Max-Margin Boltzmann Machines for Object Segmentation

Jimei Yang, Simon Sáfár, Ming-Hsuan Yang
University of California, Merced
{jyang44,ssafar,mhyang}@ucmerced.edu

Abstract

We present Max-Margin Boltzmann Machines (MMBMs) for object segmentation. MMBMs are essentially a class of Conditional Boltzmann Machines that model the joint distribution of hidden variables and output labels conditioned on input observations. In addition to image-to-label connections, we build direct image-to-hidden connections to facilitate global shape prediction, and thus derive a simple Iterated Conditional Modes algorithm for efficient maximum a posteriori inference. We formulate a max-margin objective function for discriminative training, and analyze the effects of different margin functions on learning. We evaluate MMBMs using three datasets against state-of-the-art methods to demonstrate the strength of the proposed algorithms.

1. Introduction

Object segmentation can be formulated as a structured output problem that involves making predictions collectively over correlated output labels $\mathbf{y} \in \mathcal{Y}$ from input observations $\mathbf{x} \in \mathcal{X}$. One of the core issues in structured output prediction problems is how to represent complex output variable interrelations effectively while carrying out inference and learning efficiently.

In Markov Random Fields (MRFs), output structures are represented by pairwise and high-order potential functions $p(\mathbf{y}) = \prod_{\mathbf{y}_i \subset \mathbf{y}} \phi(\mathbf{y}_i)/\mathbf{Z}$ where $\mathbf{Z}$ is the partition function. The prediction from the observations $\mathbf{x}$ to the labels $\mathbf{y}$ is usually realized in the conditional models $p(\mathbf{y}|\mathbf{x})$, i.e., Conditional Random Fields (CRFs) [15], which allow flexible use of various long-range features from observations $\mathbf{x}$. Pairwise potentials [22], although admitting efficient inference, can only capture limited local structure, such as smoothness and edges. High-order potentials are able to capture long-range interactions between pixel labels through bottom-up segmentation [13], pattern-based priors [19, 21]. Beyond the generic high-order priors, the ObjCut algorithm [14] introduces category-specific object models into MRFs and has shown good segmentation performance on articulated objects. In ObjCut, the hidden variables of pictorial structures encode the positions of object parts, but their interactions with pixel labels are manually designed.

Alternatively, Restricted Boltzmann Machines (RBMs) render more flexible models for structured output representation that learn high-order interrelations through a joint distribution of labels and a set of hidden (latent) variables $\mathbf{h}$ in $p(\mathbf{y}, \mathbf{h})$. By omitting lateral connections in a single layer, RBMs admit efficient inference and sampling from conditional probabilities. When operating with a small number of training samples, layered architectures [20] have been shown more effective in terms of model expressiveness and learning efficiency. Eslami et al. [8] propose a two-layer Boltzmann Machine $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2)$ (where $\mathbf{h}^1$ and $\mathbf{h}^2$ denote hidden variables in two layers) for modeling object shapes (ShapeBMs), and apply it onto object segmentation in a generative model $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2, \mathbf{x})$ [9].

In this paper, we present a general class of Conditional Boltzmann Machines (CBMs) for object segmentation in the form of $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ and $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2|\mathbf{x})$. In addition to the connections from image to labels, our models also include the connections from the image to hidden variables, which allows direct shape inference from image features. Based on layer-wise conditional independence of BMs, we derive a simple but efficient Iterated Conditional Modes [3] algorithm for maximum a posteriori (MAP) inference.

Learning with CRFs and CBMs is challenging as it requires handling exponentially large numbers of output combinations in data-dependent partition functions. Approximate learning algorithms are easily trapped in local optima, thereby limiting their generalization performance. Another line of research for structured output prediction is developed on max-margin formulations [23, 24, 11], that facilitates model generalizability to unseen test data. This technique has been applied to CRFs for object segmentation [22, 2]. In a similar spirit, we propose a max-margin formulation of CBMs, referred as MMBMs, and develop an online Concave-Convex Procedure (CCCP) [28] algorithm for learning efficiently with hidden variables. Note that large margin BMs have been proposed in [17] with a focus on theoretical analysis while our max-margin method is proposed for training a particular class of CBMs with applications to object segmentation. We investigate the effects of four kinds of margin functions on discriminative training, and demonstrate the importance of combined hidden and visible margin functions. We study two variants of MMBMs with a single hidden layer $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ as well as two hidden layers $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2|\mathbf{x})$, and compare them with two state-of-the-art models: superpixel based CRFs [1] and Compositional High Order Pattern Potentials [16]. We carry out experiments on the Weizmann horse [5], Penn-Fudan pedestrian [4] and Caltech-UCSD birds 200 [27] datasets. Experimental results show that the proposed MMBMs perform better than existing methods both quantitatively and qualitatively.

2. Related Work

Recent work [12, 16] on object segmentation realizes the power of Boltzmann Machines to represent high-order interactions in combining RBMs with CRFs. Li et al. [16] combine pairwise, data-dependent potentials with a one-layer RBM prior in CRFs (referred as Compositional High Order Pattern Potentials (CHOPPs) in Figure 1(b)), and show the relationship between the marginalized RBM free energy and high-order potentials [19]. Kae et al. [12] augment CRFs with an RBM shape prior in a two-layer model for image labeling. Their lower layer has nodes for every superpixel of the image, with pairwise weights connecting them. The labels for this layer are then pooled into a raster structure, enabling them to use a RBM to provide shape priors. Another attempt is to combine Deep Boltzmann Machines shape prior with a variational segmentation model [7], showing the effectiveness of strong shape priors for simple object segmentation. In all of the above approaches, the only inference pathway between the image features $\mathbf{x}$ and the hidden variables $\mathbf{h}$ representing shapes leads through the labels assigned to image pixels $\mathbf{y}$ while the shape only works as a prior. To perform inference and learning, the hidden variables are usually marginalized through an EM-like procedure. The shape information is thus not fully explored. In contrast, our MMBM models introduce connections between hidden variables $\mathbf{h}$ and image features $\mathbf{x}$, which enables a more efficient MAP inference procedure and thus max-margin learning.

3. Models

In this section, we first introduce two variants of Boltzmann Machines, RBMs and ShapeBMs, for modeling object shapes, and then describe the proposed conditional models and the maximum a posteriori inference algorithm.

3.1. Boltzmann Machines

Given a labeled image of an object, we represent the mask as a set of visible variables $\mathbf{y} \in \{0, 1\}^n$. RBMs use one layer of hidden variables $\mathbf{h} \in \{0, 1\}^m$ to capture global dependencies between visible variables (See Figure 1(a))

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

where $\mathbf{Z}$ is the partition function. RBMs do not have lateral connections within visible and hidden layers so that the energy function takes the form,

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

parametrized by $\mathbf{b}$, $\mathbf{c}$ and $\mathbf{W}$. One attractive property of RBMs is that visible variables are conditionally independent given hidden variables and vice versa. The conditional probability of each variable is essentially the sigmoid function $\sigma(y) = 1/(1 + \exp(-y))$,

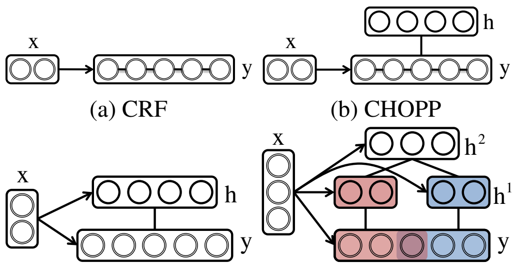
$$p(y_i = 1|\mathbf{h}) = \sigma(\sum_j w_{ij} h_j + b_i), \tag{3}$$

$$p(h_j = 1|\mathbf{y}) = \sigma(\sum_i w_{ij} y_i + c_j), \tag{4}$$

which facilitates efficient inference.

Although RBMs have the capacity of modeling complex distributions, they require a large set of hidden variables and numerous training examples. For object segmentation, it is labor intensive to collect a large number of training examples with ground truth masks, and challenging to train RBMs with a large set of variables. It is, however, possible to ameliorate this problem by considering the spatial structure of images. Eslami et al. [8] propose a particular form of Boltzmann Machine with two hidden layers $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2)$ (referred as ShapeBM) for object shape modeling. The first layer of hidden variables $\mathbf{h}^1$ is partitioned into several disjoint subsets $\{\mathbf{h}^1_k = \mathbf{h}^1(\mathbf{J}_k)\}_{k \in \mathbf{G}}$ of same size m^1_k, where $\mathbf{J}_k \in \{0, 1\}^m$ denotes the subset indexing. Each of them has a restricted receptive field and only connects to a local patch of the object mask. The local patches $\{\mathbf{y}_k = \mathbf{y}(\mathbf{I}_k)\}_{k \in \mathbf{G}}$ have the same size n_k and they overlap each other along the boundaries, where $\mathbf{I}_k \in \{0, 1\}^n$ denotes the patch index. Therefore, the pairwise potentials between visible variables $\mathbf{y}$ and the first layer hidden variables $\mathbf{h}^1$ can be represented by $\sum_{k \in \mathbf{G}} \mathbf{y}_k^\top \mathbf{W}^1_k \mathbf{h}^1_k$. Furthermore, different patches can share the same weights $\mathbf{W}^1 = \mathbf{W}^1_k, k \in \mathbf{G}$. The second layer of hidden variables $\mathbf{h}^2$ connects to all the variables $\mathbf{h}^1$ of the first layer. Similar to RBMs, there are no lateral connections between variables within any single layer. The energy function can be thus described by

$$\begin{aligned}\mathbf{E}(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2) = -\sum_{k \in \mathbf{G}} \mathbf{y}_k^\top \mathbf{W}^1 \mathbf{h}^1_k - \mathbf{b}^\top \mathbf{y} - \\ \mathbf{c}^{1\top} \mathbf{h}^1 - \mathbf{h}^{1\top} \mathbf{W}^2 \mathbf{h}^2 - \mathbf{c}^{2\top} \mathbf{h}^2.\end{aligned} \tag{5}$$



(a) CRF (b) CHOPP

(c) MMBM1 (d) MMBM2

Figure 1. Comparing graphical models of MMBMs ((c) and (d)) with pairwise CRF (a) and CHOPP [16]. The edges mean full connections between two layers. In (d), the connections between $\mathbf{y}$ and $\mathbf{h}^1$ only involve the variables of the same color.

The pairwise term of the first layer can be rewritten in the same form as RBMs by some matrix manipulation:

$$\sum_{k\in\mathbf{G}} \mathbf{y}_k^\top \mathbf{W}^1 \mathbf{h}_k^1 = \mathbf{y}^\top \tilde{\mathbf{W}}^1 \mathbf{h}^1, \qquad \text{where } \tilde{\mathbf{W}}^1(\mathbf{I}_k, \mathbf{J}_k) = \mathbf{W}^1. \tag{6}$$

Due to its structure, ShapeBM uses much fewer parameters than conventional two-layer RBMs [20], thereby facilitating efficient learning for smaller datasets. The pairwise term $\mathbf{y}^\top\tilde{\mathbf{W}}^1\mathbf{h}^1$ models the compatibility between pixels and parts while the term $\mathbf{h}^{1\top}\mathbf{W}^2\mathbf{h}^2$ defines the possible configuration of parts. Thus, when an unit of $\mathbf{h}^1$ is activated, a template stored in $\mathbf{W}^1$ is selected to enforce the group of pixels to obey a binary image pattern. Also, when an unit of $\mathbf{h}^2$ is activated, it triggers a particular configuration of parts (due to varying pose or viewpoint). The ShapeBM architecture also enjoys the property of conditional independence $p(\mathbf{y}|\mathbf{h}^1), p(\mathbf{h}^1|\mathbf{y}, \mathbf{h}^2)$ and $p(\mathbf{h}^2|\mathbf{h}^1)$, although exact inference is not tractable for this model.

3.2. Conditional Boltzmann Machines

While generative RBMs and ShapeBMs are capable of modeling object shape priors, it is still challenging to efficiently infer a binary object mask $\mathbf{y}$ from an image $\mathbf{x}$. Intuitively, we can construct a fully generative model for object images and their binary masks $p(\mathbf{y}, \mathbf{x})$ such that object shape can be inferred from an image by the conditional distribution $p(\mathbf{y}|\mathbf{x})$, and an image generated from a shape mask by $p(\mathbf{x}|\mathbf{y})$. As an example, Eslami et al. [9] present a generative multinomial joint model of appearance (object images) and shape (parts-based segmentation).

Nevertheless, constructing a joint model of object images and shape masks poses significant difficulties as the conditional distribution of images given the shape masks are intrinsically multimodal and full of ambiguities. In order to estimate the object mask $\mathbf{y}$ from an image $\mathbf{x}$, we instead propose to directly train the conditional models $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ for RBMs (MMBM1, Figure 1(c)) and $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2|\mathbf{x})$ for ShapeBMs (MMBM2, Figure 1(d)). In these conditional models, the activations of variables depend on the observations or image features, so the energy function of $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ can be represented by

$$\mathbf{E}(\mathbf{y}, \mathbf{h}, \mathbf{x}) = -\mathbf{y}^\top\mathbf{W}\mathbf{h} - \mathbf{h}^\top(\mathbf{V}^1\mathbf{x}^1 + \mathbf{c}) - \mathbf{y}^\top(\mathbf{V}^0\mathbf{x}^0 + \mathbf{b}), \tag{7}$$

while the energy function of $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2|\mathbf{x})$ takes the form,

$$\begin{aligned}\mathbf{E}(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2, \mathbf{x}) &= -\mathbf{y}^\top\tilde{\mathbf{W}}^1\mathbf{h}^1 - \mathbf{h}^{1\top}\mathbf{W}^2\mathbf{h}^2 \\ &- \mathbf{h}^{1\top}(\mathbf{V}^1\mathbf{x}^1 + \mathbf{c}^1) - \mathbf{h}^{2\top}(\mathbf{V}^2\mathbf{x}^2 + \mathbf{c}^2) - \mathbf{y}^\top(\mathbf{V}^0\mathbf{x}^0 + \mathbf{b}).\end{aligned} \tag{8}$$

In the above equations, $\mathbf{x}^0$ represents low-level image features that indicate foreground and background assignments. The variable $\mathbf{x}^1$ represents features of object parts and $\mathbf{V}^1$ contains templates of object parts. The variable $\mathbf{x}^2$ describes the holistic object features, and $\mathbf{V}^2$ is composed of object templates of different poses and viewpoints. In these two models, we connect the observations $\mathbf{x}$ to both visible and hidden layers, which enables the direct inference pathway from image features to shapes.

3.3. MAP Inference

Given a set of image features $\mathbf{x}$, the most likely configuration of $\mathbf{y}$ is computed from

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x}). \tag{9}$$

In the proposed MMBM with single hidden layer, the marginal distribution $p(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ can be represented by its free energy form $\exp(-F(\mathbf{y}, \mathbf{x}))/\mathbf{Z})$, and

$$\begin{aligned}-F(\mathbf{y}, \mathbf{x}) =& \mathbf{y}^\top(\mathbf{V}^0\mathbf{x}^0 + \mathbf{b}) + \\ & \sum_j \log(1 + \exp(c_j + \mathbf{y}^\top\mathbf{W}_{\cdot j} + \mathbf{V}^1_{j\cdot}\mathbf{x}^1)).\end{aligned} \tag{10}$$

where $\mathbf{W}.j$ and $\mathbf{W}_{j.}$ denote j-th column and row of $\mathbf{W}$, respectively. As the partition function $\mathbf{Z}$ is constant given $\mathbf{x}$, the MAP inference in (9) is exactly equivalent to optimizing the free energy function

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} -F(\mathbf{y}, \mathbf{x}). \tag{11}$$

Note that the free energy $F(\mathbf{y}, \mathbf{x})$ is not a linear function of $\mathbf{y}$, and we need to take gradients to find the optimal $\hat{\mathbf{y}}$. However, the analytic free energy is not available in the MMBM with two hidden layers. We instead optimize the variational upper bound of log-likelihood $\log p(\mathbf{y}|\mathbf{x})$ using the EM algorithm in spirit similar to techniques that have been effectively applied to training generative BMs. However, in MMBMs, both visible $\mathbf{y}$ and hidden variables are conditioned on input variables $\mathbf{x}$. The conditional distributions $p(\mathbf{y}|\mathbf{h}^1, \mathbf{x})$, $p(\mathbf{h}^1|\mathbf{y}, \mathbf{h}^2, \mathbf{x})$ and $p(\mathbf{h}^2|\mathbf{h}^1, \mathbf{x})$ are likely

Algorithm 1 MAP inference by the ICM algorithm.

```
Initialize $\mathbf{h}^1$
while do not converge do
    $\mathbf{h}^2 \leftarrow \max p(\mathbf{h}^2|\mathbf{h}^1, \mathbf{x})$
    $\mathbf{y} \leftarrow \max p(\mathbf{y}|\mathbf{h}^1, \mathbf{x})$
    $\mathbf{h}^1 \leftarrow \max p(\mathbf{h}^1|\mathbf{y}, \mathbf{h}^2, \mathbf{x})$
end while
```

highly peaked, if not unimodal, and thus they can be approximated by optimizing

$$\{\hat{\mathbf{y}}, \hat{\mathbf{h}}^1, \hat{\mathbf{h}}^2\} = \arg\max p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2|\mathbf{x}). \tag{12}$$

Similar to the block Gibbs sampling method, the independent property of conditional distributions induces an efficient Iterated Conditional Modes (ICM) algorithm (See Algorithm 1). The ICM algorithm also provides a good approximate solution to the free energy optimization problem in (11) and (10) for single layer MMBMs. Essentially, the second term in (10) can be approximated by

$$\begin{aligned}&\sum_j \log(1 + \exp(c_j + \mathbf{y}^\top \mathbf{W}_{.j} + \mathbf{V}^1_{j.}\mathbf{x}^1)) \approx \\ &\max_{\mathbf{h}}(\mathbf{c}^\top \mathbf{h} + \mathbf{y}^\top \mathbf{W}\mathbf{h} + \mathbf{h}^\top \mathbf{V}^1 \mathbf{x}^1),\end{aligned} \tag{13}$$

which can be solved by the ICM algorithm.

4. Learning

Given a training set of object image-mask pairs $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$, we learn MMBMs for object segmentation. As the proposed learning algorithm can be applied to both MMBMs with single ($p(\mathbf{y}, \mathbf{h}|\mathbf{x}; \omega)$) or two hidden layers ($p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2|\mathbf{x}; \omega)$), we denote the MMBM by a general form $p(\mathbf{y}, \mathbf{H}|\mathbf{x}; \omega)$ where $\mathbf{H} = \mathbf{h}$ for one single hidden layer or $\mathbf{H} = \{\mathbf{h}^1, \mathbf{h}^2\}$ for two hidden layers, and $\omega = \{\mathbf{W}^{1,2}, \mathbf{V}^{0,1,2}, \mathbf{c}^{1,2}, \mathbf{b}\}$ are the model parameters. The MMBMs consist of image-independent and image-dependent parts. We first initialize the image-independent part by generative pre-training, and then reformulate the joint learning problem into a max-margin optimization task which is solved effectively by a CCCP algorithm.

4.1. Pre-training

Generative pre-training $p(\mathbf{y}, \mathbf{H})$ is of crucial importance for the MMBM models. It provides the MMBM models with proper regularization between output and hidden variables, and feed sensible hidden variables to discriminative learning in the following stage. By omitting image-dependent components, the MMBM with one single hidden layer reduces to the RBM while the one with two hidden layers reduces to the ShapeBM. We thus can utilize the generative training algorithms of these methods. Indeed, the general training procedure of BMs requires minimizing the differences between the data-dependent and model-dependent expectations. We train the RBM by minimizing contrastive divergence [10]. For the ShapeBM, each layer is greedily trained.

Algorithm 2 Stochastic Gradient Descent algorithm for max-margin learning MMBMs.

```
Set $t = 0$, initialize $\omega_0, \alpha_0$ and define $\gamma$
while $t < T$ do
    Randomly select a training instance $(\mathbf{x}_i, \mathbf{y}_i)$
    Solve (16): $\mathbf{H}_i^* \leftarrow \max_{\mathbf{H}}[-E(\mathbf{y}_i, \mathbf{H}, \mathbf{x}_i; \omega_t)]$
    Solve (17): $\hat{y}_i, \hat{\mathbf{H}}_i \leftarrow \max_{\mathbf{y},\mathbf{H}}[-E(\mathbf{y}, \mathbf{H}, \mathbf{x}_i; \omega_t) + \Delta(\mathbf{y}, \mathbf{y}_i, \mathbf{H}, \mathbf{H}_i^*)]$
    Update $\omega_{t+1} \leftarrow (1 - \alpha_t\gamma)\omega_t + \alpha_t(\frac{\partial E(\hat{y}_i, \hat{\mathbf{H}}_i, \mathbf{x}_i; \omega)}{\partial \omega} - \frac{\partial E(\mathbf{y}_i, \mathbf{H}_i^*, \mathbf{x}_i; \omega)}{\partial \omega})$
    Decrease $\alpha_t$
end while
```

4.2. Max-Margin Learning

To generate accurate prediction on test images, we seek for the parameters ω that assign training labels $\mathbf{y}_i$ a greater than or equal log-likelihood of any other labeling $\mathbf{y}$ for instance i,

$$\log p(\mathbf{y}_i, \mathbf{H}|\mathbf{x}_i; \omega) \geq \log p(\mathbf{y}, \mathbf{H}|\mathbf{x}_i; \omega), \forall \mathbf{H}, \forall \mathbf{y}, \forall i. \tag{14}$$

We can cancel the partition function $\mathbf{Z}$ for both sides of (14), and express the constraints by energies,

$$-E(\mathbf{y}_i, \mathbf{H}, \mathbf{x}_i; \omega) \geq -E(\mathbf{y}, \mathbf{H}, \mathbf{x}_i; \omega), \forall \mathbf{H}, \forall \mathbf{y}, \forall i. \tag{15}$$

We refer the left term of (15) as data-dependent energy and the right term as model-dependent energy. Since the number of constraints in (15) is exponentially large, we look for the hidden variables $\mathbf{H}_i^*$ that best explain the training instance $(\mathbf{x}_i, \mathbf{y}_i)$ in the data-dependent energy

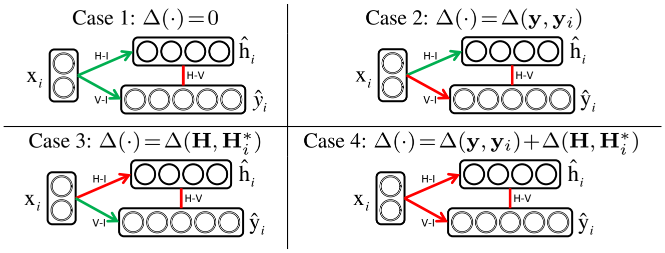
$$\mathbf{H}_i^* = \arg\max_{\mathbf{H}} - E(\mathbf{y}_i, \mathbf{H}, \mathbf{x}_i; \omega). \tag{16}$$

For the model dependent energy, we compute the best prediction from $\mathbf{x}_i$ by augmenting an energy margin $\Delta(\mathbf{y}, \mathbf{y}_i, \mathbf{H}, \mathbf{H}_i^*)$,

$$\{\hat{y}_i, \hat{\mathbf{H}}_i\} = \arg\max_{\mathbf{y},\mathbf{H}} - E(\mathbf{y}, \mathbf{H}, \mathbf{x}_i; \omega) + \Delta(\mathbf{y}, \mathbf{y}_i, \mathbf{H}, \mathbf{H}_i^*). \tag{17}$$

These two decoding problems (16) and (17) can be solved efficiently by the ICM algorithm in Algorithm 1 where the only difference is to initialize with random $\mathbf{H}$.

To deal with noisy training image data, we relax the margin constraints by introducing slack variables ξ_i. Thus, we formulate the MMBM learning with the following max-margin objective function,



Case 1: $\Delta(\cdot)=0$

Case 2: $\Delta(\cdot)=\Delta(\mathbf{y},\mathbf{y}_i)$

Case 3: $\Delta(\cdot)=\Delta(\mathbf{H},\mathbf{H}_i^*)$

Case 4: $\Delta(\cdot)=\Delta(\mathbf{y},\mathbf{y}_i)+\Delta(\mathbf{H},\mathbf{H}_i^*)$

Figure 2. Comparing margin functions. Cases 1-4 illustrate learning single-layer MMBMs with four kinds of margin functions. The particular margin functions induce the red connections between any two layers dominate the energy loss during learning while leaving the green connections unoptimized. Best viewed in color.

$$\min_{\omega}\frac{\gamma}{2}\|\omega\|^2+\sum_i \xi_i, \qquad \text{s.t.}$$
$$-E(\mathbf{y}_i,\mathbf{H}_i^*,\mathbf{x}_i;\omega) \geq \max_{\mathbf{y},\mathbf{H}}[\Delta(\mathbf{y}_i,\mathbf{y},\mathbf{H}_i^*,\mathbf{H})-E(\mathbf{y},\mathbf{H},\mathbf{x}_i;\omega)]$$
$$-\xi_i, \xi_i \geq 0, \forall i,$$
$$\text{where} \quad \mathbf{H}_i^* = \arg\max_{\mathbf{H}} -E(\mathbf{y}_i,\mathbf{H},\mathbf{x}_i;\omega). \tag{18}$$

This formulation is equivalent to minimizing the loss function,

$$\min_{\omega}\frac{\gamma}{2}\|\omega\|^2+\sum_i \max_{\mathbf{y},\mathbf{H}}[\Delta(\mathbf{y}_i,\mathbf{y},\mathbf{H}_i^*,\mathbf{H})-E(\mathbf{y},\mathbf{H},\mathbf{x}_i;\omega)+ E(\mathbf{y}_i,\mathbf{H}_i^*,\mathbf{x}_i;\omega)]. \tag{19}$$

To optimize the loss function (19), we initialize the parameters ω_0 with pre-trained $\mathbf{W}^{1,2},\mathbf{c}^{1,2},\mathbf{b}$ and random matrices $\mathbf{V}^{1,2}$. We develop a stochastic gradient descent algorithm (See Algorithm 2) for optimizing (19) by applying the Concave-Convex Procedure [28]. Note that it is easy to compute the gradients of energy functions with respect to ω as both data energy $E(\hat{y}_i,\hat{\mathbf{H}}_i,\mathbf{x}_i;\omega)$ and model energy $E(\mathbf{y}_i,\mathbf{H}_i^*,\mathbf{x}_i;\omega)$ are linear functions of parameters ω given fixed hidden and output variables.

Comparing Margin Functions. Choosing a proper margin penalty function $\Delta(\cdot)$ is crucial to effective learning. Taking the single layer MMBM as an example, we find its energy function consists of three components: hidden-visible interaction (H-V), hidden-image interaction (H-I) and visible-image interaction (V-I), which correspond to the three kinds of edges in the graphical model of MMBMs,

$$E(\mathbf{y},\mathbf{h},\mathbf{x}) = -\underbrace{\mathbf{y}^\top\mathbf{W}\mathbf{h}}_{\text{H-V}}-\underbrace{\mathbf{h}^\top(\mathbf{V}^1\mathbf{x}^1+\mathbf{c})}_{\text{H-I}}-\underbrace{\mathbf{y}^\top(\mathbf{V}^0\mathbf{x}^0+\mathbf{b})}_{\text{V-I}}, \tag{20}$$

We analyze four cases of $\Delta(\cdot)$ (See Figure 2) and evaluate their performance in the experiments.

Case 1: $\Delta(\cdot)=0$**.** If we set $\Delta(\cdot)=0$, then the loss function in (19) reduces to the perceptron loss used in [18]. As the data-dependent and model-dependent energies remain the same form, there exist several possibilities that can explain the perceptron loss, considering the potential combinations of three components. For example, learning with $\Delta = 0$ may end up with a strong H-V component but weak H-I and V-I components, as the H-V component is pre-trained. This result is clearly deficient for prediction.

Case 2: $\Delta(\cdot)=\Delta(\mathbf{y},\mathbf{y}_i)$**.** If we set $\Delta(\cdot)=\Delta(\mathbf{y},\mathbf{y}_i)$, then the loss function in (19) is closely related to the one used in latent Structured SVM [28]. The energy margin $\Delta(\mathbf{y},\mathbf{y}_i)$ only depends on $\mathbf{y}$ so that the V-I component will be better constrained to dominate the energy loss between the data energy and the augmented model energy. Considering the pre-trained H-V component, we may obtain strong H-V and V-I components but a weak H-I component. However, the H-I and V-I components take different input features and should be complementary to each other. The unoptimized H-I component very likely constrains the model generalizability to unseen data.

Case 3: $\Delta(\cdot)=\Delta(\mathbf{H},\mathbf{H}_i^*)$**.** If we set $\Delta(\cdot)=\Delta(\mathbf{H},\mathbf{H}_i^*)$, then the loss function in (19) indirectly corresponds to the output through hidden variables. That is, the H-V component functions as clustering. The margin on hidden variables essentially encourages the H-I component to correctly predict the cluster labels $\mathbf{H}_i^*$, i.e., the hidden variables that best explain the training instance $(\mathbf{x}_i,\mathbf{y}_i)$. Thus, the energy difference is likely dominated by the H-I and H-V components, which leaves the V-I component unoptimized. This approach has the same generalizability problem as Case 2.

Case 4: $\Delta(\cdot)=\Delta(\mathbf{y},\mathbf{y}_i)+\Delta(\mathbf{H},\mathbf{H}_i^*)$**.** Based on the above analysis, we use $\Delta(\cdot)=\Delta(\mathbf{y},\mathbf{y}_i)+\Delta(\mathbf{H},\mathbf{H}_i^*)$ as the margin penalty function. Since $\Delta(\mathbf{y},\mathbf{y}_i)$ and $\Delta(\mathbf{H},\mathbf{H}_i^*)$ are absorbed into the V-I component and H-I component, respectively, all three components are optimized during learning.

5. Experiments

5.1. Datasets

Penn-Fudan Pedestrians This dataset [26] consists of 170 images with bounding box annotations and ground truth foreground-background segmentation masks. The images all include one or more pedestrians. For our experiments we extracted 423 patches, each adjusted to include one person only. We resize the patches to an uniform size of 32×64 pixels, cropping the original image so that we can keep the original aspect ratio while resizing them.

In order to increase the number of training and test samples, we subsequently mirror all patches, resulting in 846 samples, some of which with severe occlusions. We then select 400 samples for training and use the rest for tests. The training-test split is done randomly except for keeping original images in the same set as their mirrored pairs.

Weizmann Horses. This dataset [5] contains 328 horse images, with a high variability of poses and scales. Before processing, we resize every image to 128x128, padding images with different aspect ratios with mirrored versions of the image itself. To get comparable results to [16], we calculate 32x32 foreground-background segmentation masks with all of our models. Also, we use their training-test split (into 200 training and 128 test images).

Caltech-UCSD Birds 200. The dataset [27] includes 6033 images of 200 bird species, each image usually including one dominant bird in the scene. The images are annotated with a bounding box and a coarse-grained segmentation mask. As the accuracy of this isn't sufficient to evaluate our segmentation methods, we manually annotate these images with accurate masks (available on the website `https://eng.ucmerced.edu/people/jyang44`). We crop 6033 bird patches and the corresponding segmentation masks from bounding boxes, and resize the image patches to 128×128 pixels. We use the same training/test partition as in [27], i.e., 3000 samples for training and the rest for tests.

5.2. Implementations

Architectures. For the MMBM with a single hidden layer (MMBM1) and RBM, we use 500 hidden units $\mathbf{h} \in \{0,1\}^{500}$. For the MMBM with two hidden layers (MMBM2), we use 500 hidden units in the first layer $\mathbf{h}^1 \in \{0,1\}^{500}$, and 200 hidden units in the second layer $\mathbf{h}^2 \in \{0,1\}^{200}$. For the birds and the horses, each mask is partitioned into 2×2 four patches $\{\mathbf{y} = \vee \mathbf{y}_k, k = 1,\dots,4, \mathbf{y}_k \in \{0,1\}^{36\times 36}\}$ with 8 pixels overlapping between adjacent patches, such that each part is connected to 125 hidden units in the first layer $\mathbf{h}^1$. For the pedestrians, we also use four patches $\{\mathbf{y} = \vee \mathbf{y}_k, k = 1,\dots,4, \mathbf{y}_k \in \{0,1\}^{22\times 32}\}$ but in a 4x1, vertical organization with 14 pixel overlaps between neighbors.

Features. One of the advantages of the proposed method is that it can handle a diverse set of features: local descriptors can be connected to the visible layer while features covering larger image areas are better suited as conditionals for one of the hidden layers.

For MMBM1, we use two sets of features: $\mathbf{x}^0$ for the visible and $\mathbf{x}^1$ for the hidden layer. For $\mathbf{x}^0$, we first segment the image into superpixels using the gPb algorithm [1]. For each superpixel, we compute dense SIFT, color and contour histograms. The histograms of densely sampled SIFT words are computed by using a codebook of size 512 and the locality-constrained linear coding method [25]. The color histograms of RGB values are computed from a codebook of size 128, and finally, the contour histograms are computed from the oriented gPb edge detector responses [1]. For per-pixel visible features we simply use those of the superpixel containing the pixel in question.

For the hidden layers of MMBM1, we use the HOG descriptors for the entire input image as $\mathbf{x}^1$. For MMBM2, the features $\mathbf{x}^2$ for the top layer is calculated the same way, while for the middle layer feature vector $\mathbf{x}^1$ we use the HOG descriptors for the four patches.

Training. For the MMBM1, we run 2000 epochs with 100-sample mini-batches in the generative training phase (RBM training). For the MMBM2, we run 2000 epochs for the first layer pre-training in the generative training phase (ShapeBM training) and 1000 epochs for the second layer pre-training. In addition, we run 5 cycles in the max-margin training phase in both cases. We set the learning rate $\alpha_0 = 0.001$ and the constant $\gamma = 0.01$ for all the experiments. The MATLAB source code and the labeled datasets will be made available for research purposes.

Baseline. We study two discriminative models for comparison: a superpixel based CRF model using bottom-level features $\mathbf{x}^0$ and Compositional High Order Potentials (CHOPPs) [16]. For the CRF model $p(\mathbf{y}|\mathbf{x}^0)$, we use the implementation in [16].

For CHOPPs, we used the code provided by the authors for the inference but we didn't get the same results, likely due to differences in our unary / pairwise potential generation code. To make the comparison fair, in the experiments we used the same unary features as in our MMBM implementation instead. As Table 2 shows, this improved their results compared to the original published in [16].

Since, unlike the combined RBM-CRF models of [16] and [12], our model doesn't have pairwise weights in the visible layer. For a better comparison with these models, we also ran Graph Cut on the output mask, using the probabilities given by the model as unary potentials and a pairwise term taken from [6], based on the magnitude of the gradients of color channels. We report results for both the original and refined masks.

5.3. Results

We use two metrics for performance evaluation: the average pixel accuracy (AP) of foreground and background classification and the foreground intersection-over-union score (IoU) of entire test set [1]. We first present segmentation results on the Penn-Fudan Pedestrians in Table 1. Overall, the MMBM1 (76.92% IoU, Case 4) and MMBM2 (77.30% IoU, Case 4) outperform the CRF (68.35% IoU) and CHOPPs (71.33% IoU) algorithms. The results show that the MMBMs are effective models for object segmentation by integrating image features and a strong shape prior. Also, as the last two rows of the table indicate, introducing pairwise constraints further improves results.

Our results for Weizmann horses are shown in Table 2. Again, both the advantage of augmenting the loss function with multiple margins and the benefits of using a two-layered architecture are demonstrated. Also, compar-

[1] The IoU score is defined as $\frac{|\mathbf{Y} \bigcap \hat{\mathbf{Y}}|}{|\mathbf{Y} \bigcup \hat{\mathbf{Y}}|}$, where $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ are the sets of ground truth and predicted foreground pixels.

Table 1. Results on the Penn-Fudan Pedestrians dataset.

		AP	IoU
CRF		84.87	68.35
CHOPPs [16]		86.55	71.33
MMBM1	Case 1	82.66	64.80
	Case 2	85.27	69.20
	Case 3	83.35	65.78
	Case 4	89.91	76.92
MMBM2		89.74	77.30
MMBM1 Case 4 w/ GC		90.42	77.97
MMBM2 Case 4 w/ GC		**90.77**	**79.42**

Table 2. Results on the Weizmann Horses dataset.

		AP	IoU
CRF		87.46	67.44
Bo and Fowlkes [4]		77.2	N/A
CHOPPs [16]		88.67	71.60 (69.90 in [16])
MMBM1	Case 1	70.59	38.01
	Case 2	85.87	62.97
	Case 3	85.37	59.35
	Case 4	89.43	69.59
MMBM2		89.80	72.09
MMBM1 Case 4 w/ GC		90.62	74.12
MMBM2 Case 4 w/ GC		**90.71**	**75.78**

isons using different margin functions (Cases 1-4) for the MMBM1 model demonstrate the importance of a max-margin formulation with multiple margins for output prediction. By using margin functions (MMBM1 Cases 2-4), we obtain 19% AP improvement and more than 30% IoU improvement over the non-margin (perceptron loss) algorithm in Case 1 of the MMBM1. The best results for the MMBM1 (89.43% AP, 69.59% IoU) from Case 4 indicate that the combining multiple margin functions $\Delta(\cdot) = \Delta(\mathbf{y}, \mathbf{y}_i) + \Delta(\mathbf{H}, \mathbf{H}_i^*)$ alleviates degenerating effects by providing stronger constraints. Our results on other datasets also strengthen this observation. The two-layer hierarchical hidden architecture also helps generating better results than a single hidden layer, as shown in the Case 4 of MMBM2 (89.80%AP, 72.09% IoU) over MMBM1.

In addition to the comparison to CRF and CHOPPs, for this dataset we also added the results from [4]. Their aim was to identify body parts and got the foreground-background segmentation as a byproduct.

Finally, the segmentation results on the Caltech-UCSD Birds 200 dataset are presented in Table 3. Different from pedestrians and horses, this dataset has large shape variations but more distinct appearances (e.g., color, textures). Thus, the appearance-based CRF model performs less well. Similar is the case of CHOPP: as its hidden nodes are not directly connected to image features, so they can only refine and correct the shape of results that are mostly right just based on local, visible-layer features, which is hard to accomplish on this dataset. In contrast, the features-to-hidden connections in the MMBM models make it possible to exploit global shape information even without reliable local features. The results, similar to the observations on the other two datasets for evaluating different margin functions demonstrate the significance of max-margin formulation and combining margin functions (Case 4). In the bird data, we observe better performance by using just one hidden layer compared to using the two-layered MMBM2 model. A possible reason is that while the weight replication for the four windows in MMBM2 is beneficial when given a small number of training samples (such as for horses and pedestrians), but for larger datasets we can learn a better prior using simple architectures (RBMs) with more parameters from the data.

Table 3. Results on the Caltech-UCSD Birds 200 dataset.

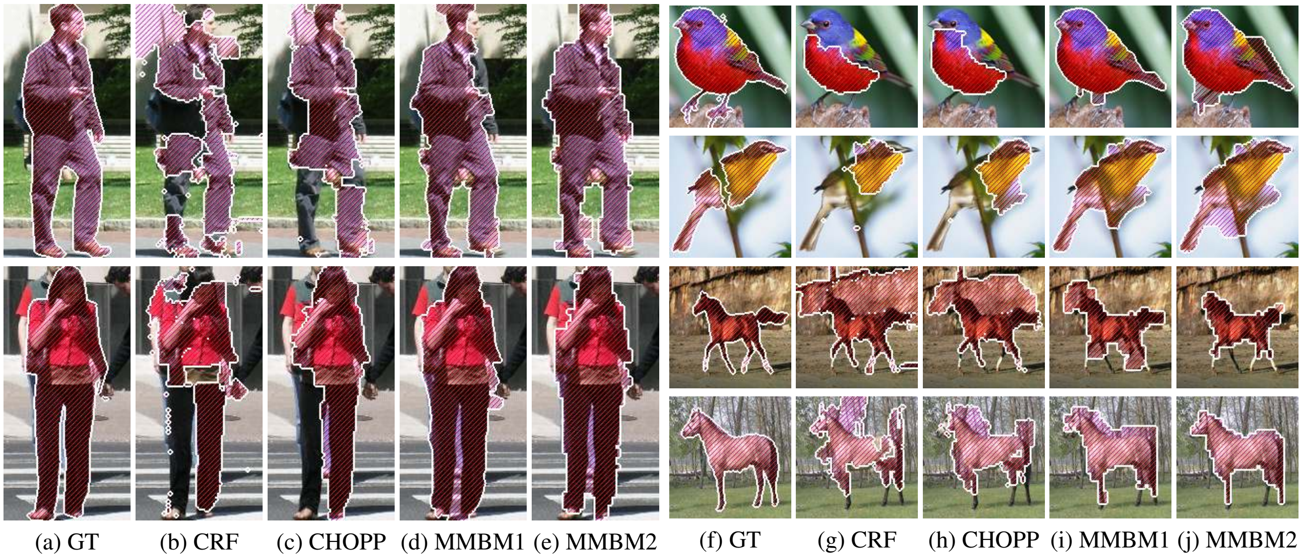
		AP	IoU
CRF		83.50	38.45
CHOPPs [16]		74.52	48.84
MMBM1	Case 1	80.96	60.37
	Case 2	87.73	72.45
	Case 3	75.73	63.22
	Case 4	88.07	72.96
MMBM2		86.38	69.87
MMBM1 Case 4 w/ GC		**90.42**	**75.92**
MMBM2 Case 4 w/ GC		90.77	72.40

We present some qualitative results in Figure 3, from which we can see more directly the importance of features-to-hidden connections for shape prediction. For example, CRF finds the most colorful parts of birds, which is corrected by CHOPP to be shaped more birdlike, but it's only MMBMs that discover the entire bird well.

6. Conclusions

In this paper, we propose MMBMs for structured output prediction problems and investigate two variants of MMBMs with single and two hidden layers for object segmentation. Instead of using BMs as shape priors, we build connections between input observations with hidden variables that opens an inference pathway from image features to object shapes. We derive a simple yet efficient ICM algorithm for MAP inference. We formulate MMBMs with a max-margin objective function for discriminative training, and discuss four margin functions as well as their effects on learning performance. The results on horses, pedestrians, birds datasets show that our algorithms perform favorably against the state-of-the-art methods.

In experiments, we have found that the pairwise edge potentials can after all improve the segmentation quality, given the predicted shapes from our models. In the future, we plan to extend MMBM models by adding pairwise potentials to the visible layer. Considering the alternating procedure of the MAP inference algorithm, this extension will not sig-



(a) GT (b) CRF (c) CHOPP (d) MMBM1 (e) MMBM2 (f) GT (g) CRF (h) CHOPP (i) MMBM1 (j) MMBM2

Figure 3. Qualitative results on the Penn-Fudan Pedestrians, Caltech-UCSD Birds and Weizmann horses where segmentation results (shown with white contours) are overlaid with the input images.

nificantly increase the complexity of inference and learning because we only need to replace Line 4 in Algorithm 1 with Graph Cut. We are also interested in integrating object detection with segmentation in MMBM models.

Acknowledgements

The work is supported partly by NSF CAREER Grant #1149783 and NSF IIS Grant #1152576. We thank Yujia Li for providing us the code to run comparative experiments, and the CVPR reviewers for their feedback on this work.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011.

[2] L. Bertelli, T. Yu, D. Vu, and B. Gokturk. Kernelized structural svm learning for supervised object segmentation. In *CVPR*, 2011.

[3] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society: Series B*, 48(3):259–302, 1986.

[4] Y. Bo and C. Fowlkes. Shape-based pedestrian parsing. In *CVPR*, 2011.

[5] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *ECCV*, 2002.

[6] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *ICCV*, 2001.

[7] F. Chen, H. Yu, R. Hu, and X. Zeng. Deep learning shape priors for object segmentation. In *CVPR*, 2013.

[8] S. M. A. Eslami, N. Heess, and J. Winn. The shape Boltzmann machine: a strong model of object shape. In *CVPR*, 2012.

[9] S. M. A. Eslami and C. K. I. Williams. A generative model for parts-based object segmentation. In *NIPS*, 2012.

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

[11] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

[12] A. Kae, K. Sohn, H. Lee, and E. Learned-Miller. Augmenting CRFs with Boltzmann Machine shape priors for image labeling. In *CVPR*, 2013.

[13] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

[14] M. P. Kumar, P. Torr, and A. Zisserman. Obj cut. In *CVPR*, 2005.

[15] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[16] Y. Li, D. Tarlow, and R. Zemel. Exploring compositional high order pattern potentials for structured output learning. In *CVPR*, 2013.

[17] X. Miao and R. P. N. Rao. Large margin boltzmann machines. In *IJCAI*, 2009.

[18] V. Mnih, H. Larochelle, and G. E. Hinton. Conditional restricted Boltzmann machines for structured output prediction. In *UAI*, 2011.

[19] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *CVPR*, 2009.

[20] R. Salakhutdinov and G. Hinton. Deep Boltzmann machines. In *AISTATS*, 2009.

[21] A. Shekhovtsov, P. Kohli, and C. Rother. Curvature prior for MRF-based segmentation and shape inpainting. In *DAGM*, 2012.

[22] M. Szummer, P. Kohli, and D. Hoiem. Learning CRFs using graph cuts. In *ECCV*, 2008.

[23] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *NIPS*, 2003.

[24] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453 – 1484, 2005.

[25] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[26] L. Wang, J. Shi, G. Song, and I.-F. Shen. Object detection combining recognition and segmentation. In *ACCV*, 2007.

[27] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical report, California Institute of Technology, 2010.

[28] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*, 2009.