Exploring Compositional High Order Pattern Potentials for Structured Output Learning

by

Yujia Li

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

Exploring Compositional High Order Pattern Potentials for Structured Output Learning

Yujia Li
Master of Science
Graduate Department of Computer Science
University of Toronto
2013

When modeling structured outputs like image segmentations, predictions can be improved by accurately modeling structure present in the labels. A key challenge is developing tractable models that are able to capture complex high level structure like shape. In this work, we study the learning of a general class of pattern-like high order potential, which we call Compositional High Order Pattern Potentials (CHOPPs). We show that CHOPPs include the linear deviation pattern potentials of Rother et al. [27] and also Restricted Boltzmann Machines (RBMs); we also establish the near equivalence of these two models.

Experimentally, we show that performance is affected significantly by the degree of variability present in the data sets, and we define a quantitative variability measure to aid in studying this. We then improve CHOPPs performance in high variability data sets with two primary contributions: (a) developing a loss-sensitive joint learning procedure, so that internal pattern parameters can be learned in conjunction with other model potentials to minimize expected loss, and (b) learning an image-dependent mapping that encourages or inhibits patterns depending on image features. We also explore varying how multiple patterns are composed and learning convolutional patterns, which have smaller receptive fields and shared parameters, that are densely tiled over the image. Quantitative results on challenging highly variable data sets show that the joint learning and image-dependent high order potentials can improve performance.

Chapter 1

Introduction

Many tasks in computer vision can be framed as making predictions about complex, structured objects. For example, image labeling problems like stereo depth estimation, optical flow, and image segmentation can all be cast as making predictions jointly over many correlated outputs. The modeling frameworks that have found the most success for this type of problems are those like Conditional Random Fields (CRFs) and Structural Support Vector Machines (SSVMs), which explicitly model the correlations over the outputs and make test-time predictions by either exactly or approximately solving a joint inference task. These formulations are collectively known as structured output learning, or structured prediction, and are the focus of this work.

A key research issue that arises when working with structured output problems is how to best trade off expressivity of the model with the ability to efficiently learn and perform inference (make predictions). Traditionally, these concerns have led to the use of overly simplistic models over labelings that make unrealistic conditional independence assumptions, such as pairwise models with grid-structured topology. Recently, there have been successful efforts that weaken these assumptions, either by moving to densely connected pairwise models [12] or by enforcing smoothness in higher order neighborhoods [9]. However, while these approaches can lead to improved performance, they do not capture much higher level structure in the data, such as information about shape. As we look to build models that more faithfully represent structure present in the world, it is desirable to explore the use of models capable of representing this higher level structure.

One promising direction towards incorporating these goals in the structured output setting appears to be the *pattern potentials* of Rother et al. [27] and Komodakis & Paragios [11], which are capable of modeling soft template structures and can dramatically outperform pairwise models in highly structured settings that arise, for example, when modeling regular textures. Yet despite the clearly powerful representational ability of pattern potentials, they have not found much success in more realistic settings, like those found in the PASCAL VOC image labeling task [3].

A model that is appropriate in similar situtations and has also found success modeling textures [8] is the Restricted Boltzmann Machine (RBM). In fact, our starting observation in this work is that the similarity is not superficial—mathematically, RBM models are nearly identical to the pattern potentials of [27]. We will make this claim precise in Chapter 3, leading to the definition of a more general class of high order potential that includes both pattern potentials and RBMs. We call this class *Compositional High Order Pattern Potentials* (CHOPPs). A primary benefit of this observation is that there is a well-developed literature on learning RBM models that becomes available for learning pattern-like potentials.

In this work we explore augmenting standard CRF models with CHOPPs. Our goal is to not only learn a tradeoff parameter between the standard and high order parts of the model, but also to learn internal pattern parameters. We then focus on the question of how effective these potentials are as the variability and complexity of the image segmentation task increases. We propose a simple method for assessing the degree of variation in the labels, then show that the performance of a vanilla application of CHOPPs degrades relative to the performance of standard pairwise potentials as this measure of variability increases.

We then turn attention to improving vanilla CHOPP-augmented CRFs, and make two primary suggestions. The first is to incorporate additional parameters in the RBM-based potential that allows the pattern activities to depend on information in the image. This is analogous to allowing standard pairwise potentials to vary depending on local image color differences [1] or more advanced boundary detectors like Pb [20]. The second is to utilize a loss function during training that is tailored to the metric used for evaluating the labeling results at test time. Our results indicate that jointly training the CHOPP potentials with the rest of the model improves performance, and training specifically for the evaluation criterion used at test time (we use an intersection-over-union measure throughout) improves over a maximum likelihood-based objective. Finally, we explore (a) different forms of compositionality: the 'min' version advocated by Rother et al. [27], which is essentially a mixture model, versus the 'sum' version, which is more compositional in nature; and (b) convolutional applications of the high order potentials versus their global application.

Since this work sits at the interface of structured output learning and RBM learning, we conclude by suggesting take-aways for both the RBM-oriented researcher and the structured output-oriented researcher, proposing what each approach has to offer the other and outlining possible directions for improving the applicability of pattern-based approaches to challenging structured output problems.

Note Much of this work was done jointly with Daniel Tarlow and Richard Zemel, and has been submitted as a conference paper to CVPR.

Chapter 2

Background & Related Work

2.1 Structured Output Learning

In structured output learning, the goal is to predict a vector of labels $\mathbf{y} \in \mathcal{Y} = \{0, \ldots, C-1\}^{D_v}$ given inputs $\mathbf{x} \in \mathcal{X}$. A standard approach, which is taken by e.g. structural SVMs, is to define an input-to-output mapping function $g_{\boldsymbol{\lambda}} : \mathcal{X} \to \mathcal{Y}$ that is governed by parameters $\boldsymbol{\lambda}$. Given feature functions $f(\mathbf{x}, \mathbf{y})$, this mapping is constructed implicitly via the maximization of a scoring function, which can be interpreted as a conditional probability distribution $p(\mathbf{y} \,|\, \mathbf{x}; \boldsymbol{\lambda})$: $g_{\boldsymbol{\lambda}}(\mathbf{x}) = \operatorname{argmax}_{\mathbf{y}} p(\mathbf{y} \,|\, \mathbf{x}; \boldsymbol{\lambda})$, where $p(\mathbf{y} \,|\, \mathbf{x}; \boldsymbol{\lambda}) \propto \exp\left\{\sum_{j=1}^{J} \lambda_j f_j(\mathbf{y}, \mathbf{x})\right\}$. In practice, we must restrict the form of $f_j(\cdot)$ functions in order to ensure tractability, typically by forcing the function's value to depend on the setting of only a small number of dimensions of $\mathbf{y}$. Also, some $f_j(\cdot)$ functions may ignore $\mathbf{x}$, which has the effect of adding input-independent prior constraints over the label space. The result is a log-linear probability distribution ($\log p(\mathbf{y} \,|\, \mathbf{x}; \boldsymbol{\lambda})$ is linear in $\boldsymbol{\lambda}$), which may be optimized using a variety of methods [33].

Latent Variable Models To increase the representational power of a model, a common approach is to introduce latent (or hidden) variables $\mathbf{h} \in \mathcal{H} = \{0, \ldots, H-1\}^J$. The above formulation can then be easily extended by defining feature functions $f(\mathbf{x}, \mathbf{y}, \mathbf{h})$ that may include latent variables, which leads to a probability distribution $p(\mathbf{y}, \mathbf{h} \,|\, \mathbf{x})$. To make predictions, it is common to either maximize out or sum out the latent variables:

$$g_{\boldsymbol{\lambda}}(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y}} \max_{\mathbf{h} \in \mathcal{H}} p(\mathbf{y}, \mathbf{h} \,|\, \mathbf{x}; \boldsymbol{\lambda}), \text{ or} \tag{2.1}$$

$$g_{\boldsymbol{\lambda}}(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y}} \sum_{\mathbf{h} \in \mathcal{H}} p(\mathbf{y}, \mathbf{h} \,|\, \mathbf{x}; \boldsymbol{\lambda}). \tag{2.2}$$

The former strategy is employed by latent structural SVMs [37], while the latter is employed by hidden CRF models [25]. A topic of ongoing investigation is the benefits of each, and alternative strategies that interpolate between the two [21].

High Order Potentials A related strategy for increasing the representational power of a model is to allow feature functions to depend on a large number of dimensions of $\mathbf{y}$. These types of interactions are known collectively as *high order potentials* and have received considerable attention in recent years. They have been used for several purposes, including modeling higher order smoothness [9], co-occurrences of labels in semantic image segmentation [13], and cardinality-based potentials [35, 36]. While the above examples provide interesting non-local constraints, they do not encode shape-based information appropriate for image labeling applications. There are other high order models that come closer to this goal, modeling star convexity [5], connectivity [34, 24], and a bounding box occupancy constraint [17]. However, these still are quite restrictive notions of shape compared to what pattern-based models are capable of representing.

Learning High Order Potentials In addition to a weighting coefficient that governs the relative contribution of each feature function to the overall scoring function, the features also have internal parameters. This is the case in CHOPPs, where internal parameters dictate the target pattern and the costs for deviating from it. These parameters also need to be set, and the approach we take in this work is to learn them. We emphasize the distinction between first learning the internal parameters offline and then learning (or fixing by hand) the trade-off parameters that controls the relative strength of the high order terms, versus the joint learning of both types of parameters. While there is much work that takes the former approach [10, 27, 13, 14], there is little work on the latter in the context of high order potentials. Indeed it is more challenging, as standard learning formulations become less appropriate (e.g., using a variant on standard SSVM learning for CHOPPs leads to a degeneracy where all patterns become equivalent), and objectives are generally non-convex.

2.2 Restricted Boltzmann Machines

A Restricted Boltzmann Machine (RBM) [29] is a form of undirected graphical model that uses hidden variables to model high-order regularities in data. It consists of the I *visible* units $\mathbf{v} = (x_1, \dots, x_I)^\top$ that represent the observations, or data; and the J *hidden* or latent units $\mathbf{h} = (h_1, \dots, h_J)^\top$ that mediate dependencies between the visible units. The system can be seen as a bipartite graph, with the visibles and the hiddens forming two layers of vertices in the graph; the restriction is that no connection exists between units in the same layer.

The aim of the RBM is to represent probability distributions over the states of the random variables. The pattern of interaction is specified through the energy function:

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{v}^\top \mathbf{W} \mathbf{h} - \mathbf{b}^\top \mathbf{v} - \mathbf{c}^\top \mathbf{h} \tag{2.3}$$

where $\mathbf{W} \in \mathbb{R}^{I \times J}$ encodes the hidden-visible interactions, $\mathbf{b} = (b_1, \dots, b_I)^\top$ the input and $\mathbf{c} \in \mathbb{R}^J$ the hidden self-connections (also known as biases). The energy function specifies the probability distribution over the joint space $[\mathbf{v}, \mathbf{h}]$ via the Boltzmann distribution

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z_\theta} \exp(-E_\theta(\mathbf{v}, \mathbf{h})) \tag{2.4}$$

with the partition function Z_θ given by $\sum_{\mathbf{v},\mathbf{h}} \exp(-E_\theta(\mathbf{v}, \mathbf{h}))$. Based on this definition, the probability for any subset of variables can be obtained by conditioning and marginalization.

Learning in RBMs For maximum likelihood learning, the goal is to make the data samples likely, which entails computing the probability for any input $\mathbf{v}$; this can be derived by performing the exponential sum over all possibly hidden vectors $\mathbf{h}$: $p(\mathbf{v}) = \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h})$, effectively marginalizing them out. For an RBM with I binary visible units, this takes on a particular nice form:

$$\begin{aligned} p(\mathbf{x}) &= \sum_{\mathbf{h}} \frac{1}{Z_\theta} \exp(\mathbf{v}^\top \mathbf{W}\mathbf{h} + \mathbf{b}^\top \mathbf{v} + \mathbf{c}^\top \mathbf{h}) \\ &= \frac{1}{Z_\theta} \exp\left(\mathbf{b}^\top \mathbf{v} + \sum_j \log\left(1 + \exp(\mathbf{v}^\top \mathbf{w}_j + c_j)\right) \right) \end{aligned} \tag{2.5}$$

where $\mathbf{w}_j$ is the jth column in $\mathbf{W}$ and each of the terms inside the summation over j is known as a *softplus*. The standard approach to learning in RBMs uses an approximation to maximum likelihood learning known as Contrastive Divergence (CD) [7].

Vision Applications There have been numerous applications of RBM to vision problems. RBMs are typically trained to model the input data such as an image, and most vision applications have focused on this standard unsupervised training paradigm. For example, they have been used to model object shape [2], images under occlusion [18], and noisy images [32]. They have also been applied in a discriminative setting, as joint models of inputs and a class [15].

The focus of the RBMs we explore here, as models of image labels, has received relatively little attention. Note that in this case the visible units of the RBM now correspond to the image labels $\mathbf{y}$. The closest work to our's is that of [6]. The RBMs in that work only captured very local or global patterns in the label field, and did not address shape information as we do, and it also combined the RBM with a very restricted form of CRF, which greatly simplified inference and learning. [22] also tried to use RBMs for structured output problems, but the model used in the paper did not have pairwise connections in the label field, and the actual loss was not considered during training.

Chapter 3

Equating Pattern Potentials and RBMs

This chapter provides the detailed proof of the equivalence between pattern potentials and RBMs. The high level idea of the proof is to treat each hidden variable in an RBM as encoding a pattern.

We first introduce the definition of pattern potentials by Rother et al. in [27], a few necessary change of variable tricks, and two different ways to compose more general high order potentials, "sum" and "min".

Then we relates the composite pattern potentials to RBMs. We show in Section 3.2 that *minimizing out* hidden variables in RBMs are equivalent to pattern potentials. When there are no constraints on hidden variables, we recover the "sum" composite pattern potentials; when there is a 1-of-J constraint on hidden variables, we recover the "min" composite pattern potentials. In Section 3.3, we show that *summing out* hidden variables in RBMs approximates pattern potentials, and similarly with and without constraints on hidden variables would lead us to "min" and "sum" cases respectively.

The RBM formulation offers considerable generality via choices about how to constrain hidden unit activations. This allows a smooth interpolation between the "sum" and "min" composition strategies. Also, this formulation allows the application of learning procedures that are appropriate for cases other than just the "min" composition strategy.

In Chapter 4, we provide a way to unify minimizing out hidden variables and summing out hidden variables by introducing a temperature parameter in the model.

Notation In this chapter, we use f for pattern potentials and g for the high order potentials induced by RBMs. Superscripts 's' and 'm' on f corresponds to two composition schemes, sum and min. Superscripts on g correspond to two types of constraints on RBM hidden variables, and subscripts on g correspond to minimizing out or summing out hidden variables.

3.1 Pattern potentials

In [27], a basis pattern potential for a clique of binary variables $\mathbf{y}_a$ is defined as

$$f(\mathbf{y}_a) = \min\{d(\mathbf{y}_a) + \theta_0, \theta_{\max}\} \tag{3.1}$$

where $d : \{0,1\}^{|a|} \to \mathbb{R}$ is a deviation function specifying the penalty for deviating from a specific pattern. The pattern potential penalizes configurations of $\mathbf{y}_a$ that deviates from the pattern, and the penalty is upper bounded by $\theta_{\max}$ while θ_0 is a base penalty.

For a specific pattern $\mathbf{Y}$, the deviation function $d(\mathbf{y}_a)$ is defined as[1]

$$d(\mathbf{y}_a) = \sum_{i \in a} abs(w_i)(y_i \neq \mathbf{Y}_i) \tag{3.2}$$

where $abs()$ is the absolute value function. This is essentially a weighted hamming distance of $\mathbf{y}_a$ from $\mathbf{Y}$. Since $\mathbf{y}_a$ and $\mathbf{Y}$ are both binary vectors, we have the following alternative formulation

$$\begin{aligned} d(\mathbf{y}_a) &= \sum_{i \in a : \mathbf{Y}_i = 1} (-w_i)(1 - y_i) + \sum_{i \in a : \mathbf{Y}_i = 0} w_i y_i \\ &= \sum_{i \in a} w_i y_i + \sum_{i \in a : \mathbf{Y}_i = 1} (-w_i) \end{aligned} \tag{3.3}$$

w_i specifies the cost of assigning y_i to be 1. $w_i > 0$ when $\mathbf{Y}_i = 0$ and $w_i < 0$ when $\mathbf{Y}_i = 1$.

We can subtract constant $\theta_{\max}$ from Eq. 3.1 to get

$$f(\mathbf{y}_a) = \min \left\{ \sum_{i \in a} w_i y_i + \sum_{i \in a : \mathbf{Y}_i = 1} (-w_i) - \theta, 0 \right\} \tag{3.4}$$

Making the change of variables $w_i' = -w_i$, $c = \theta + \sum_{i \in a : \mathbf{Y}_i = 1} w_i$, we can rewrite the above equation as

$$f(\mathbf{y}_a) = \min \left\{ -c - \sum_{i \in a} w_i' y_i, 0 \right\} \tag{3.5}$$

This formulation is useful for establishing connections with RBMs as shown later in this section.

[27] proposed two ways to compose more general high order potentials from basis pattern potentials defined above. One is to take the sum of different pattern potentials

$$\begin{aligned} f^s(\mathbf{y}_a) &= \sum_{j=1}^{J} \min\{d_j(\mathbf{y}_a) + \theta_j, \theta_{\max}\} \\ &= \sum_{j=1}^{J} \min\{d_j(\mathbf{y}_a) + \theta_j', 0\} + \text{const} \end{aligned} \tag{3.6}$$

and the other is to take the minimum of them, to get

$$f^m(\mathbf{y}_a) = \min_{1 \le j \le J} \{d_j(\mathbf{y}_a) + \theta_j\} \tag{3.7}$$

In both cases, $d_j(.)$'s are J different deviation functions, and θ_j's are base penalties for different patterns. In the "min" case, we can also fix one deviation function to be 0 (e.g. by setting all weights $w_i = 0$), to get a constant threshold.

Using the change of variable tricks introduced above, we can rewrite the "sum" composite pattern

[1] Note that in [27], there is also a factor θ in this definition ($d(\mathbf{y}_a)$ is given by the product of factor θ and the sum), but actually the θ factor can always be absorbed in w_i's to get this equivalent formulation.

potential as

$$f^s(\mathbf{y}_a) = \sum_{j=1}^{J} \min\left\{-c_j - \sum_{i \in a} w_{ij} y_i, 0\right\} \tag{3.8}$$

where we ignored the constant term, and rewrite the "min" composite pattern potential as

$$f^m(\mathbf{y}_a) = \min_{1 \le j \le J} \left\{-c_j - \sum_{i \in a} w_{ij} y_i\right\} \tag{3.9}$$

3.2 Minimizing out hidden variables in RBMs

We start from minimizing hidden variables out. The probability distribution defined by a binary RBM is given by

$$p(\mathbf{y}, \mathbf{h}) = \frac{1}{Z} \exp\left(-E(\mathbf{y}, \mathbf{h})\right) \tag{3.10}$$

where the energy

$$E(\mathbf{y}, \mathbf{h}) = -\sum_{i=1}^{I}\sum_{j=1}^{J} w_{ij} y_i h_j - \sum_{i=1}^{I} b_i y_i - \sum_{j=1}^{J} c_j h_j \tag{3.11}$$

Minimizing out the hidden variables, the equivalent high order potential is

$$g_{\min}(\mathbf{y}) = \min_{\mathbf{h}} \left\{-\sum_{j=1}^{J}\left(c_j + \sum_{i=1}^{I} w_{ij} y_i\right) h_j\right\} \tag{3.12}$$

When there is no constraint on hidden variables, i.e. they are independent binary variables, the minimization can be factorized and moved inside the sum

$$g^{\text{uc}}_{\min}(\mathbf{y}) = \sum_{j=1}^{J} \min\left\{-c_j - \sum_{i=1}^{I} w_{ij} y_i, 0\right\} \tag{3.13}$$

The superscript "uc" is short for "unconstrained". This is exactly the same as the "sum" composite pattern potentials in Eq. 3.8.

When we put a 1-of-J constraint on hidden variables, i.e. forcing $\sum_{j=1}^{J} h_j = 1$, the minimization becomes

$$g^{1\text{of}J}_{\min}(\mathbf{y}) = \min_{1 \le j \le J} \left\{-c_j - \sum_{i=1}^{I} w_{ij} y_i\right\} \tag{3.14}$$

This is exactly the same as the "min" composite pattern potentials in Eq. 3.9.

3.3 Summing out hidden variables in RBMs

The key observation that relates the pattern potentials and RBMs with hidden variables summed out is the following approximation,

$$\min\{x, 0\} \approx -\log(1 + \exp(-x)) \tag{3.15}$$

It is easy to see that when x is a large positive value, the right hand side will be close to 0 and when x is a large negative value, the right hand side will be linear in x. This is illustrated in Fig 3.1 (a).

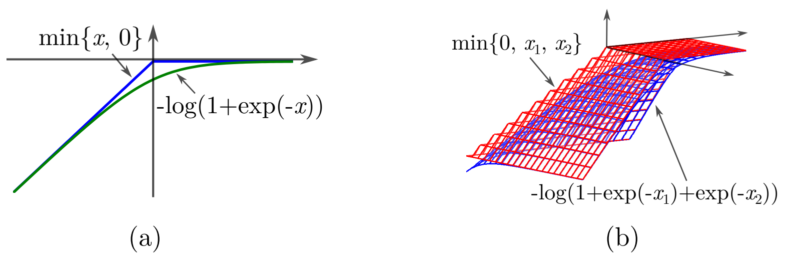


Figure 3.1: (a) $-\log(1+\exp(-x))$ is a smoothed approximation to $\min\{x, 0\}$; (b) $-\log(1+\exp(-x_1)+\exp(-x_2))$ is a smoothed approximation to $\min\{x_1, x_2, 0\}$.

With this approximation, we can rewrite the basis pattern potential in Eq. 3.5 as

$$f(\mathbf{y}_a) \approx -\log\left(1+\exp\left(c+\sum_{i\in a} w'_i y_i\right)\right) \tag{3.16}$$

On the other hand, summing out hidden variables in an RBM with no constraints on hidden variables, the marginal distribution becomes

$$p(\mathbf{y}) = \frac{1}{Z}\exp\left(\sum_{i=1}^{I} b_i y_i\right)\prod_{j=1}^{J}\left(1+\exp\left(c_j+\sum_{i=1}^{I} w_{ij} y_i\right)\right) \tag{3.17}$$

Eq. 2.5 is another equivalent form of this. Therefore the equivalent high order potential induced by summing out the hidden variables is

$$g_{\text{sum}}^{\text{uc}}(\mathbf{y}) = -\sum_{j=1}^{J}\log\left(1+\exp\left(c_j+\sum_{i=1}^{I} w_{ij} y_i\right)\right) \tag{3.18}$$

which is exactly a sum of potentials in the form of Eq. 3.16.

Now we turn to the "min" case. We show that the composite pattern potentials are equivalent to RBMs with a 1-of-J constraint on hidden variables and hidden variables summed out, up to the following approximation

$$\min\{x_1, x_2, ..., x_J, 0\} \approx -\log\left(1+\sum_{j=1}^{J}\exp(-x_j)\right) \tag{3.19}$$

This is a high dimensional extension to Eq. 3.15. The 2-D case is illustrated in Fig 3.1 (b).

We use the definition of "min" composite pattern potentials in Eq. 3.7, but fix $d_J(\mathbf{y}_a)$ to be 0, to make a constant threshold on the cost.

Then we can subtract constant θ_J from the potential and absorb θ_J into all other θ_j's (with the same change of variable tricks) to get

$$f^m(\mathbf{x}_a) = \min\left\{-c_1-\sum_{i\in a} w_{i1} y_i, ..., -c_{J-1}-\sum_{i\in a} w_{i,J-1} y_i, 0\right\} \tag{3.20}$$

Composition Scheme for Pattern Potentials	Operation on RBM		Constraint on $\mathbf{h}$
	Minimizing out $\mathbf{h}$	Summing out $\mathbf{h}$	
Min	$\min_{1\le j\le J}\left\{-c_j-\sum_{i=1}^I w_{ij}y_i\right\}$	$-\log\left(1+\sum_{j=1}^{J-1}\exp\left(c_j+\sum_{i\in a}w_{ij}y_i\right)\right)$	1-of-J
Sum	$\sum_{j=1}^J \min\left\{-c_j-\sum_{i=1}^I w_{ij}y_i, 0\right\}$	$-\sum_{j=1}^J \log\left(1+\exp\left(c_j+\sum_{i=1}^I w_{ij}y_i\right)\right)$	None

Table 3.1: Equivalent compositional high order potentials by applying different operations and constraints on RBMs. Minimizing out hidden variables results in high order potentials that are exactly equivalent to pattern potentials. Summing out hidden variables results in approximations to pattern potentials. 1-of-J constraint on hidden variables corresponds to the "min" compositional scheme. No constraints on hidden variables corresponds to "sum" compositional scheme.

Using the approximation, this high order potential becomes

$$f^m(\mathbf{x}_a) \approx -\log\left(1+\sum_{j=1}^{J-1}\exp\left(c_j+\sum_{i\in a}w_{ij}y_i\right)\right) \tag{3.21}$$

In an RBM with J hidden variables, the 1-of-J constraint is equivalent to $\sum_{j=1}^J h_j = 1$. With this constraint, the energy (Eq. 3.11) can be transformed into

$$\begin{aligned} E(\mathbf{y},\mathbf{h}) &= -\sum_{i=1}^I b_i y_i - \sum_{j=1}^{J-1}\left(c_j - c_J + \sum_{i=1}^I (w_{ij}-w_{iJ})y_i\right)h_j - \left(c_J + \sum_{i=1}^I w_{iJ}y_i\right) \\ &= -\sum_{i=1}^I (b_i - w_{iJ})y_i - \sum_{j=1}^{J-1}\left(c_j - c_J + \sum_{i=1}^I (w_{ij}-w_{iJ})y_i\right) - c_J \end{aligned} \tag{3.22}$$

We can therefore use a new set of parameters $b'_i = b_i - w_{iJ}$, $c'_j = c_j - c_J$ and $w'_{ij} = w_{ij} - w_{iJ}$, and get

$$E(\mathbf{y},\mathbf{h}) = -\sum_{i=1}^I b'_i y_i - \sum_{j=1}^{J-1}\left(c'_j + \sum_{i=1}^I w'_{ij}y_i\right)h_j \tag{3.23}$$

We ignored the constant c_J because it would cancel out when we normalize the distribution. Note that now the set of $J-1$ hidden variables can have at most one on, and they can also be all off, corresponding to the case that the Jth hidden variable is on.

Summing out $\mathbf{h}$, we get

$$p(\mathbf{y}) = \frac{1}{Z}\exp\left(\sum_{i=1}^I b'_i y_i\right)\left(1+\sum_{j=1}^{J-1}\exp\left(c'_j+\sum_{i=1}^I w'_{ij}y_i\right)\right) \tag{3.24}$$

The constant 1 comes from the Jth hidden variable. The equivalent high-order potential for this model is then

$$g_{\text{sum}}^{1\text{of}J}(\mathbf{y}) = -\log\left(1+\sum_{j=1}^{J-1}\exp\left(c'_j+\sum_{i=1}^I w'_{ij}y_i\right)\right) \tag{3.25}$$

which has exactly the same form as Eq. 3.21.

Our results in this section are summarized in Table 3.1.

Chapter 4

The CHOPP-Augmented CRF

Understanding the equivalence between RBMs and pattern potentials leads us to define a more general potential — Compositional High Order Pattern Potential (CHOPP)

$$f_T(\mathbf{y}) = -T \log \left(\sum_{\mathbf{h}} \exp \left(\frac{1}{T} \sum_{j=1}^{J} \left(c_j + \sum_{i=1}^{I} w_{ij} y_i \right) h_j \right) \right) \tag{4.1}$$

where T is a temperature parameter. The sum over $\mathbf{h}$ is a sum over all possible configurations of hidden variables. As did by Schwing et al. in [28], introducing a temperature parameter can smoothly interpolate minimization and summation.

Setting $T = 1$, this CHOPP becomes

$$f_{T=1}(\mathbf{y}) = -\log \left(\sum_{\mathbf{h}} \exp \left(\sum_{j=1}^{J} \left(c_j + \sum_{i=1}^{I} w_{ij} y_i \right) h_j \right) \right) \tag{4.2}$$

this is the equivalent RBM high order potential with hidden variables summed out. When there is no constraint on $\mathbf{h}$, the above potential becomes

$$f_{T=1}^{\text{uc}}(\mathbf{y}) = -\sum_{j=1}^{J} \log \left(1 + \exp \left(c_j + \sum_{i=1}^{I} w_{ij} y_i \right) \right) \tag{4.3}$$

When there is a 1-of-J constraint on $\mathbf{h}$, the above potential is

$$f_{T=1}^{1\text{of}J}(\mathbf{y}) = -\log \left(\sum_{j=1}^{J} \exp \left(c_i + \sum_{i=1}^{I} w_{ij} y_i \right) \right) \tag{4.4}$$

Setting $T \to 0$, the CHOPP becomes

$$f_{T\to 0}(\mathbf{y}) = \min_{\mathbf{h}} \left\{ -\sum_{j=1}^{J} \left(c_j + \sum_{i=1}^{I} w_{ij} y_i \right) h_j \right\} \tag{4.5}$$

this is exactly the same as the high order potential induced by an RBM with hidden variables minimized out, and therefore equivalent to composite pattern potentials as shown in Section 3.2. When there are no constraints on hidden variables we will get the "sum" composite pattern potentials, while adding a 1-of-J constraint will give us the "min" composite pattern potentials.

Therefore, by using a temperature parameter T, CHOPPs can smoothly interpolate summing out hidden variables (usually used in RBMs) and minimizing out hidden variables (used in Rother et al.[27]). On the other hand, by using extreme sparsity (the 1-of-J constraint), it interpolates the "sum" and "min" composition schemes.

Though this uniform formulation is nice, one problem with it is that it is not straight-forward to find a joint distribution of $\mathbf{y}$ and $\mathbf{h}$ that would lead to this high order potential of $\mathbf{y}$ given in Eq. 4.1 by summing out or minimizing out $\mathbf{h}$. This is because we actually introduced the parameter T after summing out $\mathbf{h}$. However, the case $T = 1$, which is equivalent to an RBM, does not have this problem and is still an approximation to the pattern potentials as shown in Chapter 3. In the following discussion, we stay in a probabilistic framework and always fix $T = 1$. The model for other T values would be an interesting direction to explore in the future.

In this section, we show how to augment standard pairwise CRFs with this type of CHOPPs and describe inference and learning algorithms. We do not enforce any constraint on hidden variables in the following discussion, but it is possible to derive the inference and learning algorithms for the case where we have a soft sparsity or hard 1-of-J constraints on hidden variables.

4.1 Model

We augment a standard pairwise CRF by directly adding the CHOPP to the energy function along with a bias term. For a labeling $\mathbf{y}$, given input image $\mathbf{x}$, the conditional distribution is defined as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\lambda^u \sum_i f_i(y_i|\mathbf{x}) + \sum_k \lambda_k^p \sum_{i,j} f_{ij}^k(y_i, y_j|\mathbf{x}) \right.$$
$$\left. + \sum_i b_i y_i + \sum_j \log\left(1 + \exp\left(c_j + \sum_i w_{ij} y_i\right)\right)\right) \tag{4.6}$$

where $f_i(y_i|\mathbf{x})$ are unary potentials, $f_{ij}^k(y_i, y_j|\mathbf{x})$ are K different types of pairwise potentials, λ^u and λ_k^p are trade-off parameters for unary and pairwise potentials respectively, and w_{ij}, b_i, c_j are RBM parameters. To simplify notation, for a given $\mathbf{x}$ we can denote $\psi^u(\mathbf{y}) = \lambda^u \sum_i f_i(y_i|\mathbf{x})$ for unary potentials, $\psi^p(\mathbf{y}) = \sum_k \lambda_k^p \sum_{i,j} f_{ij}^k(y_i, y_j|\mathbf{x})$ for pairwise potentials, and $\mathbf{b}$, $\mathbf{c}$ and $\mathbf{W}$ for RBM bias vectors and the weight matrix. As mentioned above, we do not enforce any constraint on hidden variables, so we got the CHOPP in the above form.

This is equivalent to the marginal distribution of $\mathbf{y}$ with a vector of binary hidden variables $\mathbf{h}$ summed out from the joint distribution

$$p(\mathbf{y}, \mathbf{h}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\psi^u(\mathbf{y}) + \psi^p(\mathbf{y}) + \sum_i b_i y_i + \sum_{ij} w_{ij} y_i h_j + \sum_j c_j h_j \right) \tag{4.7}$$

Given $\mathbf{y}$, the distribution of $\mathbf{h}$ factorizes, and we have

$$p(h_j = 1|\mathbf{y}, \mathbf{x}) = \sigma\left(c_j + \sum_i w_{ij} y_i\right) \tag{4.8}$$

where σ is the logistic function $\sigma(x) = \frac{1}{1+\exp(-x)}$.

Given $\mathbf{h}$, the distribution of $\mathbf{y}$ becomes a pairwise MRF with only unary and pairwise potentials

$$p(\mathbf{y}|\mathbf{h}, \mathbf{x}) \propto \exp\left((\mathbf{b} + \mathbf{W}\mathbf{h})^\top \mathbf{y} + \psi^u(\mathbf{y}) + \psi^p(\mathbf{y})\right) \tag{4.9}$$

where $(\mathbf{b} + \mathbf{W}\mathbf{h})^\top \mathbf{y} + \psi^u(\mathbf{y})$ is the new unary potential.

These factorization properties are very useful in inference and learning.

One way to make this model even more expressive is to make the RBM energy also conditioned on the input image $\mathbf{x}$. The current formulation of CHOPPs is purely unconditional, but knowing some image evidence can help the model determine which pattern should be active. We achieve this by making the hidden biases $\mathbf{c}$ a function of the input image feature vector $\phi(\mathbf{x})$. The simplest form of this is a linear function $\mathbf{c}(\mathbf{x}) = \mathbf{c}_0 + \mathbf{W}_0^\top \phi(\mathbf{x})$, where $\mathbf{c}_0$ and $\mathbf{W}_0$ are parameters. Then the joint distribution of $\mathbf{y}$ and $\mathbf{h}$ becomes

$$p(\mathbf{y}, \mathbf{h}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\psi^u(\mathbf{y}) + \psi^p(\mathbf{y}) + \mathbf{b}^\top \mathbf{y} + \mathbf{y}^\top \mathbf{W}\mathbf{h} + \phi(\mathbf{x})^\top \mathbf{W}_0 \mathbf{h} + \mathbf{c}_0 \mathbf{h}\right) \tag{4.10}$$

where $\phi(\mathbf{x})$ acts as extra input to the RBM and $\mathbf{W}_0$ is the weight matrix for it. The conditional RBM in [22] used a similar formulation where both the biases for $\mathbf{y}$ and $\mathbf{h}$ are conditioned on image evidence but there is no CRF part in the model.

Another variant of the current formulation is to make the RBM *convolutional* which entails shrinking the window of image labels $\mathbf{y}$ on which a given hidden unit depends, and devoting a separate hidden unit to each application of one of these feature functions to every possible location in the image [16, 23]. These can be trained by tieing together the weights between $\mathbf{y}$ and hidden variables $\mathbf{h}$ at all locations in an image. This significantly reduces the number of parameters in the model, and may have the effect of making the CHOPPs capture more local patterns.

4.2 Inference

The task of inference is to find the $\mathbf{y}$ that maximize the log probability $\log p(\mathbf{y}|\mathbf{x})$ for a given $\mathbf{x}$. Direct optimization is hard, but we utilize a variational lower bound:

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{x}) &= \mathbb{E}_q[\log p(\mathbf{y}, \mathbf{h}|\mathbf{x})] + H(q) + \mathrm{KL}(q||p) \\ &\geq \mathbb{E}_q[\log p(\mathbf{y}, \mathbf{h}|\mathbf{x})] + H(q) \end{aligned} \tag{4.11}$$

where $q(\mathbf{h})$ is any distribution of $\mathbf{h}$, $H(q)$ is the entropy of q and $\mathrm{KL}(q||p)$ is the Kullback-Leibler divergence between q and posterior distribution $p(\mathbf{h}|\mathbf{y}, \mathbf{x})$.

We can use the EM algorithm to optimize the lower bound of the log probability in Eq. 4.11. Starting from an initial labeling $\mathbf{y}$, we alternate the following E step and M step:

In the E step, we fix $\mathbf{y}$ and maximize the bound with respect to q, which is achieved by setting

$q(\mathbf{h}) = p(\mathbf{h}|\mathbf{y}, \mathbf{x})$. We can compute this distribution efficiently using Eq. 4.8.

In the M step, we fix q and find the $\mathbf{y}$ that maximizes the bound. $H(q)$ now becomes a constant since q is fixed, so we only need to maximize $\mathbb{E}_q[\log p(\mathbf{y}, \mathbf{h}|\mathbf{x})]$, which is essentially

$$\left(\mathbf{b} + \mathbf{W}\mathbb{E}_q[\mathbf{h}]\right)^\top \mathbf{y} + \psi^u(\mathbf{y}) + \psi^p(\mathbf{y}) \tag{4.12}$$

plus some constants that do not depend on $\mathbf{y}$. This is again just a set of unary potentials plus pairwise potentials, so we can use standard optimization methods for pairwise CRFs to find an optimal $\mathbf{y}$. In the experiments, we use graph cuts for this step. If the CRF inference algorithm used in the M step is exact, this algorithm will find a sequence of $\mathbf{y}$'s that monotonically increase the log probability, and is guaranteed to converge.

In [22], the authors used mean-field to do inference on a similar model. We can also use mean-field here, where we estimate an approximation $q(\mathbf{y}, \mathbf{h})$ to the conditional joint distribution $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$. We assume $q(\mathbf{y}, \mathbf{h})$ factorizes into $q(\mathbf{y})q(\mathbf{h})$, which would naturally lead to the factorization of $q(\mathbf{h}) = \prod_j q(h_j)$. The update formula for $q(\mathbf{h})$ would be the same as the E-step in the above EM algorithm. However, unlike in [22], $q(\mathbf{y})$ is not easy to estimate because of the pairwise connections in the CRF part of the model. We can further assume $q(\mathbf{y})$ also factorizes as $q(\mathbf{y}) = \prod_i q(y_i)$, then update each $q(y_i)$ based on all its neighbors in each iteration of the mean-field inference. The final prediction is made by independently find the y_i's that maximizes each $q(y_i)$. This will be one of the things to explore in future works.

Remarks on LP Relaxation Inference The CHOPP in Eq. 4.6 is a sum over J terms and each of these terms is a high order potential. It is possible to use modern methods for MAP inference based on linear program (LP) relaxations [30]. In fact, we tried this approach, formulating the "marginal MAP" problem as simply a MAP problem with high order potentials, then using Dual Decomposition to solve the LP relaxation. The key computational requirement is a method for finding the minimum free energy configuration of visibles for an RBM with a single hidden unit, which we were able to do efficiently. However, we found that the energies achieved by this approach were worse than those achieved by the EM procedure described above. We attribute this to looseness in the resulting LP relaxation. This hypothesis is also supported by the results reported by Rother et al. [27], where ordinary belief propagation outperformed LP-based inference, which tends to occur when LP relaxations are loose. Going forward, it would be worthwhile to explore methods for tightening LP relaxations [31].

More details about this can be found in Appendix A.

4.3 Learning

Here we fix the unary and pairwise potentials and focus on learning the parameters in the RBM. CD style learning is the usual way to train RBMs. We can use CD to maximize the conditional likelihood of data under our model. However we found that CD does not work very well because it is only learning the shape of the distribution in a neighborhood around the ground truth (by raising the probability of the ground truth and lowering the probability of everything else). In practice, when doing prediction using the EM algorithm on test data, inference does not generally start near the ground truth. In fact, it typically starts far from the ground truth (we use the prediction by a model with only unary and pairwise potentials as the initialization, which is not bad, but still far from ground truth), and the model has not been trained to move the distribution from this region of label configurations towards the target labels.

Instead, we train the model to minimize expected loss. For any image $\mathbf{x}$ and the ground truth labeling $\mathbf{y}^*$, we have a loss $\ell(\mathbf{y}, \mathbf{y}^*) \geq 0$ for any $\mathbf{y}$. The expected loss is defined as $L = \sum_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})\ell(\mathbf{y}, \mathbf{y}^*)$, where $p(\mathbf{y}|\mathbf{x})$ is the marginal distribution attained by summing out $\mathbf{h}$. The expected loss for a dataset is simply a sum over all individual data cases. The following discussion will be for a single data case to simplify notation.

Taking the derivative of the expected loss with respect to model parameter θ, which can be $\mathbf{b}$, $\mathbf{c}$ or $\mathbf{W}$ ($\mathbf{c}_0$ and $\mathbf{W}_0$ as well if we use the conditioned RBMs), we have

$$\frac{\partial L}{\partial \theta} = \mathbb{E}_{\mathbf{y}}\left[\left(\ell(\mathbf{y}, \mathbf{y}^*) - \mathbb{E}_{\mathbf{y}}[\ell(\mathbf{y}, \mathbf{y}^*)]\right)\mathbb{E}_{\mathbf{h}|\mathbf{y}}\left[-\frac{\partial E}{\partial \theta}\right]\right] \tag{4.13}$$

where $\mathbb{E}_{\mathbf{y}}[.]$ is the expectation under $p(\mathbf{y}|\mathbf{x})$ and $\mathbb{E}_{\mathbf{h}|\mathbf{y}}[.]$ is the expectation under $p(\mathbf{h}|\mathbf{y}, \mathbf{x})$.

$$E(\mathbf{y}, \mathbf{h}) = -\psi^u(\mathbf{y}) - \psi^p(\mathbf{y}) - \mathbf{b}^\top\mathbf{y} - \mathbf{y}^\top\mathbf{W}\mathbf{h} - \mathbf{c}^\top\mathbf{y} \tag{4.14}$$

is the energy function, and $\mathbb{E}_{\mathbf{h}|\mathbf{y}}[-\frac{\partial E}{\partial \theta}]$ is easy to compute.

Using a set of samples $\{\mathbf{y}^n\}_{n=1}^N$ from $p(\mathbf{y}|\mathbf{x})$, we can compute an unbiased estimation of the gradient

$$\frac{\partial L}{\partial \theta} \approx \frac{1}{N-1}\sum_n \left(\ell(\mathbf{y}^n, \mathbf{y}^*) - \frac{1}{N}\sum_{n'}\ell(\mathbf{y}^{n'}, \mathbf{y}^*)\right)\mathbb{E}_{\mathbf{h}|\mathbf{y}^n}\left[-\frac{\partial E}{\partial \theta}\right] \tag{4.15}$$

This gradient has an intuitive explanation: if a sample has a loss lower than the average loss of the batch of samples, then we should reward it by raising its probability, and if its loss is higher than the average, then we should lower its probability. Therefore even when the samples are far from the ground truth, we can still adjust the relative probabilities of the samples. In the process, the distribution is shifted in the direction of lower loss.

We sample from the joint distribution $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ using Gibbs sampling and discard $\mathbf{h}$ to get samples from $p(\mathbf{y}|\mathbf{x})$. Here due to the special model structure, we can use block Gibbs sampling, which significantly improves sampling efficiency. We also use several Markov chains for each image to generate samples, where each chain is initialized at the same initialization as is used for inference. The model parameters are updated after every sampling step.

The learning algorithm we proposed here is different from both of the two algorithms described in [22] for similar conditional RBMs, where neither of the two used any information from the loss function. Both of the two algorithms are optimizing the distribution locally around ground truth, while our method can optimize the distribution more globally. However, as pointed out in [22], running seperate persistent chains for each training example would not work well when using minibatches because the states of the persistent chains would be far from the model distribution after a full pass through the data set since the model parameters will be changed significantly. But we found in the experiments that the gradient computed by Eq. 4.15 is usually quite small, probably because the samples are not too far from each other, so the problems with persistent chains are not that significant here. Additionally, the gradient estimate still makes some sense even when the $\mathbf{y}^n$'s are not samples from $p(\mathbf{y}|\mathbf{x})$, where the effect of the gradient is adjusting the model to assign higher probabilities for the ones with lower loss and do the opposite for the ones with higher loss. This can also help alleviate the problems caused by persistent chains.

More detailed settings for training can be found in Chapter 5.

Chapter 5

Experiments

We evaluate our CHOPP-augmented CRF on synthetic and real data sets. The settings for synthetic data sets will be explained later. For all the real datasets, we extracted a 107 dimensional descriptor for each pixel in an image by applying a filter bank, which includes color features (RGB and Lab, 6 features), responses to Gabor filters (5 filter frequencies and 4 filter orientations, which gives us $5 \times 4 \times 2 = 40$ features), Leung-Malik filters (48 features) and Schmid filters (13 features). We used the implementation of Leung-Malik and Schmid filterbank from http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html. All the filters are applied to the grey scale image. We trained a 2-layer (1 layer of hidden units) neural network classifier using these descriptors as input and use the log probability of each class for each pixel as the unary potentials.

For pairwise potentials, we used a standard 4-connected grid neighborhood and the common Potts model, where $f_{ij}(y_i, y_j|\mathbf{x}) = p_{ij}\mathbf{I}[y_i \neq y_j]$ and p_{ij} is a penalty for assigning different labels for y_i and y_j. Three different ways to define p_{ij} yield three pairwise potentials:

(1) Set p_{ij} to be constant, this would enforce smoothing for the whole image;

(2) Set p_{ij} to incorporate local contrast information by computing RGB differences between pairs of pixels as in [1], where $p_{ij} = \exp\left(-\frac{(I_i - I_j)^2}{2\sigma^2}\right)$, I_i, I_j are RGB values for the two pixels and σ is a parameter controlling the sensitivity to contrast;

(3) Set p_{ij} to represent higher level boundary information given by Pb boundary detector [20], more specifically, we define $p_{ij} = -\max\{\log Pb_i, \log Pb_j\}$ where Pb_i and Pb_j are the probability of boundary for pixel i and j.

For each dataset, we hold out a part of the data to make a validation set, and we use it to choose hyper parameters, e.g. the number of iterations to run in training. We choose the model that performs the best on the validation set and report its performance on the test set.

5.1 Data Sets & Variability

Throughout the experiments, we use six synthetic and three real world data sets. To explore data set variability in a controlled fashion, we generated a series of increasingly variable synthetic data sets. The datasets are composed of between 2 and 4 ellipses with centers and sizes chosen to make the figures

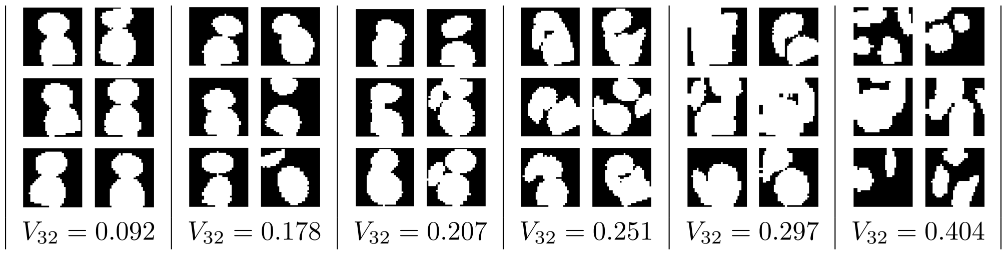


Figure 5.1: Randomly sampled examples from synthetic data set labels. Hardness increases from left to right. Quantitative measures of variability using $K = 32$ are reported in the bottom row. Variabilities of Horse, Bird, and Person data sets are 0.176, 0.370, and 0.413.

look vaguely human-like (or at least snowman-like). We then added noise to the generation procedure to produce a range of six increasingly difficult data sets, which are illustrated in Fig. 5.1 (top row). To generate associated unary potentials, we added Gaussian noise with standard deviation 0.5. In addition, we added structured noise to randomly chosen 5-pixel diameter blocks.

The real world data sets come from two sources: first, we use the Weizmann horses and resized all images as well as the binary masks to 32×32; second, we use the PASCAL VOC 2011 segmentation data [3] to construct a bird and a person data set. For these, we take all bounding boxes containing the target class and created a binary segmentation of the inside of the bounding box, labeling all pixels of the target class as 1, and all other pixels as 0. We then transformed these bounding boxes to be of size 32×32. This gives us a set of silhouettes that preserve the challenging aspects of modeling shape in a realistic structured output setting. Images in all three real data sets are shown in Section 5.4.

The two PASCAL datasets are challenging due to variability in the images and segmentations, while the number of images is quite small (214 images for birds and 1169 for person), especially compared to the settings where RBM models are typically used. When we are only training the trade-off parameters, this is not a major problem, because the number of parameters is small. But here we also train internal parameters of high order potentials, which require more data for training to work well. To deal with this problem, we generated 5 more bounding boxes for each original bounding box by randomly shifting coordinates by a small amount. We also mirrored all images and segmentations. This augmentation gives us 12 times as many training examples.

For each data set, we then evaluated variability. To do so, we propose a measure inspired by the learning procedure suggested by Rother et al. [27]. First, cluster segmentations using K-means clustering with Euclidean distance as the metric. Then for each cluster and pixel, compute the fraction of cases for which the pixel is on across all instances assigned to the cluster. This yields q_{ij}^k, the probability that pixel ij is assigned label 1 given that it comes from an instance in cluster k. Now define the within cluster average entropy $H^k = -\frac{1}{D_v}\sum_{ij}\left(q_{ij}^k \log q_{ij}^k + (1-q_{ij}^k)\log(1-q_{ij}^k)\right)$, where D_v is the number of pixels in the image. Finally, the variability measure is a weighted average of within cluster average entropies: $V_K = \sum_{k=1}^{K} \mu_k H^k$, where μ_k is the fraction of data points assigned to cluster k. We found $K = 32$ to work well and used it throughout. We found the quantitative measure matches intuition about the variability of data sets as shown in Fig. 5.1.

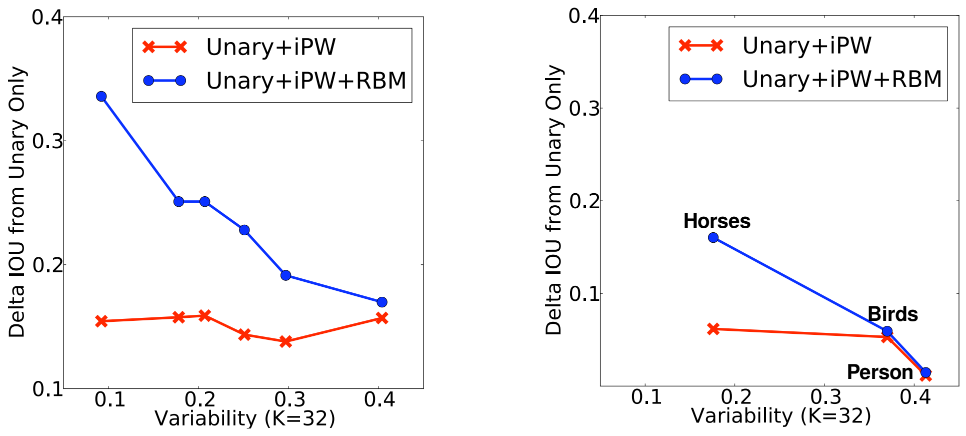


Figure 5.2: Results on (left) synthetic and (right) real data showing test intersection-over-union scores as a function of data set variability. The y-axis is the difference relative to Unary Only model. Note that these results are for the pretrained RBM model.

5.2 Performance vs. Variability

Next we report results for a pre-trained RBM model added to a standard CRF (denoted RBM). Here, we learn the RBM parameters offline and set tradeoff parameters so as to maximize accuracy on the training set. We compare the Unary Only model to the Unary+Pairwise model and the Unary+Pairwise+RBM model. Pairwise terms are image dependent, meaning that all 3 types of pairwise potentials are used, which is denoted by iPW. Fig. 5.2 shows the results as a function of the variability measure described in the previous section. On the y-axis, we show the difference in performance between the Unary+iPW and Unary+iPW+RBM models versus the Unary Only model. In all but the Person data set, the Unary+iPW model provides a consistent benefit over the Unary Only model. For the Unary+iPW+RBM model, there is a clear trend that as the variability of the data set increases, the benefit gained from adding the RBM declines.

5.3 Improving on Highly Variable Data

We now turn our attention to the challenging real data sets of Birds and Person and explore methods for improving the performance of the RBM component when the data becomes highly variable.

Training with Expected Loss The first approach to extending the pretrained RBM+CRF model that we consider is to jointly learn the internal potential parameters $\mathbf{W}$. Initial experiments with standard contrastive divergence learning on the Horse data led to poor performance, as the learning was erratic in the first few iterations and then steadily got worse during training. So here we focus on the offline pretraining and the expected loss training described in Section 4.3. We use 2 sampling chains[1] for each image and use the validation set to do early stopping. The learning rate is fixed and chosen from $\{10, 1, 0.1, 0.01\}$ (the gradients are quite small so we tried some large learning rates here) so that it is small enough to avoid erratic behavior and big enough to make significant updates of the weights in reasonable time. We denote the resulting RBM models as jRBM to indicate joint training. Results comparing these approaches on the three real data sets are given in Fig. 5.3, with Unary+iPW results

[1]We tried 10 sampling chains for each image as well, but it didn't give us any significant performance boost over 2 sampling chains and it was much slower.

Method	Horse IOU	Bird IOU	Person IOU
Unary Only	0.5119	0.5055	0.4979
iPW	0.5736	0.5585	0.5094
iPW+RBM	0.6722	0.5647	0.5126
iPW+jRBM	**0.6990**	**0.5773**	**0.5253**

Figure 5.3: Expected loss test results. RBM is a pretrained RBM. jRBM is jointly trained using expected loss.

Method	Bird IOU	Person IOU
PW	0.5321	0.5082
iPW	0.5585	0.5094
iPW+jRBM	0.5773	**0.5253**
iPW+ijRBM	**0.5858**	**0.5252**

Figure 5.4: Test results using image-specific hidden biases on the high variability real data sets. PW uses image-independent pairwise potentials, and iPW uses image-dependent pairwise potentials. jRBM is jointly trained but image independent. ijRBM is jointly trained and has learned image-dependent hidden biases.

given as a baseline. We see that training with the expected loss criterion improves performance across the board.

Image-dependent Hidden Biases Here, we consider learning image-dependent hidden biases as described in Section 4.1 (modeling hidden biases $\mathbf{c}$ as a linear function of some image feature $\phi(\mathbf{x})$). As inputs, we use the learned unary potentials and the response of the Pb boundary detector [20], both downsampled to be of size 16×16. We jointly learned the RBM internal parameters using the intersection-over-union expected loss, as this gave the best results in the previous experiment. We refer to these jointly trained, image-dependent RBMs with ijRBM. Results are shown in Fig. 5.4. For comparison, we also train Unary+Pairwise models with a image-independent pairwise potentials (PW) along with the standard image-dependent pairwise potentials (iPW). In the Bird data, we see that the image-specific information helps the ijRBM similarly as how image-dependent pairwise potentials improve over image-independent pairwise potentials. In the Person data, the gains from image-dependent information is minimal in both cases.

Convolutional Structures Our final experiment explores the convolutional analog to the RBM models discussed in Section 4.1 . Unfortunately, we were unable to achieve good results. We tried two variants: (a) a vanilla pre-trained convolutional RBM, and (b) a pre-trained convolutional RBM with conditional hidden biases as described in Section 4.1. We tried two different patch sizes (8×8, 12×12) and tiled the images densely. Though the conditional variant outperformed the unconditional variant, overall results were discouraging—performance was not even as good as the simple Unary+Pairwise model. This is surprising because a convolutional RBM should in theory be able to easily represent pairwise potentials, and convolutional RBMs have fewer parameters than their global counterparts, so overfitting should not be an issue. We believe the explanation for the poor performance is that learning methods for convolutional RBMs are not nearly as evolved as methods for learning ordinary RBMs, and thus the learning methods that we have at our disposal do not perform as well. On the bright side, this can be seen as a challenge to overcome in future work. A few methods developped for tiled convolutional (not fully convolutional) RBMs achieved good results modeling textures [8][19], which shows some potential that this may be a good way to go.

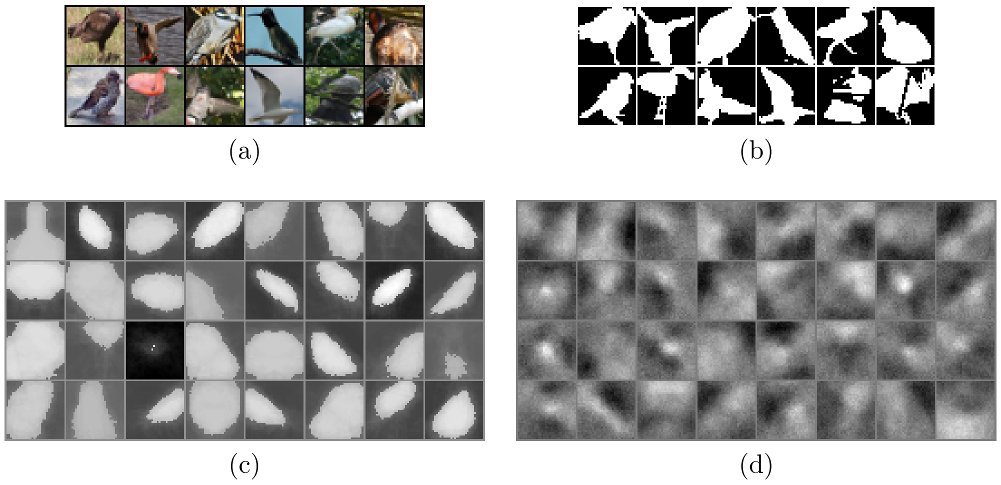


Figure 5.5: (a) Images from Bird data set. (b) Ground truth labels. (c) Patterns learned by clustering-style approach of [27]. (d) Patterns learned by compositional-style learning used in this paper.

Composition Schemes We qualitatively compare patterns learned for the "min" composition approach presented in [27] using k-means versus the patterns learned by a simple pre-trained RBM, which are appropriate for "sum" composition. While a quantitative comparison that explores more degrees of freedom offered by CHOPPs is a topic for future work, we can see in Fig. 5.3 that the filters learned are very different. As the variability of the data grows, we expect the utility of the "sum" composition scheme to increase.

5.4 More Experiment Results

5.4.1 Real Data Sets

Images in the three real data sets are shown in Fig. 5.6 and Fig. 5.7.

The original Weizmann horses data set can be found from `http://www.msri.org/people/members/eranb/` and PASCAL VOC data set from `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2011/`.

Our version of the three data sets as well as the 6 synthetic data sets will be available online.

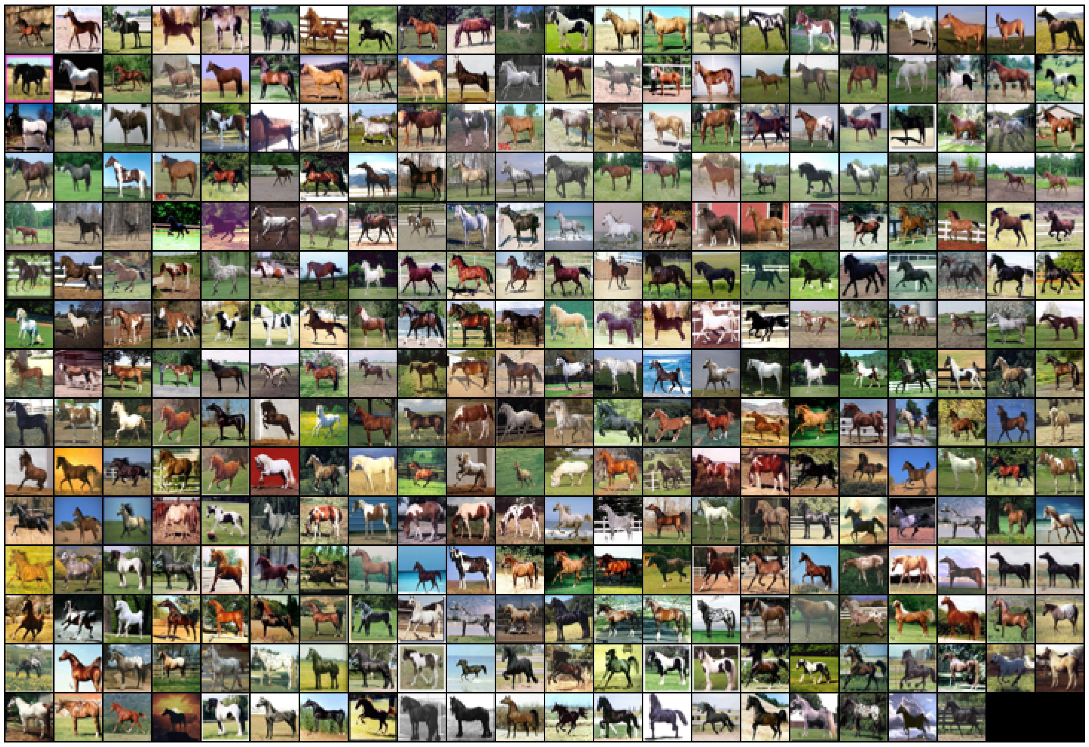
5.4.2 Learned Filters

The learned filters, i.e. weights w_{ij}, with a pretrained RBM for each of the three data sets, are shown in Fig. 5.8. Filters for 6 synthetic data sets are shown in Fig. 5.9 and Fig. 5.10, with hardness level from easy to hard (0 to 5). For each filter, the weights are positive for bright regions and negative for dark regions. In other words, filters favor bright regions to be on and dark regions to be off.

We can see the compositional nature of RBMs from these filters. For example, each single horse filter is actually expressing soft rules like "if the head of a horse is here, then the legs are likely to be there". Any single filter would not make too much sense, but only when a few different filters are combined can we recover a horse.

5.4.3 Prediction Results

Some example segmentations for horse, bird and person data sets are given in Fig. 5.11, Fig. 5.12 and Fig. 5.13.


(a) Horse data set, 328 images in total.

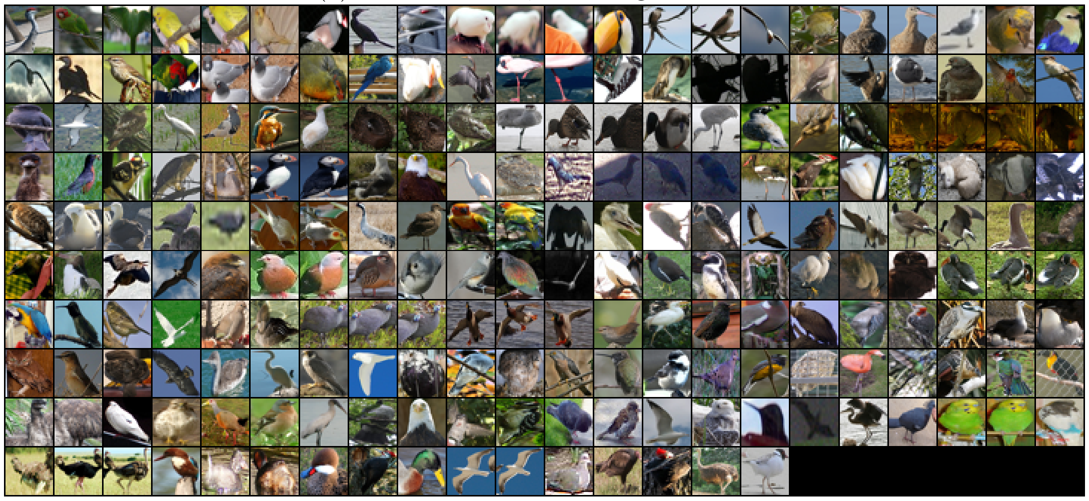
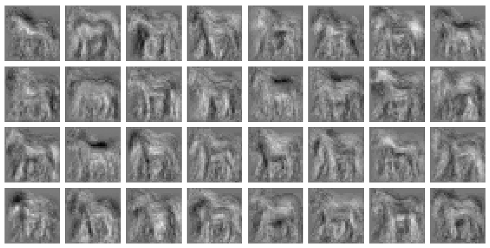
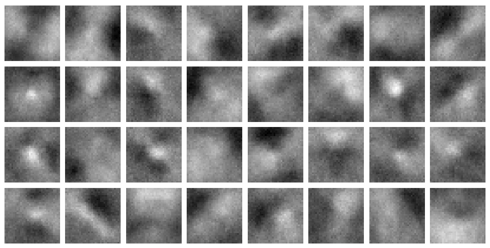

(b) Bird data set, 214 images in total.

Figure 5.6: Horse and bird data sets.

Figure 5.7: Person data set, 1169 images in total.


(a) Horse filters


(b) Bird filters

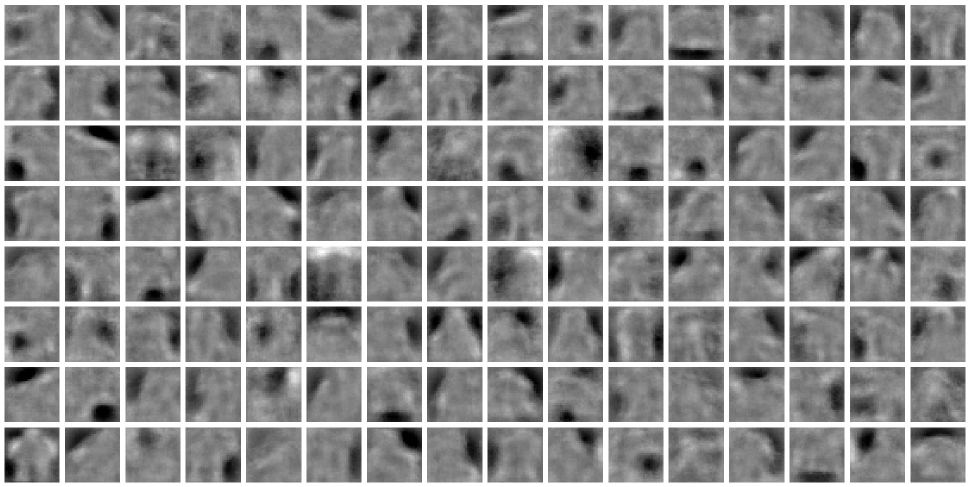
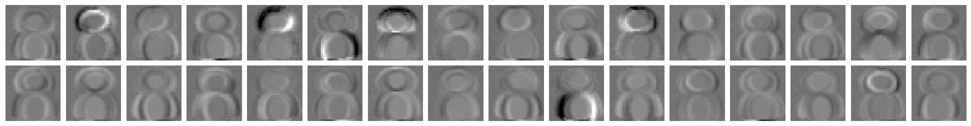
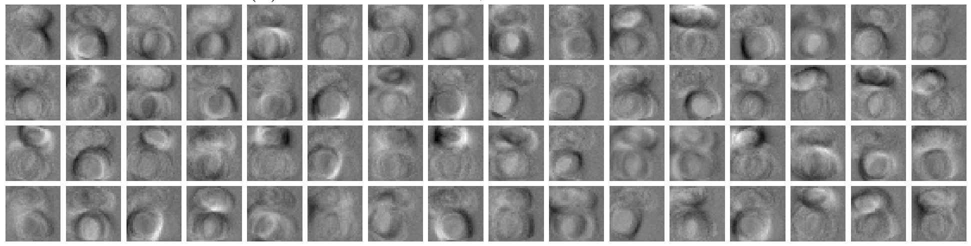
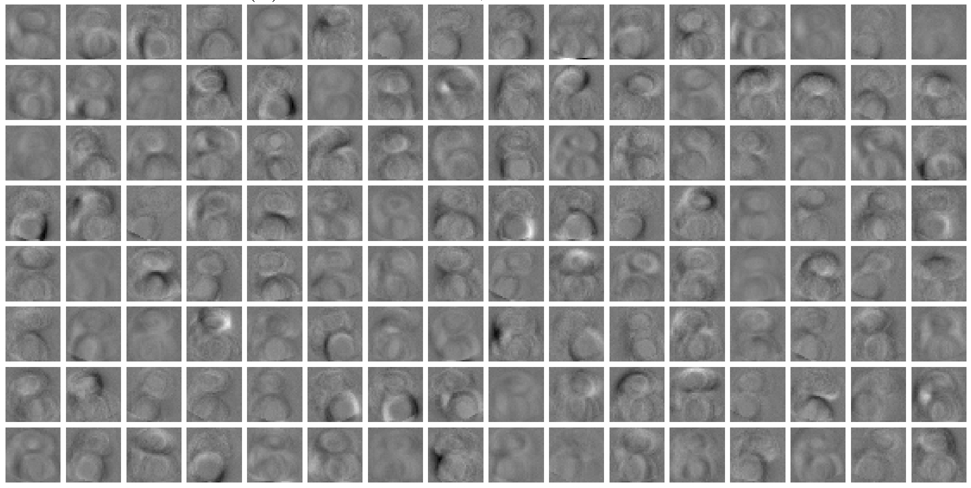

(c) Person filters.

Figure 5.8: Filters learned on three real data sets.


(a) Hardness level 0, 32 hidden variables.


(b) Hardness level 1, 64 hidden variables.


(c) Hardness level 2, 128 hidden variables.

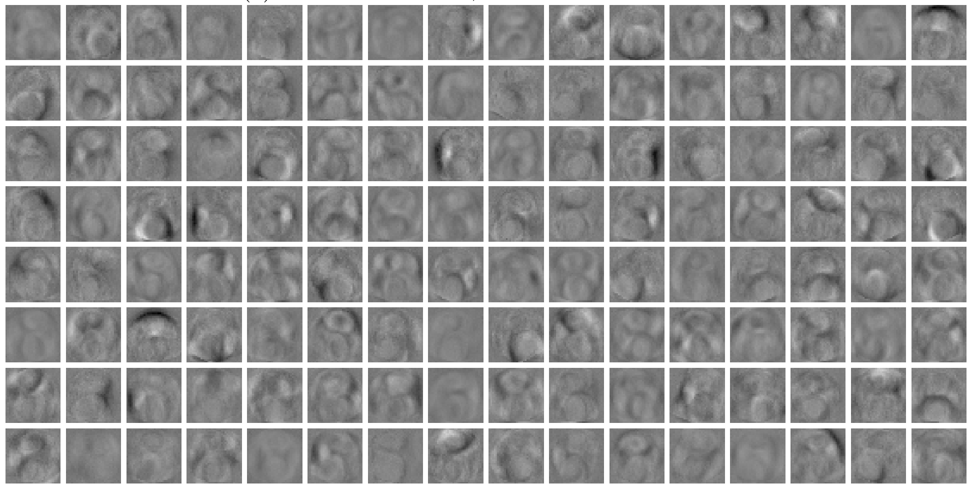
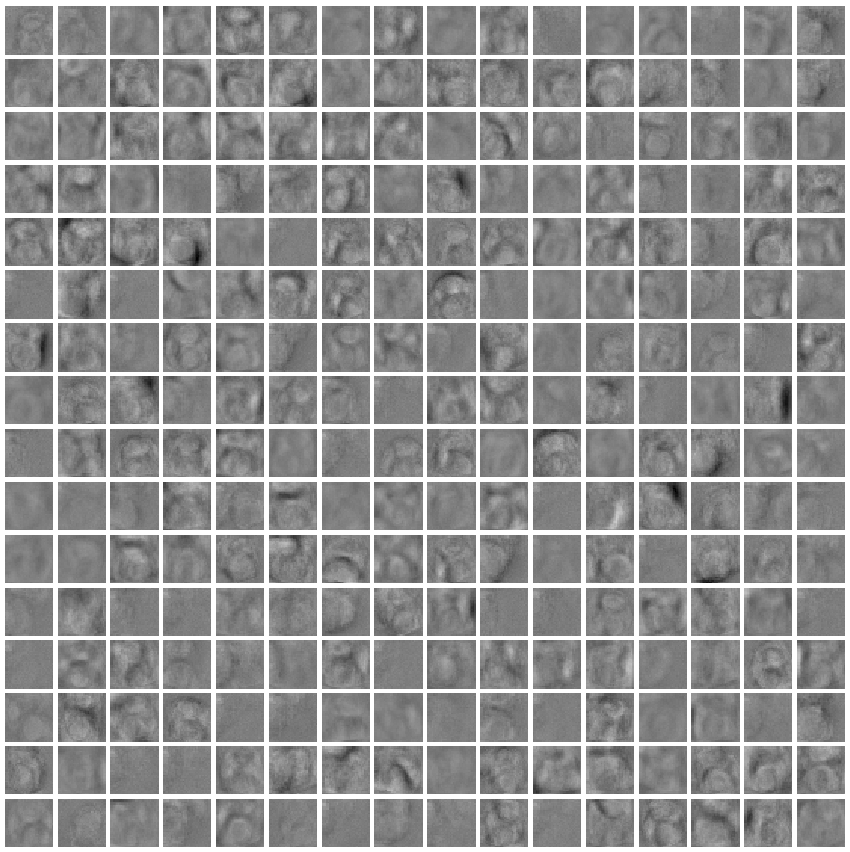

(d) Hardness level 3, 128 hidden variables.

Figure 5.9: Filters learned on synthetic data sets.


(e) Hardness level 4, 256 hidden variables.

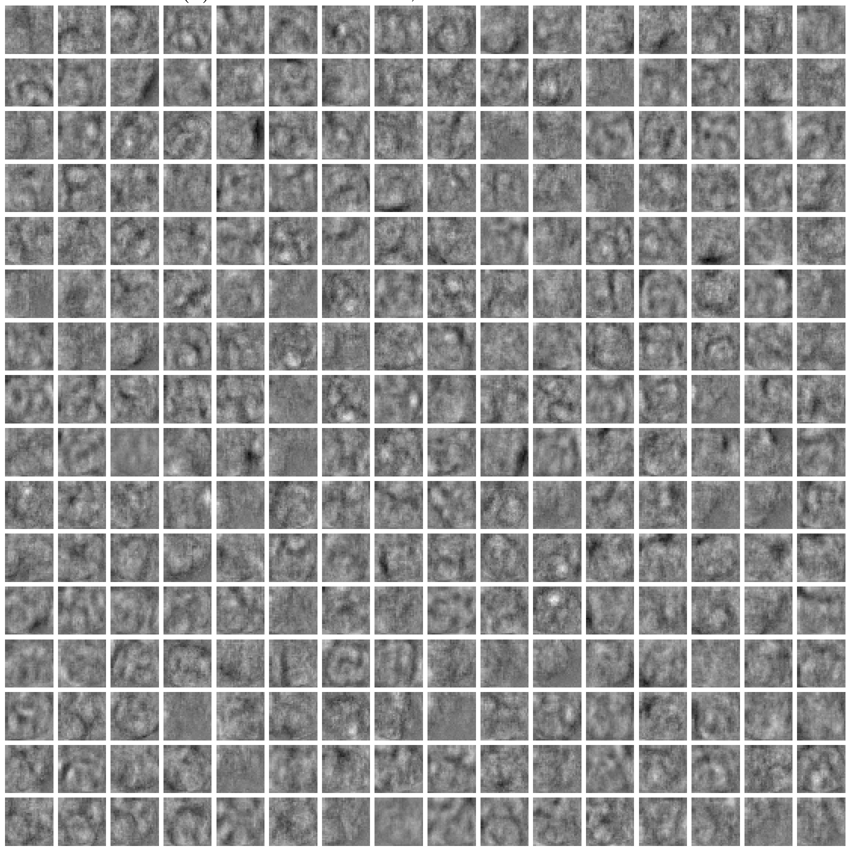
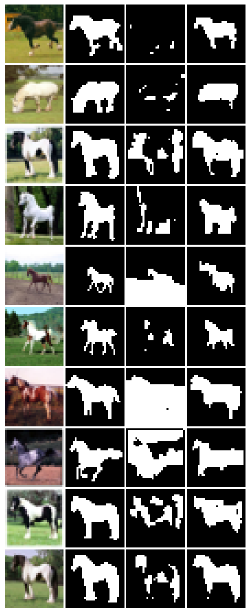
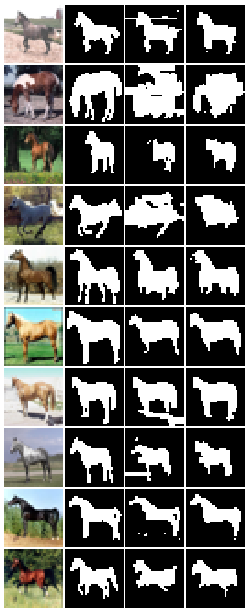

(f) Hardness level 5, 256 hidden variables.

Figure 5.10: Filters learned on synthetic data sets, continued.


(a) Best


(b) Average

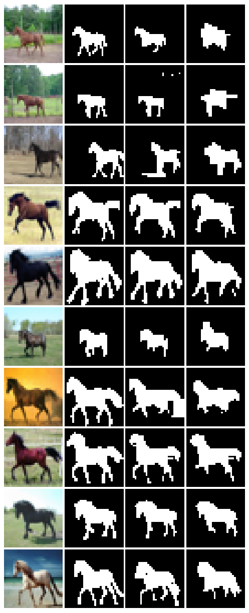
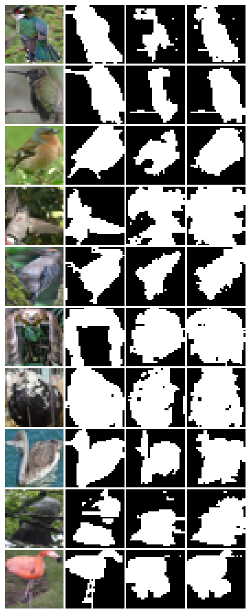
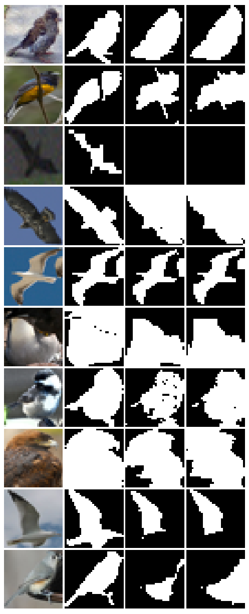

(c) Worst

Figure 5.11: Prediction results on horse data set. The three categories best, average and worst are measured by the improvement of Unary+Pairwise+RBM over Unary+Pairwise. Each row left to right: original image, ground truth, Unary+Pairwise prediction, Unary+Pairwise+RBM prediction.


(a) Best


(b) Average

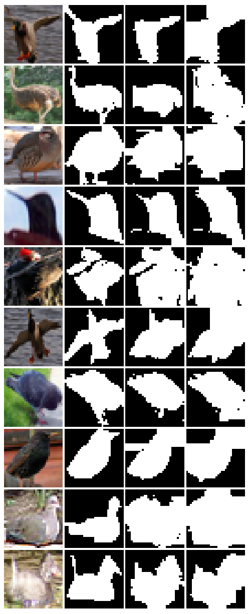
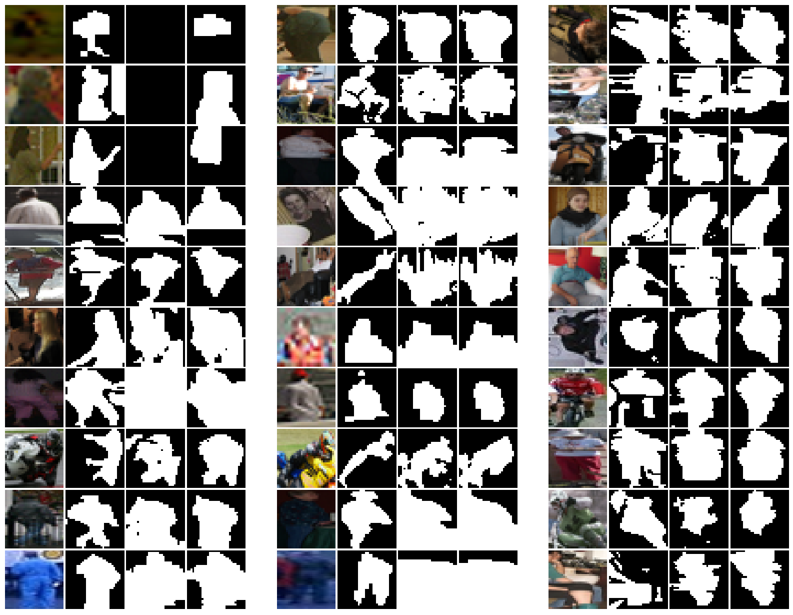

(c) Worst

Figure 5.12: Prediction results on bird data set.



(a) Best (b) Average (c) Worst

Figure 5.13: Prediction results on person data set.

Chapter 6

Discussion & Future Work

We began by precisely mapping the relationship between pattern potentials and RBMs, and generalizing both to yield CHOPPs, a class of high order potential that includes both as special cases. The main benefit of this mapping is that it allows the leveraging of complementary work from two mostly distinct communities. First, it opens the door to the large and highly evolved literature on learning RBMs. These methods allow efficient and effective learning when there are hundreds or thousands of latent variables. There are also well-studied methods for adding structure over the latent variables, such as sparsity. Conversely, RBMs may benefit from the highly developed inference procedures that are more common in the structured output community e.g. those based on linear programming relaxations. Also interesting is that pairwise potentials provide benefits that are reasonably orthogonal to those offered by RBM potentials.

Empirically, our work emphasizes the importance of data set variability in the performance of these methods. It is possible to achieve large gains on low variability data, but it is a challenge on high variability data. Our proposed measure for quantitatively measuring data set variability is simple but useful in understanding what regime a data set falls in. This emphasizes that not all "real" data sets are created equally, as we see moving from Horse to Bird to Person. While we work with small images and binary masks, we believe that the high variability data sets we are using preserve the key challenges that arise in trying to model shape in real image segmentation applications. Note that it would be straightforward to have a separate set of shape potentials per object class within a multi-label segmentation setting.

To attain improvements in high variability settings, more sophisticated methods are needed. Our contributions of training under an expected loss criterion and adding conditional hidden biases to the model yield improvements on the high variability data. There are other architectures to explore for making the high order potentials image-dependent. In future work, we would like to explore multiplicative interactions [26]. The convolutional approach appears promising, but it did not yield improvements in our experiments, which we attribute to the relatively nascent nature of convolutional RBM learning techniques. A related issue that should be explored in future work is the issue of sparsity in latent variable activations. We showed in Chapter 3 that this sparsity can be used to control the type of compositionality employed by the model (extreme 1-of-J sparsity vs. no sparsity). An interesting direction for future work is exploring sparse variants of RBMs, which sit in between these two extremes, and other forms of structure over latent variables like in deep models.

Bibliography

[1] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2001. 2, 17

[2] S. Eslami, N. Heess, and J. Winn. The shape Boltzmann machine: a strong model of object shape. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5

[3] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision (IJCV)*, 2010. 1, 18

[4] S. Gould. Max-margin learning for lower linear envelope potentials in binary markov random fields. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2011. 36

[5] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 4

[6] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale conditional random fields for image labelling. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. 5

[7] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 2002. 5

[8] J. Kivinen and C. Williams. Multiple texture Boltzmann machines. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 22, 2012. 1, 20

[9] P. Kohli, L. Ladickỳ, and P. Torr. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision (IJCV)*, 82(3), 2009. 1, 3

[10] N. Komodakis. Efficient training for pairwise or higher order CRFs via dual decomposition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 4

[11] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order mrfs. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 1

[12] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS) 24*, pages 118–126, 2011. 1

[13] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Inference methods for CRFs with co-occurrence statistics. *International Journal of Computer Vision (IJCV)*, 2011. 3, 4

[14] L. Ladickỳ, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision (IJCV)*, 2012. 4

[15] H. Larochelle and Y. Bengio. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pages 536–543. ACM, 2008. 5

[16] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of International Conference on Machine Learning (ICML)*, 2009. 13

[17] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2009. 4

[18] N. LeRoux, N. Heess, J. Shotton, and J. Winn. Learning a generative model of images by factoring appearance and shape. *Neural Computation*, 2011. 5

[19] H. Luo, P. Carrier, A. Courville, and Y. Bengio. Texture modeling with convolutional spike-and-slab rbms and deep extensions. *arXiv preprint arXiv:1211.5687*, 2012. 20

[20] M. Maire, P. Arbeláez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 2, 17, 20

[21] K. Miller, M. Kumar, B. Packer, D. Goodman, and D. Koller. Max-margin min-entropy models. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012. 3

[22] V. Mnih, H. Larochelle, and G. Hinton. Conditional restricted boltzmann machines for structured output prediction. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2011. 5, 13, 14, 15

[23] M. Norouzi, M. Ranjbar, and G. Mori. Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 13

[24] S. Nowozin and C. Lampert. Global connectivity potentials for random field models. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 4

[25] A. Quattoni, S. Wang, L. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2007. 3

[26] M. R. and G. Hinton. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural Computation*, 2010. 30

[27] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. ii, 1, 2, 4, 6, 7, 12, 14, 18, 21

[28] A. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction with latent variables for general graphical models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2012. 11

[29] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In *Parallel distributed processing*. 1986. 4

[30] D. Sontag, A. Globerson, and T. Jaakkola. Introduction to dual decomposition for inference. In S. Sra, S. Nowozin, and S. J. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2011. 14, 35

[31] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening lp relaxations for MAP using message passing. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, 2008. 14

[32] Y. Tang, R. Salakhutdinov, and G. Hinton. Robust boltzmann machines for recognition and denoising. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5

[33] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of International Conference on Machine Learning (ICML)*, 2004. 3

[34] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 4

[35] S. Vicente, V. Kolmogorov, and C. Rother. Joint optimization of segmentation and appearance models. In *International Conference on Computer Vision (ICCV)*. 2009. 3

[36] O. Woodford, C. Rother, and V. Kolmogorov. A global perspective on MAP inference for low-level vision. In *International Journal of Computer Vision (IJCV)*, 2009. 3

[37] C. Yu and T. Joachims. Learning structural svms with latent variables. In *Proceedings of International Conference on Machine Learning (ICML)*, 2009. 3

Appendix A

Dual Decomposition Inference for CHOPP-Augmented CRFs

The negative energy function of CHOPP-augmented CRF defined in Eq. 4.6 is a sum of standard pairwise CRF energy terms and J high order terms. It can be written as

$$f(\mathbf{y}) = \psi(\mathbf{y}) + \sum_{j=1}^{J} \phi_j(\mathbf{y}) \tag{A.1}$$

where $\phi_j(\mathbf{y}) = \log\left(1 + \exp\left(c_j + \sum_i w_{ij} y_i\right)\right)$ is the softplus term induced by summing out the jth hidden variable, and $\psi(\mathbf{y})$ includes everything else, which is just a sum of unary and pairwise potentials.

Dual decomposition inference applies when the function to be maximized can be written as a sum, where maximizing each individual term in the sum is tractable. We have a few different ways to formulate Eq. A.1 as a sum:

Form 1 The current formulation is already a sum of $J + 1$ terms;

Form 2 we can divide $\psi(\mathbf{y})$ equally into J pieces, then the formulation becomes a sum of J terms in the form of $\frac{1}{J}\psi(\mathbf{y}) + \phi_j(\mathbf{y})$.

We will show later that maximizing each individual term for these formulations are tractable in polynomial time even with the high-order potential $\phi_j(\mathbf{y})$.

Once the inference problem becomes

$$\max_{\mathbf{y}} \sum_j \phi'_j(\mathbf{y}) \tag{A.2}$$

we can introduce a set of auxiliary variables $\boldsymbol{\delta} = \{\boldsymbol{\delta}_j\}$, constructed so that $\sum_j \boldsymbol{\delta}_j = \mathbf{0}$. So

$$\begin{aligned}
\max_{\mathbf{y}} \sum_j \phi'_j(\mathbf{y}) &= \max_{\mathbf{y}} \left(\sum_j \phi'_j(\mathbf{y}) + \left(\sum_j \boldsymbol{\delta}_j^\top \right) \mathbf{y} \right) \\
&= \max_{\mathbf{y}} \sum_j \left(\phi'_j(\mathbf{y}) + \boldsymbol{\delta}_j^\top \mathbf{y} \right) \qquad \text{(A.3)} \\
&\le \sum_j \max_{\mathbf{y}} \left(\phi'_j(\mathbf{y}) + \boldsymbol{\delta}_j^\top \mathbf{y} \right) \qquad \text{(A.4)}
\end{aligned}$$

This bound holds for any $\boldsymbol{\delta}$, therefore

$$\max_{\mathbf{y}} \sum_j \phi'_j(\mathbf{y}) \le \min_{\boldsymbol{\delta}} \sum_j \max_{\mathbf{y}} \left(\phi'_j(\mathbf{y}) + \boldsymbol{\delta}_j^\top \mathbf{y} \right) \qquad \text{(A.5)}$$

It is shown in [30] that this bound is tight when all the maximizing configurations of $\mathbf{y}$ are consistent across all the individual terms in the sum. Though this condition is usually not met in practice, the right hand side can still be a reasonable bound on the original MAP problem. The maximization inside the sum is easy to deal with, as we will show later, and we can use sub-gradient descent to update $\boldsymbol{\delta}$. Therefore to minimize the bound, we iterate the two steps until convergence. Then we try to combine all the $\mathbf{y}$'s we get for each individual term to make a final prediction.

We now focus on two parts in the procedure: (1) maximization of each individual term in the sum; (2) making the final prediction after inference has converged.

A.1 Maximization of an Individual Term in the Sum

Each individual maximization problem has the following form[1]

$$\max_{\mathbf{y}} \psi'(\mathbf{y}) + \boldsymbol{\delta}^\top \mathbf{y} + \log \left(1 + \exp \left(c + \sum_i w_i y_i \right) \right) \qquad \text{(A.6)}$$

where $\psi'(\mathbf{y})$ is a sum of unary and pairwise potentials(for Form 1 it is 0, for Form 2 it is $\psi(\mathbf{y})/J$). The softplus function is the hard part for inference. We ignored the subscript j because we are only dealing with a single term in the sum. Denote $h(x) = \log(1 + \exp(x))$ and use convex duality, we have

$$h(x) = \max_{\lambda} [\lambda x - h^*(\lambda)] \qquad \text{(A.7)}$$

where

$$h^*(x) = \max_{\lambda} [\lambda x - h(\lambda)] = x \log x + (1 - x) \log(1 - x) \qquad \text{(A.8)}$$

[1]We omitted the $\psi(\mathbf{y}) + \boldsymbol{\delta}^\top \mathbf{y}$ term in Form 1 because it is nothing but a standard CRF inference problem ($\boldsymbol{\delta}^\top \mathbf{y}$ is just a sum of unary potentials) and easy to deal with.

Therefore the maximization becomes

$$\max_{\mathbf{y}} \left\{ \psi'(\mathbf{y}) + \boldsymbol{\delta}^\top \mathbf{y} + \max_{\lambda} \left[\lambda \left(c + \sum_i w_i y_i \right) - h^*(\lambda) \right] \right\} \tag{A.9}$$

$$= \max_{\lambda} \left\{ \max_{\mathbf{y}} \left[\psi'(\mathbf{y}) + (\boldsymbol{\delta} + \lambda \mathbf{w})^\top \mathbf{y} \right] + \lambda c - h^*(\lambda) \right\} \tag{A.10}$$

For Form 1, there is nothing in $\psi'(\mathbf{y})$; for Form 2, if we only have unary potentials in $\psi'(\mathbf{y})$ the maximization over $\mathbf{y}$ can be written as $\max_{\mathbf{y}} (\boldsymbol{\delta}' + \lambda \mathbf{w})^\top \mathbf{y}$, where the unary potentials from $\psi'(\mathbf{y})$ have been absorbed into $\boldsymbol{\delta}'$. In both cases the optimization can be rewritten as

$$\max_{\lambda} \left[\max_{\mathbf{y}} \left(\boldsymbol{\delta}' + \lambda \mathbf{w} \right)^\top \mathbf{y} + \lambda c - h^*(\lambda) \right] \tag{A.11}$$

Since $\mathbf{y}$ is binary, the optimal $y_i = \mathbf{I}[\delta'_i + \lambda w_i > 0]$ where $\mathbf{I}$ is the indicator function and $\mathbf{I}[x] = 1$ if x is true and $\mathbf{I}[x] = 0$ otherwise. Each y_i would only flip from 1 to 0 or 0 to 1 when λ crosses the break point $-\delta'_i / w_i$. The break points divide the real numbers in a few continuous regions and in each region there is one optimal $\mathbf{y}$ for the corresponding λ. Therefore by enumerating all the regions, we get a polynomial time algorithm to find the optimal $\mathbf{y}$.

If we have pairwise terms for $\psi'(\mathbf{y})$ in Form 2, the optimization would be harder but still solvable in polynomial time, for example, using the lower linear envelope graph cut algorithm [4].

A.2 Making the Final Prediction

We would not have the problem of making a final prediction if all optimal $\mathbf{y}$'s for each individual term in the sum agree. However this is usually not the case so we need a "decoding" method to combine all different $\mathbf{y}$'s and get a final prediction from all the terms and λ's.

A straight-forward method to combine all the $\mathbf{y}$'s is to use the majority for each y_j.

A better way is to utilize the λ's for each individual term. Note that in Eq. A.7, the λ that makes the equality holds is given by

$$\lambda = \frac{1}{1 + \exp(-x)} = \sigma(x) \tag{A.12}$$

which is the logistic sigmoid function. Substitute x by $c + \sum_i w_i y_i$, where $\mathbf{y}$ is the optimal $\mathbf{y}$ we found for this term, we then have

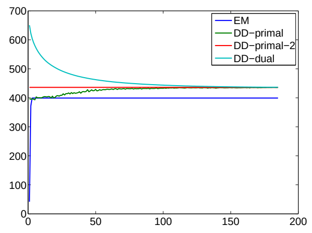
$$\lambda = \sigma \left(c + \sum_i w_i y_i \right) \tag{A.13}$$

which is exactly the same as $p(h|\mathbf{y})$ in Eq. 4.8. We can then run one M-step inference to combine all the terms and get a final prediction of $\mathbf{y}$.

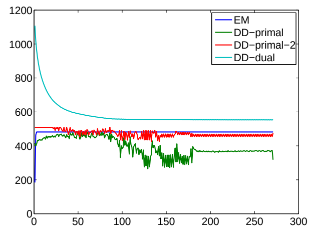
A.3 Experiment Results

We tried dual decomposition inference for a CHOPP-augmented CRF which does not have pairwise potentials. Form 2 of the above is used to develop the inference algorithm.

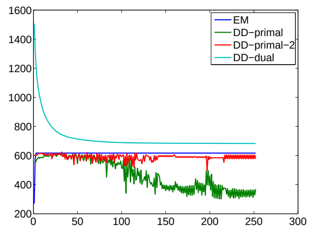
We compared the bound given by dual-decompositoin inference with the bound given by the EM algorithm introduced in Section 4.2. The dual-decomposition bound is quite tight when the number



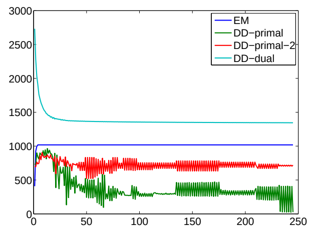
(a) 4 hidden variables



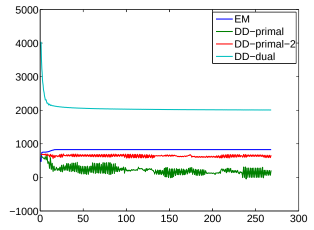
(b) 8 hidden variables



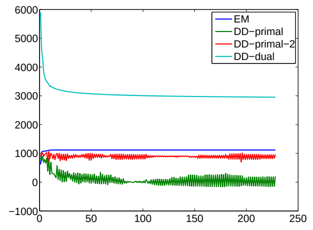
(c) 16 hidden variables



(a) 32 hidden variables



(b) 64 hidden variables



(c) 128 hidden variables

Figure A.1: Inference results for an example image, using models with varying number of hidden variables. The x-axis shows the number of iterations, and the y-axis shows the value of the objective function. DD-primal (green) is the solution found by taking majority, DD-primal-2 (red) is the solution found by the introduced decoding method, and DD-dual (blue) is the dual upper bound. We can see that dual decomposition can find better solutions than EM for small models, while EM is usually better for larger models, and the dual bound found by dual decomposition becomes extremely loose for larger models.

of hidden variables is small. Sometimes it is even possible to find the global optimal solution using dual-decomposition. The bound gets looser as the number of hidden variables grows. The bound get very loose even for small models with only 32 hidden variables. Though the decoding method described in the section above gives better solutions than the straight-forward majority method, they are still far worse than EM on large models, which makes it not applicable for practical problems. Fig. A.1 shows the inference results for an example image using models with varying number of hidden variables.

We also found that better inference does not always lead to better performance. We developed a branch and bound inference algorithm, which is guaranteed to find the optimal $\mathbf{y}$ (in terms of the energy) for small models and will at least find a better $\mathbf{y}$ than both the EM prediction and the dual-decomposition prediction within a limited time period. However, we found that the solutions given by branch and bound do not always lead to better performance than EM predictions, evaluated using the given loss. Part of the reason may be that the EM inference algorithm is better matched with the learning algorithm, which is actually optimizing an approximation (using Monte Carlo samples) of the true objective. Therefore it may be beneficial to match the inference algorithm with the learning algorithm.