Exploring Compositional High Order Pattern Potentials for Structured Output Learning

Yujia Li, Daniel Tarlow, Richard Zemel
University of Toronto
Toronto, ON, Canada, M5S 3G4
{yujiali, dtarlow, zemel}@cs.toronto.edu

Abstract

When modeling structured outputs such as image segmentations, prediction 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. [26] 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 datasets, and we define a quantitative variability measure to aid in studying this. We then improve CHOPPs performance in high variability datasets 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. Quantitative results on challenging highly variable datasets show that the joint learning and image-dependent high order potentials can improve performance.

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.

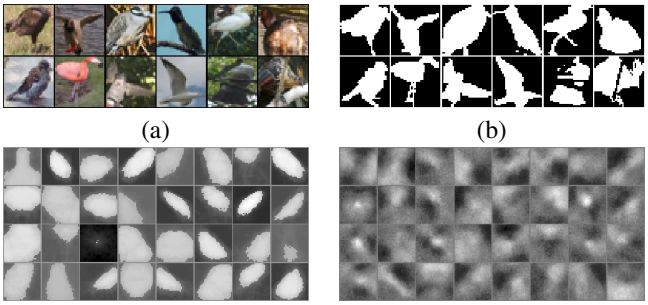

Figure 1. (a) Images from Bird data set. (b) Ground truth labels. (c) Patterns learned by the clustering-style approach of [26]. (d) Patterns learned by the compositional-style approach proposed here.

A key research issue that arises when working with structured output problems is how to best tradeoff 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 [13] or by enforcing smoothness in higher order neighborhoods [10]. 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. [26] and Komodakis & Para-

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.14

gios [12], which are capable of modeling soft template structures and can dramatically outperform pairwise models in highly structured settings that arise, e.g., 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 [4].

A model that is appropriate in similar situtations and has also found success modeling textures [9] 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 [26]. We will make this claim precise in Section 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 that allow 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 detector responses like Pb [19]. 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 (IOU) 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. [26], 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.

2. Background & Related Work

2.1. Structured Output Learning

In structured output learning, the goal is to predict a vector of labels $\mathbf{y} \in \mathcal{Y} = \{1, \ldots, C\}^{D_v}$ given inputs $\mathbf{x} \in \mathcal{X}$, where D_v is the dimensionality of the output. A standard approach, which is taken by e.g. structural SVMs, is to define an input-to-output mapping function $g_{\boldsymbol{\lambda}} : \mathcal{X} \rightarrow \mathcal{Y}$ that is governed by parameters $\boldsymbol{\lambda}$. Given feature functions $\{f_j(\mathbf{y}, \mathbf{x})\}_{j=1}^{J}$, 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 [31].

Latent Variable Models. To increase the representational power of a model, a common approach is to introduce latent (or hidden) variables $\mathbf{h} \in \mathcal{H} = \{1, \ldots, H\}^{J}$. 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. The former strategy is employed by latent structural SVMs [35], while the latter is employed by hidden CRF models [24]. A topic of ongoing investigation is the benefits of each, and alternative strategies that interpolate between the two [20].

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 [10], co-occurrences of labels in semantic image segmentation [14], and cardinality-based potentials [33, 34]. 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, e.g., modeling star

convexity [5], connectivity [32, 23], 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 tradeoff 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 [11, 26, 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) [28] 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 $\mathbf{v} = (v_1, \ldots, v_I)^\top$ that represent the observations, or data; and (2) the J *hidden* or latent units $\mathbf{h} = (h_1, \ldots, h_J)^\top$ 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:

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{v}^\top \mathbf{W}\mathbf{h} - \mathbf{b}^\top \mathbf{v} - \mathbf{c}^\top \mathbf{h}, \qquad (1)$$

where $\mathbf{W} \in \mathbb{R}^{I \times J}$ encodes the hidden-visible interactions, $\mathbf{b} \in \mathbb{R}^I$ are the input biases, and $\mathbf{c} \in \mathbb{R}^J$ are the hidden biases. The energy function specifies the probability distribution over the joint space $(\mathbf{v}, \mathbf{h})$ via the Boltzmann distribution

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h})) \qquad (2)$$

with the partition function Z given by $\sum_{\mathbf{v},\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}))$. 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 J binary hidden units, this takes on a particular nice form:

$$\begin{aligned} p(\mathbf{v}) &= \sum_{\mathbf{h}} \frac{1}{Z} \exp\left(\mathbf{v}^\top \mathbf{W}\mathbf{h} + \mathbf{b}^\top \mathbf{v} + \mathbf{c}^\top \mathbf{h}\right) \\ &= \frac{1}{Z} \exp\left(\mathbf{b}^\top \mathbf{v} + \sum_{j=1}^{J} \log\left(1 + \exp(\mathbf{v}^\top \mathbf{w}_j + c_j)\right)\right), \end{aligned} \qquad (3)$$

where 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) [8].

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 unsupervised training paradigm. For example, they have been used to model object shape [2], images under occlusion [18], and noisy images [30]. 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 [7]. That work did not address shape information as we do, and it also combined the RBM with a very restricted form of CRF. [21] also tried to use RBMs for structured output problems, but there are no pairwise connections between labels, and the actual loss was not considered during training. Also related is the work of [3], which uses a generative framework to model labels and images.

3. Equating Pattern Potentials and RBMs

In [26], the basic pattern potential is defined as $g(\mathbf{y}) = \min\{d(\mathbf{y}) + \theta_0, \theta_{\max}\}$, where θ_0 and $\theta_{\max}$ are constants, $d(\mathbf{y}) = \sum_i w_i y_i + K$ is a deviation cost for $\mathbf{y}$ to deviate from a specific pattern $\mathbf{Y}$, and 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 $\mathbf{Y}_i = 1$. Rother et al. propose two methods for compositing several of these potentials. In the "sum" case, they sum over these potentials, and in the "min" case they minimize, yielding a potential of the form $g(\mathbf{y}) = \min\{d_1(\mathbf{y}) + \theta_1, ..., d_J(\mathbf{y}) + \theta_J\}$, where we assume one deviation function is constant (i.e. all w_i are 0). In the supplementary material, we give more details and show that the "sum" case is equivalent to the free energy that arises after summing out hidden units in an RBM, up to the approximation that $\min\{x, 0\} \approx -\log(1 + \exp(-x))$ (alternatively, they are exactly equivalent if hidden units are maximized out). In the same sense, the "min" case is equivalent to an RBM with a constraint that only one hidden unit can be active. See Table 1 in the supplementary material.

4. The CHOPP-Augmented CRF

Understanding the equivalence between RBMs and pattern potentials leads us to define a more general potential,

$$f(\mathbf{y};T) = -T\log\left(\sum_{\mathbf{h}}\exp\left(\frac{1}{T}(\mathbf{y}^\top\mathbf{W}\mathbf{h}+\mathbf{c}^\top\mathbf{h})\right)\right), \quad (4)$$

where T is a temparature parameter. Setting $T=1$ gives the exact RBM high order potential (softplus functions in Eq. 3). If there is no constraint on hidden variables, setting $T\to 0$ gives the "sum" compositional high order pattern potential. If we restrict hidden variables to have a 1-of-J constraint, setting $T\to 0$ then gives the "min" compositional high order pattern potential.

Therefore the CHOPP permits manipulations along two separate axes: the temperature T interpolates the RBM and pattern potential, while the constraints on hidden variable activities interpolates the "sum" and "min" composition strategies. [6, 27] used similar techniques to interpolate "sum" and "min". A longer exposition of these claims is given in the supplementary material.

In this section, we augment a standard pairwise CRF with the CHOPP 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 constraint on hidden variables, e.g. using cardinality potentials [29].

4.1. Model

The conditional distribution of a labeling $\mathbf{y}$ given input image $\mathbf{x}$ is defined as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\exp\left\{\lambda^u\sum_{i=1}^{I}f_i(y_i|\mathbf{x}) + \sum_k\lambda_k^p\sum_{i,j}f_{ij}^k(y_i,y_j|\mathbf{x}) + \mathbf{b}^\top\mathbf{y} + T\log\left(\sum_{\mathbf{h}}\exp\left(\frac{1}{T}(\mathbf{y}^\top\mathbf{W}\mathbf{h}+\mathbf{c}^\top\mathbf{h})\right)\right)\right\} \quad (5)$$

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 $\mathbf{W}$, $\mathbf{b}$, $\mathbf{c}$ are RBM parameters. To simplify notation, for a given $\mathbf{x}$ we use shorthand $\psi^u(\mathbf{y}) = \lambda^u\sum_i f_i(y_i|\mathbf{x})$ for unary potentials and $\psi^p(\mathbf{y}) = \sum_k\lambda_k^p\sum_{i,j}f_{ij}^k(y_i,y_j|\mathbf{x})$ for pairwise potentials.

$T=1$ **Special Case.** For the special case $T=1$, the posterior distribution $p(\mathbf{y}|\mathbf{x})$ is equivalent to a joint distribution over $\mathbf{y}$ and $\mathbf{h}$, with $\mathbf{h}$ summed out

$$p(\mathbf{y},\mathbf{h}|\mathbf{x}) \propto \exp\left(\psi^u(\mathbf{y}) + \psi^p(\mathbf{y}) + \mathbf{y}^\top\mathbf{W}\mathbf{h} + \mathbf{b}^\top\mathbf{y} + \mathbf{c}^\top\mathbf{h}\right). \quad (6)$$

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

$$p(h_j=1|\mathbf{y},\mathbf{x}) = \sigma\left(c_j + \sum_{i=1}^{I}w_{ij}y_i\right), \quad (7)$$

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

$$p(\mathbf{y}|\mathbf{h},\mathbf{x}) \propto \exp\left((\mathbf{b}+\mathbf{W}\mathbf{h})^\top\mathbf{y} + \psi^u(\mathbf{y}) + \psi^p(\mathbf{y})\right), \quad (8)$$

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

Model Variants. One way to make this model even more expressive is to allow CHOPP terms to depend 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.

Another variant of the current formulation is to make the CHOPPs *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, 22]. These can be trained by tying 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. MAP Inference

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

$$-f(\mathbf{y};T) \geq (\mathbf{c}+\mathbf{W}^\top\mathbf{y})^\top\mathbb{E}_q[\mathbf{h}] + H(q), \quad (9)$$

where $q(\mathbf{h})$ is any distribution of $\mathbf{h}$, $H(q)$ is the entropy of q. Note the temperature T canceled out. The difference between the left and right side is the KL divergence between q and p^* where

$$p^*(\mathbf{h}|\mathbf{y}) = \frac{\exp\left(\frac{1}{T}(\mathbf{c}+\mathbf{W}^\top\mathbf{y})^\top\mathbf{h}\right)}{\sum_{\mathbf{h}}\exp\left(\frac{1}{T}(\mathbf{c}+\mathbf{W}^\top\mathbf{y})^\top\mathbf{h}\right)}. \quad (10)$$

When there is no constraint on $\mathbf{h}$, this is also a factorial distribution. Therefore

$$\log p(\mathbf{y}|\mathbf{x}) \geq \psi^u(\mathbf{y}) + \psi^p(\mathbf{y}) + \mathbf{b}^\top\mathbf{y} + (\mathbf{c}+\mathbf{W}^\top\mathbf{y})^\top\mathbb{E}_q[\mathbf{h}] + H(q) + \text{const.} \quad (11)$$

We can use the EM algorithm to optimize this lower bound. 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 $q = p^*$. When $T = 1$ this becomes Eq. 7; when $T \to 0$, it puts all the mass on one configuration of $\mathbf{h}$, i.e. minimizes the energy over hidden units.

In the M step, we fix q and find the $\mathbf{y}$ that maximizes the bound. The relevant terms are

$$\left(\mathbf{b} + \mathbf{W}\mathbb{E}_q[\mathbf{h}]\right)^\top \mathbf{y} + \psi^u(\mathbf{y}) + \psi^p(\mathbf{y}), \tag{12}$$

which 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 $\mathbf{y}$; we use graph cuts. 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.

Note that this is not the usual EM algorithm used for learning parameters in latent variable models. Here all parameters are fixed and we use the EM algorithm to make predictions.

Remark. When there is no sparsity constraint on $\mathbf{h}$, it is possible to analytically sum out the hidden variables, which leads to a collapsed energy function with J high order factors, one for each original hidden unit. It is then possible to develop a linear program relaxation-based inference routine that operates directly on the high order model. We did this but found its performance inferior to the above EM procedure. More details are in the supplementary materials.

4.3. Learning

Here we fix the unary and pairwise potentials and focus on learning the parameters in the CHOPP.

For the $T = 1$ case, we can use Contrastive Divergence (CD) [8] to approximately maximize the conditional likelihood of data under our model, which is standard for learning RBMs. 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), 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 which we believe allows the model to more globally learn the distribution. 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 γ, which can be $\mathbf{b}$, $\mathbf{c}$ or $\mathbf{W}$ ($\mathbf{c}_0$ and $\mathbf{W}_0$ as well if we use the conditioned CHOPPs), we get

$$\frac{\partial L}{\partial \gamma} = \mathbb{E}_{\mathbf{y}}\left[\left(\ell(\mathbf{y}, \mathbf{y}^*) - \mathbb{E}_{\mathbf{y}}[\ell(\mathbf{y}, \mathbf{y}^*)]\right)\left[-\frac{\partial E(\mathbf{y})}{\partial \gamma}\right]\right], \tag{13}$$

where $\mathbb{E}_{\mathbf{y}}[.]$ is the expectation under $p(\mathbf{y}|\mathbf{x})$, $E(\mathbf{y})$ is the energy of CHOPP-augmented CRF and we have

$$-\frac{\partial E(\mathbf{y})}{\partial \gamma} = \mathbb{E}_{p^*(\mathbf{h}|\mathbf{y})}\left[-\frac{\partial E(\mathbf{y}, \mathbf{h})}{\partial \gamma}\right]. \tag{14}$$

$p^*(\mathbf{h}|\mathbf{y})$ is given in Eq. 10 and $E(\mathbf{y}, \mathbf{h})$ is the standard RBM energy in Eq. 1 with $\mathbf{v}$ substituted by $\mathbf{y}$.

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

$$\frac{\partial L}{\partial \gamma} \approx \frac{1}{N-1}\sum_{n=1}^{N}\left(\ell(\mathbf{y}^n, \mathbf{y}^*) - \frac{1}{N}\sum_{n'=1}^{N}\ell(\mathbf{y}^{n'}, \mathbf{y}^*)\right)\left[-\frac{\partial E(\mathbf{y}^n)}{\partial \gamma}\right]. \tag{15}$$

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.

For the $T = 1$ case, we sample from the joint distribution $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ using standard block Gibbs sampling and discard $\mathbf{h}$ to get samples from $p(\mathbf{y}|\mathbf{x})$. We also use several persistent Markov chains for each image to generate samples, where in the first iteration of learning each chain is initialized at the same initialization as is used for inference. The model parameters are updated after every sampling step.

For the other choices of T, it is not easy to get samples from $p(\mathbf{y}|\mathbf{x})$, but we can sample from $p^*(\mathbf{h}|\mathbf{y})$ and $p(\mathbf{y}|\mathbf{h}, \mathbf{x})$ alternatively, as if we are running block Gibbs sampling. The properties of the samples that result from this procedure is an interesting topic for future research.

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. We trained a 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, where the first is image-independent: (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]; (3) Set p_{ij} to represent higher level boundary information given by Pb boundary detector [19], 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 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 validation set and report its performance on test set.

For all experiments, we always set $T = 1$, so CHOPP becomes equivalent to an RBM high order potential.

In minimum expected loss learning, we use 2 persistent sampling chains for each image, generate 1 sample from each chain, divide $(\mathbf{y}, \mathbf{h})$ into 3 blocks of variables ($\mathbf{h}$ and 2 $\mathbf{y}$ blocks for the 4-connected grid structure) and update parameters after every full block Gibbs sampling pass.

Additional results are in the supplementary material.

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 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. 2 (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 segmentation data [4] to construct a bird and a person data set. For these, we took all bounding boxes containing the target class and created a binary segmentation inside 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 32×32. This gives us a set of silhouettes that preserve the challenging aspects of modeling shape in a realistic structured output setting.

The two PASCAL datasets are challenging due to variability in the images and segmentations, while the number of images is quite small, 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 examples for each original bounding box by randomly shifting coordinates by a small amount. We also mirrored all images and segmentations. This augmentation gives us 11 times more data.

For each data set, we then evaluated variability using a measure inspired by the learning procedure suggested by Rother et al. [26]. 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_i^k, the probability that pixel i is assigned label 1 given that it comes from an instance in cluster k. Now define the within cluster average entropy $H^k = -\frac{1}{D_v}\sum_i \left(q_i^k \log q_i^k + (1 - q_i^k)\log(1 - q_i^k)\right)$. 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. See Fig. 2.

5.2. Performance vs. Variability

Next we report results for a pre-trained RBM model added to a standard CRF (denoted RBM), where 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, hence denoted iPW. Fig. 3 show 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 Bird 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. Joint learning with standard contrastive divergence on the Horse data led to poor performance, as the objective was unstable in the first few iterations and then steadily got worse during

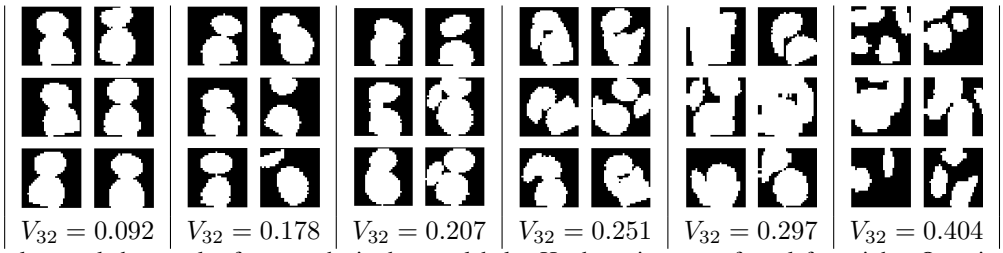

Figure 2. Randomly sampled examples from synthetic data set labels. Hardness increases from left to right. Quantitative measures of variability using $K = 32$ are reported in the bottom row. Variabilities of Horse, Bird, and Person data sets are 0.176, 0.370, and 0.413.

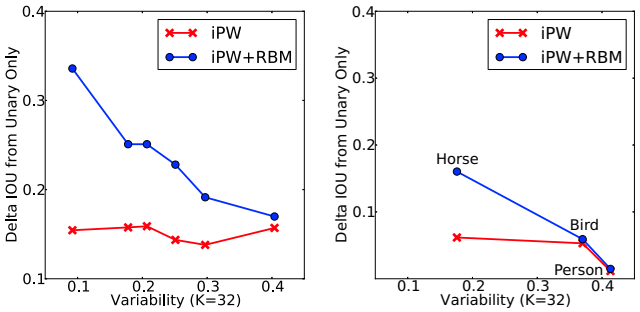

Figure 3. Results on (left) synthetic and (right) real data showing test intersection-over-union scores as a function of data set variability. The y-axis is difference relative to a Unary Only model. All models here and below utilize unary potentials. Note that these results utilize a pretrained RBM model.

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**

Table 1. Expected loss test results. RBM is a pretrained RBM. jRBM is jointly trained using expected loss.

training. So here we focus on the expected loss training described in Section 4.3, and denote the resulting models as jRBM to indicate joint training. Results comparing these approaches on the three real data sets are given in Tab. 1, with Unary and iPW results given as baselines. 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. As inputs we use the learned unary potentials and the response of the Pb boundary detector [19], both downsampled to be of size 16×16. We learned the internal parameters of this ijRBM model, the image-dependent jointly trained RBM, using the intersection-over-union expected loss as this gave the best results in the previous experiment. Results are shown in Tab. 2. For comparison, we also train Unary+Pairwise models with an image-independent pairwise potentials (PW) and an image-dependent pairwise potentials (iPW). In the Bird data, we see that the image-specific information helps the ijRBM

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**

Table 2. 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.

similarly as to 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, which we attribute to the fact that learning methods for convolutional RBMs are not nearly as evolved as methods for learning ordinary RBMs. We provide more details in the supplementary material.

Composition Schemes. We qualitatively compare patterns learned by the "min" composition approach presented in [26] 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. 1 that the learned filters are very different. As the variability of the data grows, we expect the utility of the "sum" composition scheme to increase.

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 [25]. 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 Section 3 that this sparsity can be used to control the type of compositionality employed by the model. An interesting direction for future work is exploring sparse variants of RBMs, which sit between these two extremes, and other forms of structure over latent variables like in deep models.

References

[1] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV*, 2001. 2, 6

[2] S. M. A. Eslami, N. Heess, and J. Winn. The shape Boltzmann machine: a strong model of object shape. In *CVPR*, 2012. 3

[3] S. M. A. Eslami and C. Williams. A generative model for parts-based object segmentation. In *NIPS*, pages 100–107, 2012. 3

[4] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 2, 6

[5] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR*, 2010. 3

[6] T. Hazan and R. Urtasun. A primal-dual message-passing algorithm for approximated large scale structured prediction. In *NIPS*, 2010. 4

[7] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale conditional random fields for image labelling. In *CVPR*, 2004. 3

[8] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 2002. 3, 5

[9] J. Kivinen and C. Williams. Multiple texture Boltzmann machines. In *AISTATS*, volume 22, 2012. 2

[10] P. Kohli, L. Ladickỳ, and P. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 82(3), 2009. 1, 2

[11] N. Komodakis. Efficient training for pairwise or higher order CRFs via dual decomposition. In *CVPR*, 2011. 3

[12] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *CVPR*, 2009. 2

[13] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *NIPS*, 2012. 1

[14] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Inference methods for CRFs with co-occurrence statistics. *IJCV*, 2011. 2, 3

[15] H. Larochelle and Y. Bengio. Classification using discriminative restricted Boltzmann machines. In *ICML*, 2008. 3

[16] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, 2009. 4

[17] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009. 3

[18] N. LeRoux, N. Heess, J. Shotton, and J. Winn. Learning a generative model of images by factoring appearance and shape. *Neural Computation*, 2011. 3

[19] M. Maire, P. Arbeláez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, 2008. 2, 6, 7

[20] K. Miller, M. Kumar, B. Packer, D. Goodman, and D. Koller. Max-margin min-entropy models. In *AISTATS*, 2012. 2

[21] V. Mnih, H. Larochelle, and G. Hinton. Conditional restricted Boltzmann machines for structured output prediction. In *UAI*, 2011. 3

[22] M. Norouzi, M. Ranjbar, and G. Mori. Stacks of convolutional restricted Boltzmann machines for shift-invariant feature learning. In *CVPR*, 2009. 4

[23] S. Nowozin and C. Lampert. Global connectivity potentials for random field models. In *CVPR*, 2009. 3

[24] A. Quattoni, S. Wang, L. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *PAMI*, 2007. 2

[25] M. R. and G. Hinton. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural Computation*, 2010. 8

[26] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *CVPR*, 2009. 1, 2, 3, 6, 7

[27] A. Shekhovtsov, P. Kohli, and C. Rother. Curvature prior for MRF-based segmentation and shape inpainting. In *Pattern Recognition*, pages 41–51. Springer, 2012. 4

[28] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In *Parallel distributed processing*. 1986. 3

[29] K. Swersky, D. Tarlow, I. Sutskever, R. Salakhutdinov, R. Zemel, and R. Adams. Cardinality restricted Boltzmann machines. In *NIPS*, 2012. 4

[30] Y. Tang, R. Salakhutdinov, and G. Hinton. Robust Boltzmann machines for recognition and denoising. In *CVPR*, 2012. 3

[31] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004. 2

[32] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008. 3

[33] S. Vicente, V. Kolmogorov, and C. Rother. Joint optimization of segmentation and appearance models. In *ICCV*. 2009. 2

[34] O. Woodford, C. Rother, and V. Kolmogorov. A global perspective on MAP inference for low-level vision. In *IJCV*, 2009. 2

[35] C. Yu and T. Joachims. Learning structural SVMs with latent variables. In *ICML*, 2009. 2