (Zhou et al., 2009) is:

$$K_{mi}(\mathbf{X_1}, \mathbf{X_2}) = \frac{\sum_s \sum_{s'} w_s(\mathbf{X_1}) w_{s'}(\mathbf{X_2}) k(\mathbf{x_1}^{(s)}, \mathbf{x_2}^{(s')})}{\sum_s w_s(\mathbf{X_1}) \sum_{s'} w_{s'}(\mathbf{X_2})}$$

where the vectorial kernel $k$ is the Gaussian kernel and where the weights assigned to a vector within a set is inversely proportional to the number of "edges" that can be drawn with the other vectors in the same set:

$$w_s(\mathbf{X}) = 1/\sum_{s'} \mathbb{1}_{\|\mathbf{x}^{(s)} - \mathbf{x}^{(s')}\| < \sigma(\mathbf{X})}$$

given an adaptive distance threshold $\sigma(\mathbf{X})$ defined as the average pairwise distance within the set:

$$\sigma_{mi}(\mathbf{X}) = \frac{T(T-1)}{2} \sum_{s \neq s'} \|\mathbf{x}^{(s)} - \mathbf{x}^{(s')}\| \ .$$

**SVM-miGraph2**: This system is a variant of SVM-miGraph where the distance threshold to compute the graphs is the same for all sets[4] $\sigma_{mi2}(\mathbf{X}) = \sigma_0$. This hyper-parameter is tuned by validation, such as the $C$ in the SVM loss and the $\gamma$ of the Gaussian kernel.

**SVM-max** : Like miGraph kernels, local kernels for sets (Wallraven et al., 2003) are computed from Gaussian kernel values on all inter-sets pairs of vectors. Instead of computing a weighted average, only the kernel values that are maximal for each vector are summed.

In all our experiments, we perform k-fold cross validation, and at each fold the model hyper-parameters are optimized on a subset of training inputs (20%), not used to train the model. The reported results are obtained by averaging over all test fold examples. For all models except the SVMs, we train by stochastic gradient descent: the hyper-parameters are the learning rate and the number of updates (early-stopping)[5]. The number of hidden neurons used was 100 (varying this number had little influence on the results). RBMs were either trained discriminatively only or using the hybrid objective.

## 5.1 Experiments on MIL benchmark

We start by evaluating our approach on the public and popular MIL datasets[6]: *Musk1*, *Musk2* (drug activity prediction) and *Elephant*, *Fox*, *Tiger* (image annotation). We carried out 5-times repeated 10-fold cross-validation. Table 1 shows the results of the several proposed variants of ClassSetRBM and of the baseline models. For each dataset, the best performance is indicated in a gray cell, and results in bold are the ones with no significant difference with this best reference, based on a 95% two-sided paired Student's t-test. Overall, we see the ClassSetRBM variants obtain good results compared to the many baselines. In particular, the best performing variant, ClassSetMaxRBM$^{\text{XOR}}$ , has the highest average accuracy over all datasets and is never statistically worse than the best reference. Most importantly, ClassSetMaxRBM$^{\text{XOR}}$ clearly outperforms ClassRBM-poolIn and ClassRBM-maxOut, which confirms the usefulness of having a pooling mechanism at the level of the hidden layer, as opposed to at the input or output level. Hybrid training does not clearly improve over purely discriminative training. This might be explained by the fact that the ClassSetRBMs modeled the input units (scaled in $[0, 1]$) as binary variables. The use of a "hard" max pooling however does appear to be quite useful and almost consistently improves on the softmax variant. In fact, further experiments showed that the softmax ClassSetRBM$^{\text{XOR}}$ is always able to reach the same performance as its "hard" max variant when initialized

[3] We tried some variants to generalize to multiclass, but the performance was always poor.

[4] Personal communication with (Zhou et al., 2009).

[5] To initialize gradient descent, we set all biases to 0 and weights are randomly chosen using a uniform distribution in $[-b, b]$ where $b = 1/\sqrt{\textit{input size}}$ .

[6] `http://www.cs.columbia.edu/∼andrews/mil/datasets.html`

Table 1: Classification accuracies (%) on MIL datasets

| Model | Training | Musk1 | Musk2 | Elephant | Fox | Tiger | Average |
|---|---|---|---|---|---|---|---|
| ClassSetRBM$^{\mathrm{XOR}}$ | Discriminative | **83.04** | 80.39 | 82.30 | 58.50 | **82.40** | 77.33 |
| ClassSetRBM$^{\mathrm{OR}}$ | | 82.61 | **83.73** | 82.70 | **58.60** | **80.90** | 77.71 |
| ClassSetMaxRBM$^{\mathrm{XOR}}$ | | **83.91** | **84.12** | **87.80** | **60.30** | **82.60** | 79.75 |
| ClassSetMaxRBM$^{\mathrm{OR}}$ | | **85.65** | 80.39 | **87.10** | **59.70** | **82.60** | 79.09 |
| ClassSetRBM$^{\mathrm{XOR}}$ | Hybrid | **84.57** | **84.12** | 82.80 | 55.70 | **82.10** | 77.86 |
| ClassSetRBM$^{\mathrm{OR}}$ | | **84.35** | 84.71 | 82.60 | 55.50 | **82.70** | 77.97 |
| ClassSetMaxRBM$^{\mathrm{XOR}}$ | | **83.70** | **81.18** | **86.40** | 58.00 | **83.20** | 78.50 |
| ClassSetMaxRBM$^{\mathrm{OR}}$ | | **85.87** | **81.76** | 85.50 | 56.60 | **82.60** | 78.47 |
| ClassRBM-poolIn | - | 81.52 | **81.37** | 82.70 | **59.80** | 76.80 | 76.44 |
| ClassRBM-maxOut | | **83.91** | 80.98 | 81.60 | 57.60 | 75.50 | 75.92 |
| MLP-maxOut | | **85.65** | 78.82 | 82.00 | 55.40 | 74.40 | 75.25 |
| Logit-maxOut | | 81.09 | 80.00 | 81.90 | **58.80** | 75.90 | 75.54 |
| SVM-miGraph | - | **86.74** | **82.35** | 83.80 | **61.50** | **81.20** | 79.12 |
| SVM-miGraph2 | | **85.43** | **82.35** | 83.80 | **61.30** | **80.80** | 78.74 |
| SVM-max | | 83.48 | **84.51** | 84.60 | **59.70** | **81.70** | 78.80 |

with the "hard" max trained parameters. This shows that the improvement using "hard" max is not due to differences in data models but to better learning.

### 5.2 Experiments on mail categorization

The task which motivated this work is image document categorization, where the documents are pieces of mail that can be considered as sets of pages. These pages can be printed or handwritten letters, official papers, forms, envelops or white pages. The main application is mailroom automation, which is of great interest for large organizations where the volume of incoming mail can reach tens of thousands per day and must be processed within a couple of days. Routing of documents can then be done automatically with a classifier embedded in the document management system.

Each image of a page is processed by an OCR for printed and handwritten text, which produces binary input features that correspond to the presence/absence of a given word in the page. The vocabulary size is limited to the 10 000 most frequently recognized words. Other features from image analysis are also appended: A sub-resolution image (composed of average gray-scaled pixel values from a $6 \times 8$ regular grid); 17 geometric statistics of Document Layout Analysis (based on a segmentation of the image into boxes, lines, printed and handwritten text zones); 18 predefined page class detectors, engineered to detect common types of pages, such as checks and official papers (each output recognition score is used as a feature).

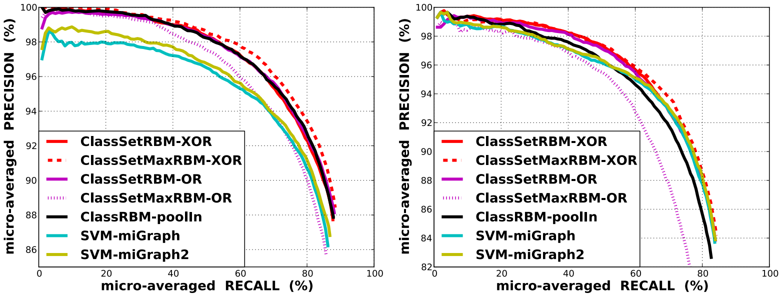
The resulting feature vectors are high-dimensional, sparse, and noisy to some extent (the word error rate of OCR typically lies between 5% and 50%). The application of mail classification typically does not fit well the assumptions made in MIL. In particular, we expect each page to provide only partial clues for predicting the set label, such that considering label assignments at the page level is not natural. In other words, while there might be enough informative pages to confidently identify a set's label, this does not imply that this label is appropriate for any one of these pages individually. Moreover, this problem is a multiclass classification problem, while most MIL algorithms are developed for binary classification and do not always generalize clearly to the multiclass setting.

We carried out experiments on two collected datasets for mail classification: *DsDe* (14 593 pieces of mail, 102 071 pages, 11 classes) and *DsUs* (16 808 pieces of mail, 160 372 pages, 6 classes). The class labels are related to the purpose of the mail: change of address, request for a business operation, etc. Figure 2(a) shows the 5-fold cross-validation average accuracy of different models. It is also important in mailroom automation to be able to reject pieces of mail for which the prediction is too uncertain: the pieces of mail rejected by the system can then be processed by a human agent, in order to limit classification errors of the whole process. The rejection is done by comparing the classifier's output confidence to a threshold, thus this confidence estimate has to be reliable. A standard way to evaluate the goodness of the output confidence scores in multiclass classification is to plot *micro-averaged* recall and precision (Sebastiani, 2002) for different values of the rejection threshold. This is shown in Figure 2(b,c).

Again, we observe that ClassSetMaxRBM$^{\mathrm{XOR}}$ achieves the best performance overall. We emphasize that, for both these two large datasets, ClassSetMaxRBM$^{\mathrm{XOR}}$ was much more efficient than

| Model (disc. training) | DsUs | DsDe |
|---|---|---|
| ClassSetRBM$^{XOR}$ | 87.71 | 83.75 |
| ClassSetRBM$^{OR}$ | **88.13** | **83.86** |
| ClassSetMaxRBM$^{XOR}$ | **88.55** | **84.18** |
| ClassSetMaxRBM$^{OR}$ | 85.69 | 78.43 |
| ClassRBM-poolIn | **87.97** | 82.54 |
| SVM-miGraph | 86.22 | **83.74** |
| SVM-miGraph2 | 86.82 | **84.16** |
| SVM-max | 56.45 | 64.14 |

(a) Classification accuracies (%)



(b) Precision/Recall curves on DsUs

(c) Precision/Recall curves on DsDe

Figure 2: Mail classification results, with classification accuracies (a) and precision/recall curves (b,c).

the SVM approaches: at least 10 times faster to train, and more than 1000 times faster to classify.

## 6 Conclusion

We described how the classification restricted Boltzmann machine could be adapted to problems where the inputs correspond to sets of vectors. Different generalizations for this problem were investigated, with one of these variant achieving consistent, competitive performance on multiple-instance learning datasets. It was also applied with success to a mail categorization task. Our experiments confirm the usefulness of pooling at the hidden representation level, as opposed to the input or output level. Directions for future work include applying this framework in deep neural architectures.

## References

Andrews, S., Tsochantaridis, I., & Hofmann, T. (2002). Support vector machines for multi-instance learning. *proc. NIPS*.

Chevaleyre, Y., & Zucker, J. (2001). Solving multi-instance learning and multiple-part learning problems with decision trees and rule sets. *Lecture Notes in Artificial Intelligence*, *2056*.

Deselaers, T., & Ferrari, V. (2010). A conditional random field for multi-instance learning. *proc. ICML*.

Dietterich, T., Lathrop, R., & Lozano-Perez, T. (1997). Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence Journal*, *89*.

Hinton, G., Osindero, S., & Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*.

Jaakkola, T., & Haussler, D. (1998). Exploiting generative models in discriminative classifiers. *advances in NIPS*, *11*.

Jebara, T., & Kondor, R. (2004). Probability product kernels. *Journal of Machine Learning Research*, *5*.

Kondor, R., & Jebara, T. (2003). A kernel between sets of vector. *proc. ICML*.

Larochelle, H., & Bengio, Y. (2008). Classification using discriminative restricted boltzmann machines. *proc. ICML*.

Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *proc. ICML*.

Louradour, J., Daoudi, K., & Bach, F. (2007). Feature space mahalanobis sequence kernels: Application to svm speaker verification. *IEEE trans. on Speech and Audio Processing*, *15*.

Lyu, S. (2005). A kernel between unordered sets of data: the gaussian mixture approach. *proc. ECML*.

Maron, O., & Ratan, A. (1998). Multi-instance learning for natural scene classification. *proc. ICML*.

Reynolds, D. (1995). Speaker identification and verification using gaussian mixture models. *Speech Communication*, *17*.

Sabato, S., Srebreo, N., & Tishby, N. (2010). Reducing label complexity by learning from bags. *proc. AISTATS*.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, *34*.

Seeger, M. (2002). Covariance kernels from bayesian generative models. *advances in NIPS*, *14*.

Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. *proc. ICML*.

Wallraven, C., Caputo, B., & Graf, A. (2003). Recognition with local features: the kernel recipe. *proc. ICCV*.

Wang, J., & Zucker, J.-D. (2000). Solving the multi-instance problem: a lazy learning approach. *proc. ICML*.

Welling, M., Rosen-Zvi, M., & Hinton, G. E. (2005). Exponential family harmoniums with an application to information retrieval. *advances in NIPS*, *17*.

Zhang, Q., & Goldman, S. (2001). Em-dd: An improved multi-instance learning technique. *proc. NIPS*.

Zhou, Z.-H., Sun, Y.-Y., & Li, Y.-F. (2009). Probabilistic multi-instance learning by treating instances as non-i.i.d. samples. *proc. ICML*.

Zhou, Z.-H., & Zhang, M. (2002). Neural networks for multi-instance learning. *proc. ICIIT*.