Local Perturb-and-MAP for Structured Prediction

Gedas Bertasius
University of Pennsylvania

Qiang Liu
Dartmouth College

Lorenzo Torresani
Dartmouth College

Jianbo Shi
University of Pennsylvania

Abstract

Conditional random fields (CRFs) provide a powerful tool for structured prediction, but cast significant challenges in both the learning and inference steps. Approximation techniques are widely used in both steps, which should be considered jointly to guarantee good performance (a.k.a. "inferning"). Perturb-and-MAP models provide a promising alternative to CRFs, but require global combinatorial optimization and hence they are usable only on specific models. In this work, we present a new Local Perturb-and-MAP (locPMAP) framework that replaces the global optimization with a local optimization by exploiting our observed connection between locPMAP and the pseudolikelihood of the original CRF model. We test our approach on three different vision tasks and show that our method achieves consistently improved performance over other approximate inference techniques optimized to a pseudolikelihood objective. Additionally, we demonstrate that we can integrate our method in the fully convolutional network framework to increase our model's complexity. Finally, our observed connection between locPMAP and the pseudolikelihood leads to a novel perspective for understanding and using pseudolikelihood.

1 Introduction

Probabilistic graphical models, such as Markov random fields, and conditional random fields (Lafferty et al., 2001) provide a powerful framework for solving challenging learning problems that require structured output prediction (Lauritzen, 1996). The use of graphical models consists of two main steps: *learning*, which estimates the model parameters from the data, as well as *inference*, which makes predictions based on the learned model.

Unfortunately, both maximum likelihood learning and probabilistic inference involve calculating the normalization constant (i.e. a partition function), which is intractable to compute in general. In practice, approximation methods, such as variational inference and MCMC, are widely used for inference and learning. There also exist other consistent learning methods such as the maximum pseudolikelihood (PL) estimator Besag (1975) which does not require calculating likelihood and is computationally tractable.

It is well known that there are strong interactions between learning and inference. As a result, the choice of the learning and inference algorithms should be considered jointly, an idea which is referred to as "inferning".[1] Although it is relatively easy to identify "inferning" pairs when using variational or MCMC approximations, it is unclear what the natural inference counterpart of pseudolikelihood (PL) is. For instance, even if PL learning estimates a true model, a poor choice of a subsequent approximate inference algorithm may deteriorate the overall prediction accuracy.

An alternative way to achieve "inferning" is to employ models that are computationally more tractable. One such framework is the *Perturb-and-MAP* model (Papandreou & Yuille, 2011; Tarlow et al., 2012; Hazan et al., 2013) which involves injecting noise into the log-probability (the potential function), and generating random samples to find the global maximum, i.e., the maximum *a posteriori* (MAP) estimate of the perturbed probability. These models have a sound probabilistic interpretation, which can be exploited to make predictions with an uncertainty measure. Another benefit is that models from the *Perturb-and-MAP* class require combinatorial optimization, which is easier to solve than in the case of probabilistic inference models, which instead require marginalizing over variables or drawing MCMC samples.

Unfortunately, despite being easier than marginalization, MAP estimation still requires a global optimization that is

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

[1] see `http://inferning.cs.umass.edu`.

generally NP-hard. Thus, this prevents the use of Perturb-and-MAP for generic graphical models. In addition, the Perturb-and-MAP model can be viewed as a hidden variable model with deterministic constraints, and its training casts another challenging learning task that involves maximizing a non-convex likelihood function. This is often solved using variants of EM, combined with approximation schemes (Gane et al., 2014; Tarlow et al., 2012).

In this work, we propose "Local Perturb-and-MAP" (locPMAP), a model that only requires finding a local maximum of the perturbed potential function, which is much easier than global optimization required for Global-MAP methods. Our locPMAP model has a close connection with the classical pseudolikelihood, in that pseudolikelihood can be interpreted as a type of partial information likelihood of our locPMAP model. This motivates us to decode pseudolikelihood training with a locPMAP inference procedure. We test our approach on three different vision tasks, where we show that locPMAP applied to models learned with PL yields consistently better inference results than those achieved using other approximate inference techniques.

In addition, we demonstrate that we can integrate our method in the fully convolutional network framework Long et al. (2015) to address the small complexity limitation of log-linear CRF models and improve performance on challenging structured prediction problems. Finally, our approach provides a novel view for pseudolikelihood and opens up opportunities for many useful extensions.

2 Related Work

The idea of Perturb-and-MAP was motivated by the classical "Gumbel-Max trick" that connects the logistic function with discrete choice theory (McFadden et al., 1973; Yellott, 1977). It was first applied to graphical model settings by Papandreou & Yuille (2011) and Hazan & Jaakkola (2012). These studies gave rise to a rich line of research (see e.g., Hazan et al., 2013; Gane et al., 2014; Tarlow et al., 2012). We remark that these methods all require global optimization, in contrast with the local optimization in our method.

The idea of "inferning," which enforces the consistency between learning and inference, was probably first discussed by Wainwright (2006), who showed that when exact inference is intractable, it is better to learn a "wrong" model to compensate the errors made by the subsequent approximate inference methods. Empirical analysis of influence of learning and inference procedures can also be found in Sutton & McCallum (2009); Gelfand (2014); Xiang & Neville. A line of work has been developed to explicitly tune parameters in approximate inference procedures (Meshi et al., 2010; Stoyanov & Eisner, 2012; Domke, 2013). In addition, Srivastava et al. proposed an approximate inference method that interacts with learning in order to train Deep Boltzman Machines more efficiently. It is also relevant to mention (Poon & Domingos, 2011), which provides another class of models that enables efficient inference.

3 Background

In subsection 3.1, we introduce some background information on conditional random fields (CRFs). Additionally, in subsection 3.2, we present some key ideas related to the Gumbel-Max trick and Perturb-and-MAP. We will use all of these ideas to introduce our method in Section 4.

3.1 Structured Prediction with CRFs

CRFs Lafferty et al. (2001) provide a framework to solve challenging structured prediction problems. Let $\boldsymbol{x}$ be an input (e.g., an image), and $\boldsymbol{y} \in \mathcal{Y}$ a set of structured labels (e.g., a semantic segmentation). A CRF assumes that the labels $\boldsymbol{y}$ are drawn from an exponential family distribution

$$p(\boldsymbol{y}|\boldsymbol{x};\ w) = \frac{1}{Z(\theta)} \exp(\theta(\boldsymbol{y}, \boldsymbol{x}, w)) \tag{1}$$

where θ is a potential function and w are model parameters that need to be estimated from data; the normalization constant $Z(\boldsymbol{x}, w) = \sum_{\boldsymbol{y}} \exp(\theta(\boldsymbol{y}, \boldsymbol{x}, w)$ is difficult to compute unless the corresponding graph is tree-structured.

Now let us assume that we are given a set of labeled training examples $\{\boldsymbol{x}^i, \boldsymbol{y}^i\}$. A typical maximum likelihood estimator learns parameters w by maximizing the log likelihood function:

$$\hat{w} = \arg\max_{w} \sum_{i} \log p(\boldsymbol{y}^i|\boldsymbol{x}^i; w). \tag{2}$$

With the estimated parameters $\hat{w}$ we can then make predictions for a new testing image $\boldsymbol{x}^*$:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x}^*; \hat{w}) = \arg\max_{\boldsymbol{y}} \theta(\boldsymbol{y}, \boldsymbol{x}^*; \hat{w}). \tag{3}$$

However, both the learning and inference steps in Equations (2)-(3) are computationally intractable for general loopy graphs. Instead, a popular computationally-efficient alternative for MLE is the pseduolikelihood (PL) estimator (Besag, 1975), defined as:

$$\hat{w} = \arg\max_{w} \sum_{i} \sum_{j} \log p(y_j^i|\boldsymbol{x}^i, \boldsymbol{y}_{\neg j}^i; w), \tag{4}$$

where $\neg j$ refers to the neighborhood of the nodes in the graph that are connected to the node j. Based on this formulation, each conditional likelihood does not involve Z, and can be calculated efficiently. Besag (1975) showed that PL is an asymptotically consistent estimator, meaning that $\hat{w}$ approaches the true parameter w as the size of the dataset is increased.

However, even if PL estimates the true parameters perfectly, the prediction step in (3) still requires approximation. Iterated Conditional Modes (ICM) is one of the simplest inference algorithm that returns a local maximum from the neighborhood of nodes. Other widely used inference techniques include loopy belief propagation (LBP), the mean field algorithm (MF), and Gibbs sampling.

The problem with using these approximate inference algorithms is that they may not work well together with the PL learning algorithm. This is because learning and inference are performed disjointly without considering how one may affect the other. Much of the difficulty comes from the fact that the definition of PL in (4) is not "generative", since the $\boldsymbol{y}^i$ depend on each other in a loopy fashion.

3.2 Perturb-and-MAP

In contrast with the CRF defined in (1), the Perturb-and-MAP model (Hazan & Jaakkola, 2012; Papandreou & Yuille, 2011) considers distributions of the form

$$\Pr[\boldsymbol{y} \in \arg\max\{\theta(\boldsymbol{y}) + \epsilon(\boldsymbol{y})\}], \quad \epsilon \sim q \tag{5}$$

where we dropped the dependency on $\boldsymbol{x}$ to simplify the notation. That is, we first perturb the potential function $\theta(\boldsymbol{y})$ with random noise $\epsilon(\boldsymbol{y})$ from distribution q and then draw the sample by finding the maximum point $\boldsymbol{y}$. Consider special perturbation noise $\epsilon(\boldsymbol{y})$ drawn i.i.d. from the zero mean Gumbel distribution with cumulative distribution function $F(t) = \exp(-\exp(-(t+c))$, where c is the Euler constant. The Gumbel-Max trick (Yellott, 1977; McFadden et al., 1973) shows that the Perturb-and-MAP model is then equivalent to the distribution in (1), that is,

$$\Pr[\boldsymbol{y} \in \arg\max\{\theta(\boldsymbol{y}) + \epsilon(\boldsymbol{y})\}] = \frac{\exp(\theta(\boldsymbol{y}))}{\sum_{\boldsymbol{y}} \exp(\theta(\boldsymbol{y})))}. \tag{6}$$

This connection provides a basic justification for Perturb-and-MAP models. It is also possible to use more general perturbations beyond the Gumbel perturbation, but then the training of the Perturb-and-MAP model becomes substantially more difficult, requiring EM-type non-convex optimization with Monte Carlo or other approximations (Tarlow et al., 2012; Hazan et al., 2013; Gane et al., 2014).

4 Local Perturb-and-MAP Optimization

A major limitation of Perturb-and-MAP, even when using Gumbel noise, is that it requires global optimization over the perturbed potentials. We address this problem by replacing the global optimum with a local optimum. We start by defining the notion of local optimality.

Definition 4.1. *Let $\mathcal{B} = \{\beta_k\}$ be a set of non-overlapping sets of variable indices such that $\beta_k \cap \beta_l = \emptyset$ for $\forall\ k \neq l$. Then we say that*

$$\boldsymbol{y} \in \mathrm{Loc}\big[\theta(\boldsymbol{y});\ \mathcal{B}\big]$$

if $\boldsymbol{y}_\beta \in \arg\max_{\boldsymbol{y}'_\beta} \big[\theta(\boldsymbol{y}'_\beta, \boldsymbol{y}_{\neg\beta})\big]$ for $\forall \beta \in \mathcal{B}$, where $\neg\beta = [p] \setminus \beta$. This implies that $[\boldsymbol{y}_\beta, \boldsymbol{y}_{\neg\beta}]$ is no worse than $[\boldsymbol{y}'_\beta, \boldsymbol{y}_{\neg\beta}]$ for any $\boldsymbol{y}'_\beta \in Y_\beta$, $\beta \in \mathcal{B}$. In other words, $\boldsymbol{y}$ is a block-coordinate-wise maximum of $\theta(\boldsymbol{y})$ on the set $\mathcal{B}$.

We are now ready to establish our main result. We show that by exploiting random Gumbel perturbations over the potential functions, we can formulate a Local Perturb-and-MAP model, which yields a close connection with pseudo-likelihood.

Theorem 4.2. *Let us now perturb $\theta(\boldsymbol{y})$ to get $\tilde{\theta}(\boldsymbol{y}) = \theta(\boldsymbol{y}) + \sum_{\beta\in\mathcal{B}} \epsilon(\boldsymbol{y}_\beta)$, where each element $\epsilon(\boldsymbol{y}_\beta)$ is drawn i.i.d. from a Gumbel distribution with CDF $F(t) = \exp(-\exp(-(t+c)))$, where c is the Euler constant. Then we have, $\forall \boldsymbol{y}$,*

$$\Pr\Big(\boldsymbol{y} \in \mathrm{Loc}\big[\tilde{\theta}(\boldsymbol{y});\ \mathcal{B}\big]\Big) = \prod_{\beta\in\mathcal{B}} p(\boldsymbol{y}_\beta|\boldsymbol{y}_{\neg\beta};\ \theta).$$

Note that the right hand side has a form of composite likelihood (Lindsay, 1988), which reduces to the pseudolikelihood when taking $\mathcal{B} = \{k\colon k \in [n]\}$.

Proof. Note that based on our definition of $\mathrm{Loc}()$ we can write $\mathrm{Loc}\big[\tilde{\theta}(\boldsymbol{y}); \mathcal{B}\big] = \cap_{\beta\in\mathcal{B}} A_\beta$, where $A_\beta = \{\boldsymbol{y}\colon \boldsymbol{y}_\beta \in \arg\max_{\boldsymbol{y}_\beta} \big[\tilde{\theta}(\boldsymbol{y}_\beta,\ \boldsymbol{y}_{\neg\beta})\big]\}$. Then, we can write:

$$\begin{aligned}
\Pr(\boldsymbol{y} \in A_\beta) &= \Pr(\boldsymbol{y}_\beta \in \arg\max_{\boldsymbol{y}_\beta} \big[\theta(\boldsymbol{y}_\beta,\ \boldsymbol{y}_{\neg\beta}) + \epsilon(\boldsymbol{y}_\beta)\big]) \\
&= \frac{\exp(\theta(\boldsymbol{y}_\beta, \boldsymbol{y}_{\neg\beta}))}{\sum_{\boldsymbol{y}'_\beta} \exp(\theta(\boldsymbol{y}'_\beta, \boldsymbol{y}_{\neg\beta})} \\
&= p(\boldsymbol{y}_\beta|\boldsymbol{y}_{\neg\beta}; \theta),
\end{aligned}$$

where we use Equation 6 to derive these equalities. Note that Equation 6 results from the application of the Gumbel-Max trick. In the context of our problem, this equation holds because $\epsilon(\boldsymbol{y}_\beta)$ are drawn i.i.d. from a zero mean and a unit variance Gumbel distribution. Additionally, since $\epsilon(\boldsymbol{y}_\beta)$ are drawn independently from each other, the events $[\boldsymbol{y} \in A_\beta]$ are independent too. Therefore, we can write:

$$\begin{aligned}
\Pr(\boldsymbol{y} \in \mathrm{Loc}\big[\tilde{\theta}(\boldsymbol{y});\ \mathcal{B}\big]) &= \Pr(\boldsymbol{y} \in \cap_{\beta\in\mathcal{B}} A_\beta) \\
&= \prod_{\beta\in\mathcal{B}} p(\boldsymbol{y}_\beta|\boldsymbol{y}_{\neg\beta};\ \theta).
\end{aligned}$$

□

Our locPMAP model defines a procedure for generating *random subsets* $\mathrm{Loc}\big[\tilde{\theta}(\boldsymbol{y});\ \mathcal{B}\big]$ formed by the local maxima of the random function $\tilde{\theta}(\boldsymbol{y})$. Theorem 4.2 suggests that for any given (deterministic) configuration $\boldsymbol{y}$, the probability that $\boldsymbol{y}$ is an element of $\mathrm{Loc}\big[\tilde{\theta}(\boldsymbol{y});\ \mathcal{B}\big]$ equals the composite likelihood $\ell_{\mathcal{B}}(\boldsymbol{y};\ \theta) := \prod_{\beta\in\mathcal{B}} p(\boldsymbol{y}_\beta|\boldsymbol{y}_{\neg\beta};\ \theta)$. Here the

point $\boldsymbol{y}$ is deterministic, while the set $\mathrm{Loc}\big[\tilde{\theta}(\boldsymbol{y});\ \mathcal{B}\big]$ is random, similar to the case of confidence intervals in statistics.

We should point out that $\ell_{\mathcal{B}}(\boldsymbol{y};\ \theta)$ is not a properly normalized distribution over $\boldsymbol{y} \in \mathcal{Y}$, because there may be multiple local maxima in each random set $\mathrm{Loc}\big[\tilde{\theta}(\boldsymbol{y});\ \mathcal{B}\big]$. In fact, it is easy to see that the expected number $Z_{\mathcal{B}}$ of local maxima of $\tilde{\theta}(\boldsymbol{y})$ is

$$Z_{\mathcal{B}} \stackrel{def}{=} \mathbb{E}(\big|\mathrm{Loc}\big[\tilde{\theta}(\boldsymbol{y});\ \mathcal{B}\big]\big|) = \sum_{\boldsymbol{y}\in\mathcal{Y}} \prod_{\beta\in\mathcal{B}} p(\boldsymbol{y}_\beta|\boldsymbol{y}_{\neg\beta};\ \theta). \tag{7}$$

This can be used to define a normalized probability over $\mathcal{Y}$ via $\ell_{\mathcal{B}}(\boldsymbol{y};\ \theta)/Z_{\mathcal{B}}$, which, however, is computationally intractable due to the difficulty for computing $Z_{\mathcal{B}}$.

It is interesting to draw a comparison with the global Perturb-and-MAP model when $\mathcal{B}$ includes only the global set of all the elements of $\boldsymbol{y}$, in which case we can show that $Z_{\mathcal{B}} = \mathbb{E}(\big|\mathrm{Loc}\big[\tilde{\theta}(\boldsymbol{y});\ \mathcal{B}\big]\big|) = 1$ in (7). Because there always exists at least one optimum point, we must always have $\big|\mathrm{Loc}\big[\tilde{\theta}(\boldsymbol{y});\ \mathcal{B}\big]\big| \geq 1$. Therefore, we have $\big|\mathrm{Loc}\big[\tilde{\theta}(\boldsymbol{y});\ \mathcal{B}\big]\big| = 1$ with probability 1 in this case, that is, there only exists one unique global optimum point.

locPMAP and Pseudolikelihood In practice, we can not exactly enumerate, nor observe the whole set $\mathcal{L} = \mathrm{Loc}\big[\tilde{\theta}(\boldsymbol{y});\ \mathcal{B}\big]$. We instead observe a single point $\boldsymbol{y}$ which we can assume to belong to $\mathcal{L}$. Without further assumption on how $\boldsymbol{y}$ is selected from $\mathcal{L}$, the only information available through observing a point $\boldsymbol{y}$ is $\Pr(\boldsymbol{y} \in \mathcal{L})$. As a result, given a set of i.i.d. observation $\{\boldsymbol{y}_i\}$, it is natural to maximize their overall observation likelihood:

$$\begin{aligned}\hat{w} &= \arg\max_w \sum_i \log p(\boldsymbol{y}^i \in \mathrm{Loc}\big[\tilde{\theta}(\boldsymbol{y};\ w);\ \mathcal{B}\big]) \\ &= \arg\max_w \sum_i \sum_{\beta\in\mathcal{B}} \log p(\boldsymbol{y}^i_\beta|\boldsymbol{y}^i_{\neg\beta};\ w).\end{aligned}$$

This formulation interprets maximum composite likelihood (CL) (Lindsay, 1988) as a type of partial information likelihood for locPMAP. Here locPMAP is not a complete generative model in terms of the observation points $\boldsymbol{y}$. Although it is possible to complete the model by defining a specific mechanism to map from the local maxima set $\mathcal{L}$ to point $\boldsymbol{y}$, the corresponding full likelihood would become more challenging to compute and analyze. We can see that the essence of CL and PL is to trade off information for computational tractability, which is also reflected in the original definition (4) of PL where only the conditional information $p(y_j|\boldsymbol{y}_{\neg j};\ w)$ is taken into account. This also makes CL/PL more robust than the full MLE when the full model is misspecified, but the partial information used by CL/PL is correct (Xu & Reid, 2011).

Our perspective motivates us to decode CL/PL, interpreted as training locPMAP, by mimicking the locPMAP procedure: we first generate a randomly perturbed potential function $\tilde{\theta}(\boldsymbol{y})$ using i.i.d. Gumbel noise, and then find a local maximum with an arbitrary greedy optimization procedure (such as ICM with uniform random initialization). Although we do need to specify a particular greedy optimization method to select $\boldsymbol{y}$ from $\mathcal{L}$ for the purpose of inference, this seems to be a minor approximation, and may not influence the result significantly unless the selected greedy optimization is strongly biased in a certain way.

Algorithm 1 Local Perturb-and-MAP (locPMAP)

1. Let $\tilde{\theta}(\boldsymbol{y}, \boldsymbol{x};\ w) = \theta(\boldsymbol{y}, \boldsymbol{x};\ w) + \epsilon(\boldsymbol{y})$, where $\epsilon(\boldsymbol{y})$ are drawn i.i.d. from Gumbel $\sim G(0, 1)$.
2. Run an greedy optimization method (such as ICM) on the perturbed potentials $\tilde{\theta}(\boldsymbol{y}, \boldsymbol{x};\ w)$ to get $y_j = \arg\max_{y_j} \tilde{\theta}(y_j, \boldsymbol{y}_{\neg j}, \boldsymbol{x};\ w),\ \forall j$.

We outline these two steps in Algorithm 1. In this case, we assume a simple case of likelihood and $\mathcal{B}$ consists of the set of single variables. For simplicity, from now on we drop the dependency on $\mathcal{B}$ and simply use the notation $\mathrm{Loc}[\tilde{\theta}(\boldsymbol{y}, \boldsymbol{x};\ w)]$, where the dependency on the input $\boldsymbol{x}$ is added explicitly.

Because the result of Algorithm 1 is not deterministic, we can repeatedly run it for several iterations (by drawing multiple samples from our locPMAP model), and then take the mode of the returned samples. This also allows us to construct probabilistic outputs with error bars indicating the variance of the structured prediction. Note that in order to obtain analogous probabilistic results with other models, it would be necessary to perform expensive MCMC inference for the case of CRFs, or global combinatorial optimization under the typical Perturb-and-MAP models.

5 Learning Deep Unary Features with FCNs and Pseudolikelihood Loss

Background. Under log-linear CRF models, we typically assume that the potential function is a weighted combination of features:

$$\theta^U_i = \exp\sum_j w^U_j f^U_j(x_i) \qquad \theta^P_{ij} = \exp\sum_k w^P_k f^P_k(x_i, x_j)$$

where θ^U_i, w^U and $f^U(x_i)$ denote unary potentials for node i, learnable unary feature parameters, and unary features, respectively. Similarly, θ^P_{ij}, w^P and $f^P(x_i, x_j)$ are pairwise potentials between nodes i and j, learnable pairwise feature parameters, and pairwise features, respectively.

However, there are several important limitations related to log-linear CRF models. Such models have only a linear number of learnable parameters, which significantly limits the complexity of the model that can be learned from

the data. One way to address this limitation is to construct highly non-linear and complex features that would work well even with a linear classifier. However, hand-engineering complex features is a challenging and time-consuming task requiring lots of domain expertise. To address both of these limitations, we train a deep network that optimizes a pseudolikelihood criterion and automatically learns complex unary features for our model.

Recently, deep learning methods have been extremely successful in learning effective hierarchical features that achieve state-of-the-art results on a variety of vision tasks, including boundary detection, image classification, and semantic segmentation Bertasius et al. (2015); Krizhevsky et al. (2012); Donahue et al. (2013); Toshev & Szegedy (2013); Taigman et al. (2014). A particularly useful model for structured prediction on images is the Fully Convolutional Network (FCN) Long et al. (2015) used in combination with CRF models. These models combine the powerful methodology of deep learning for hierarchical feature learning with the effectiveness of CRFs for modeling structured pixel output, such as the class labels of neighboring pixels in semantic segmentation.

While in early approaches the FCN and the CRF were learned separately Chen et al. (2015), more recently there has been successful work that has integrated CRF learning into the FCN framework Zheng et al. (2015). Additionally, learning the parameters of the CRF in the neural network model has been addressed in Peng et al. (2009); Do & Artieres (2010); Kirillov et al. (2015).

In our approach, we train FCN and CRF jointly by optimizing the entire FCN via backpropagation with respect to the pseudolikelihood loss as explained below. We then use the locPMAP procedure shown in Algorithm 1 to decode the PL result, as justified by our intuition discussed earlier.

Optimizing the Pseudolikelihood Loss. Let our input be an image of size $h \times w \times c$, where h, w refer to the height and width of the image and c is the number of input channels ($c = 3$ for color RGB images, $c = 1$ for grayscale images). Then assume that our goal is to assign one of K possible labels to each pixel (i, j). The label typically denotes the class of the object located at pixel (i, j) or the foreground/background assignment. Now let us write our conditional pseudolikelihood probability as:

$$p(y_{(i,j)} = l \mid \boldsymbol{x}, \boldsymbol{y}_{\neg(i,j)}) = \frac{\exp(\theta_{i,j,l})}{\sum_{k=1}^{K} \exp(\theta_{i,j,k})} \quad (8)$$

where $\theta_{i,j,l}$ refers to the potential function values for label l at pixel (i, j), and where $\neg(i, j)$ indicates all the nodes connected to node (i, j). More specifically, $\theta_{i,j,l}$ denotes the product of a unary potential at a node (i, j) and all the pairwise potentials that are connected to the node (i, j). The subscript $l \in \{1, \ldots, K\}$ in the probability notation denotes that this is a potential associated with the class label l. Then, to obtain a proper probability distribution we can normalize this potential value as shown in Equation 8. Finally, we can write the loss of our FCN as:

$$L_{i,j,l} = -\log p(y_{(i,j)} = l \mid \boldsymbol{x}, \boldsymbol{y}_{\neg(i,j)};\ \theta_{i,j,l}) \quad (9)$$

The gradient of this loss can then be computed as:

$$\frac{\partial L_{i,j,l}}{\partial \theta_{i,j,l}} = p(y_{(i,j)} = l \mid \boldsymbol{x}, \boldsymbol{y}_{\neg(i,j)}) - 1\{y_{i,j} = l\} \quad (10)$$

where the last term in the equation is simply an indicator function denoting whether ground truth label $y_{i,j}$ is equal to the predicted label l. This gradient is computed for every node (i, j) and is then backpropagated to the previous layers of the FCN. We provide more details about our choice of deep architecture and the other learning details in the experimental section.

6 Experimental Results

In this section, we evaluate the results of our Local Perturb-and-MAP (locPMAP) method against other inference techniques such as loopy belief propagation (LBP) and mean field (MF) on three different datasets. In all our experiments, we use the following setup. First, we learn the parameters of a CRF-based model using the PL learning criterion. We note that the PL learning is done once, and the same learned parameters are then used for both our method and the other baseline inference techniques. This is done to demonstrate that in the context of PL learning, our locPMAP procedure acts as a better inference procedure than existing approximation inference techniques.

Since our method relies on ICM to make predictions, we compare our approach with traditional ICM. We also compare against an iterative version of ICM (ICM-iter) which is executed for the same number of iterations as our method, in order to give both methods the same "computational budget." In this iterative version of ICM, at each iteration we randomly perturb the potentials by setting a small fraction (e.g., 0.1) of them to zero (the technique is known as dropout in the deep learning literature Srivastava et al. (2014)). In all experiments, for our method and ICM-iter, we only perturb the unary potentials. Additionally, we use a grid-based graph model, with each node connected to its 4 neighbors, as this is standard for computer vision problems. The details of the pairwise potentials are discussed separately below for each task. We also tested inference of the learned model using loopy belief propagation (LBP), mean field (MF), simulated annealing (SA) and Gibbs sampling (MCMC). For each of the three tasks we show that our Local Perturb-and-MAP method consistently outperforms other inference techniques, thus demonstrating that

	Background		Foreground		Mean	
	Raw	Deep	Raw	Deep	Raw	Deep
LBP	0.846	**0.854**	0.029	0.112	0.438	0.446
MF	**0.861**	0.837	0.059	0.209	0.460	0.523
ICM-iter	0.722	0.797	0.312	0.379	0.517	0.588
SA	0.806	0.837	0.234	0.202	0.520	0.520
ICM	0.730	0.807	0.319	0.389	0.525	0.598
Gibbs	0.840	0.840	0.233	0.197	0.537	0.518
LocPMAP	0.753	0.826	**0.337**	**0.404**	**0.545**	**0.615**

Table 1: Results of handwritten digit denoising on the MNIST dataset. Performance is measured according to the the Intersection over Union (IoU) for both the foreground and the background mask. We compare the results when using raw corrupted pixel intensities as unary features (Raw) versus deep unary features learned via an FCN (Deep). Our locPMAP method outperforms the other baseline inference techniques. Additionally, we observe that using deep features tends to improve the overall accuracy.

locPMAP forms a better practical inference procedure for models learned from PL optimization. We now present each of our experiments in more detail.

6.1 Denoising Handwritten Digits

For our first evaluation we use the MNIST dataset LeCun & Cortes (2010) which contains black and white images of handwritten digits. We corrupt each 28×28 image using Gumbel noise with 0.25 signal-to-noise ratio. This produces corrupted grayscale images, which are used as input to our system. The objective is to recover the original black/white (background/foreground) value of each pixel.

In our experiments, we used 5000 images for training and 5000 images for testing. We performed two types of experiments on this task. First, we evaluated all methods using the corrupted pixel intensity values as unary features. For pairwise features between nodes i and j, we used the corrupted intensity values at pixels i and j.

For the second experiment, we trained a fully convolutional network (FCN) to learn the unary features. To optimize the pseudolikelihood criterion, we used pairwise potential parameters that were learned using the corrupted potentials. We kept the pairwise potential parameters fixed and performed gradient backpropagation only through the unary feature parameters.

To train the FCN, we used an architecture composed of 5 convolutional layers with kernel size of 3×3 for the first four layers and kernel size of 1×1 for the last layer. The output plane dimensions for the convolutional layers were $64, 126, 256, 512$ and 2 respectively. As hyperparameters, we used a learning rate of 10^{-6}, a momentum of 0.9, a batch size of 100, and RELU non-linear functions in between the convolutional layers. To avoid the reduction in resolution inside the deep layers, we did not use any pooling layers. We trained our FCN to minimize the pseudo-likelihood loss for ≈ 3000 iterations. For all of our deep learning experiments we used the Caffe library Jia et al. (2014). To run our locPMAP method, we used 50 iterations.

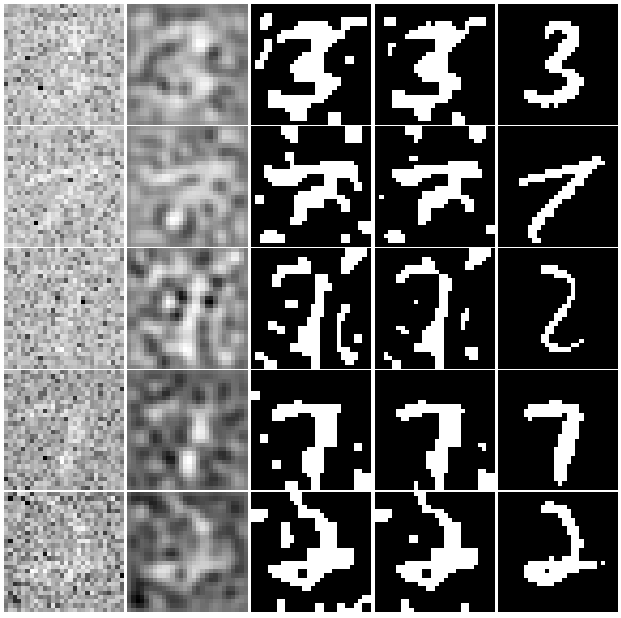


Figure 1: Visualizations of handwritten digit denoising. Images in the first column represent corrupted digit inputs. The second column shows the unary features that were learned using fully convolutional networks and a pseudo-likelihood loss. Images in the third column correspond to ICM predictions, whereas the fourth column depicts our Local Perturb-and-MAP results. In the last column we show the corresponding original ground truth black and white images. Note that compared to the ICM predictions, our method makes fewer false positive predictions. Additionally, we observe that deep features exhibit significantly less noise than the original input, which improves the overall accuracy (see quantitative results in Table 1).

In Table 1, we present quantitative results of our method and several baseline methods. The performance of each method is evaluated in terms of the Intersection over Union (IoU) metric for background and foreground classes. Additionally, we separately evaluate each baseline inference method using corrupted pixel values as unary features (Raw) and also using deep features (Deep). The results demonstrate that our locPMAP method outperforms the other inference baselines. Additionally, we note that learning unary features via FCNs substantially improves the accuracy for most methods. We also present qualitative results in Figure 1.

We also tested other inference methods such as tree-reweighted belief propagation, and graph cuts, but found that these methods performed very poorly with PL learning. Due to the limited space available, we omit these results from our paper.

	Background		Foreground		Mean	
	Raw	Deep	Raw	Deep	Raw	Deep
LBP	0.539	0.539	0.080	0.154	0.310	0.347
MF	0.686	0.695	0.598	0.674	0.642	0.684
ICM-iter	0.659	0.731	0.637	0.727	0.648	0.729
SA	0.688	0.692	0.623	0.696	0.655	0.694
Gibbs	**0.692**	0.708	0.624	0.696	0.658	0.702
ICM	0.672	0.746	0.661	0.733	0.667	0.739
LocPMAP	0.681	**0.754**	**0.666**	**0.735**	**0.673**	**0.745**

Table 2: Results on the Caltech Silhouette Reconstruction task. We evaluate the results using the Intersection over Union (IoU) metric. We test each method using raw unary features versus deeply learned FCN features. Our method achieves better accuracy than the other baseline inference methods.

6.2 Caltech Silhouette Reconstruction

For our second task, we choose a more diverse dataset consisting of noise-corrupted silhouettes generated from the ground truth foreground/background segmentations of images from Caltech-101 Fei-Fei et al. (2007), which spans 101 object classes. Each silhouette is a 28×28 image generated by adding Gaussian noise with 0.5 signal-to-noise ratio to each pixel of the ground-truth foreground/background segmentation. The goal is to reconstruct the original ground truth foreground/background segmentation from the corrupted silhouette. Due to the large number of object classes in the dataset, the variability of the silhouette shape is much larger compared to the case of the digit denoising task.

We use the exact same experimental setup as in the earlier experiment for handwritten digit denoising. We present quantitative results in Table 2. Again, the results indicate that locPMAP outperforms the other inference baselines for the model learned from PL optimization.

6.3 Scene Labeling

As our last task, we consider the problem of semantic scene segmentation. For this task, we use the Stanford background dataset Gould et al. (2009), which has per-pixel annotations for a total of 715 scene color images of size 240×320. Our goal is to assign every pixel to one of 8 possible classes: sky, tree, road, grass, water, building, mountain, and foreground. In this case the input to the system is the RGB photo and the desired output is the semantic segmentation. We randomly split the dataset into a training set of 600 images and a test set of 115 images.

Once again we perform two experiments for this task. First, we use the boosted unary potentials provided by Gould et al. (2009) as the unary features in our CRF model. Next, to construct pairwise potentials we extract HFL boundaries Bertasius et al. (2015) from the images. We then compute the gradient on the boundaries, and use it as pairwise features for every pair of adjacent pixels. Then, just as earlier, we learn the CRF parameters by optimizing the pseudolikelihood objective, and finally perform the inference using the learned parameters.

For the second experiment, our goal is to learn deep features from the data instead of using the boosted features provided by Gould et al. (2009). To do this we use a fully convolutional network architecture based on DeepLab Chen et al. (2015). This architecture contains 19 convolutional layers. To train the FCN, we use the same learning hyperparameters and setup as in the previous two experiments. As before, we fix the pairwise potential parameters, and only learn the parameters associated with the unary terms. After the unary learning is done, we learn new pairwise parameters given the learned unary features.

In Table 3, we present our results for the scene labeling task. Once again, we show that our Local Perturb-and-MAP method outperforms other inference methods in both scenarios: using boosted unary features Gould et al. (2009) and also using our learned deep features.

Interestingly, we note that for this task, the accuracy we achieve using deep features is lower than the accuracy obtained using boosted features. We hypothesize that this happens because the Stanford Background dataset is relatively small (600 training images) and thus it does not enable effective training of the large-capacity FCN.

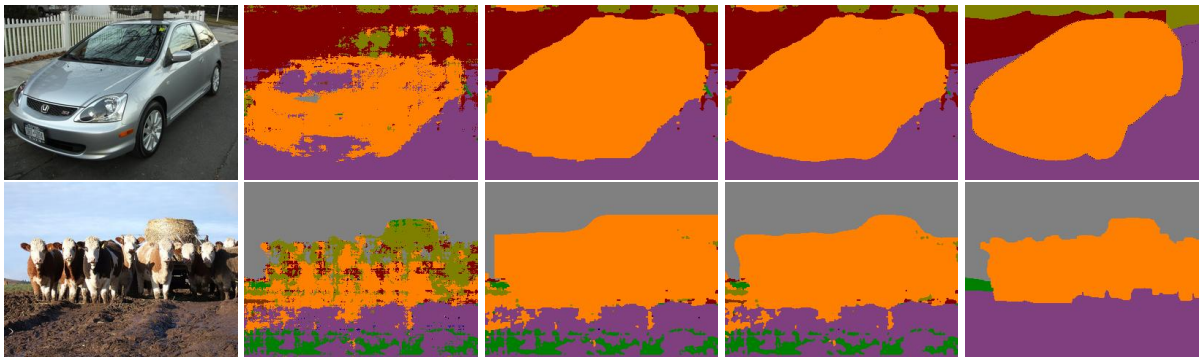
Figure 2 shows some qualitative results. Note that compared to the results achieved by Gould et al. (2009), our predictions are spatially smoother. Similarly, relative to the ICM predictions, Local Perturb-and-MAP yields crispier boundaries around the objects and more coherent segments. We also note that unlike most inference methods that can only predict the discrete label, our method outputs the prediction variance for every pixel in addition to the label (See Figure 3). This probabilistic prediction may be useful in practical scenarios, such as for the analysis of failures or when confidence estimates are needed.

7 Discussion

We introduced a Local Perturb-and-MAP (LocPMAP) framework which yields a novel connection with pseudolikelihood (PL). Our empirical analysis demonstrates that locPMAP forms a better inference procedure for models learned from PL optimization than existing approximate inference methods. Future work includes extending our method to use larger blocks (corresponding to composite likelihood). Our new perspective on pseudolikelihood may also be leveraged to solve challenging structure prediction problems in other domains.

References

Bertasius, G., Shi, J., and Torresani, L. High-for-low and low-for-high: Efficient boundary detection from deep object features



Input Image | Boosted Classifier | ICM | LocPMAP | Ground Truth

Figure 2: A figure illustrating qualitative results for the scene labeling task. In the first column, we show the original RGB input images. The second column represents boosted classifier predictions Gould et al. (2009) while in the third column, we illustrate ICM predictions. In the fourth column, we present the results of our LocPMAP method. Notice that, compared to the other methods, our predictions are spatially smoother and crispier around the object boundaries.

	Sky		Tree		Road		Grass		Water		Building		Mountain		Object		Mean	
	Raw	Deep	Raw	Deep	Raw	Deep	Raw	Deep	Raw	Deep	Raw	Deep	Raw	Deep	Raw	Deep	Raw	Deep
ICM	0.807	0.843	0.648	0.628	0.845	0.853	0.790	0.729	0.770	0.759	**0.686**	0.724	0.343	0.175	0.633	0.668	0.690	0.672
ICM-iter	0.837	**0.846**	0.649	0.631	0.845	0.855	0.790	0.736	0.776	0.767	**0.686**	0.726	0.421	0.233	0.646	0.684	0.706	0.685
LBP	**0.861**	0.840	0.640	0.629	0.832	0.842	0.783	0.690	0.771	0.707	0.675	0.708	**0.522**	0.157	0.646	0.552	0.716	0.638
Gibbs	0.854	0.840	**0.659**	0.632	**0.848**	0.856	0.792	0.738	**0.802**	0.766	0.705	0.728	0.439	0.215	0.646	0.690	0.718	0.683
LocPMAP	0.854	**0.846**	0.650	**0.636**	0.844	**0.859**	**0.794**	**0.742**	0.784	**0.769**	**0.686**	**0.729**	0.512	**0.259**	**0.652**	**0.701**	**0.722**	**0.693**

Table 3: Quantitative results for the scene labeling task. The performance is evaluated using an IoU metric for each class. For raw features we use the output predictions from Gould et al. (2009). In this case, we observe that raw unary features yield higher accuracy in comparison to deep unary features, probably due to the small dataset size. However, we observe again that our LocPMAP is overall the best approach across all inference methods considered here.

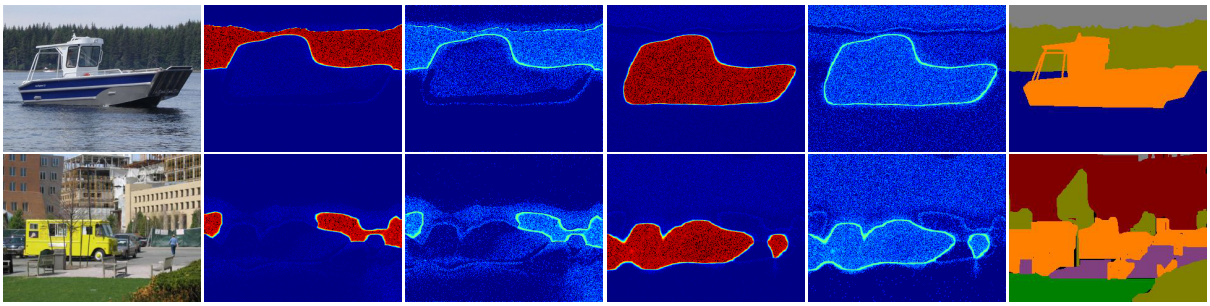


Figure 3: Figure that illustrates probabilities and variance of LocPMAP inference method applied on deeply learned FCN unary features. Note that while most other inference techniques produce a discrete solution to the problem, our LocPMAP method outputs discrete solution and also probabilities and variance for each pixel. In the the second and third columns, we show probabilities and variance for the "Tree" class predictions. The fourth and fifth columns depict the probabilities and the variance of the predictions for the "Object" class respectively. In the last column we show images corresponding to the ground truth.

and its applications to high-level vision. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Besag, J. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 24(3):pp. 179–195, 1975. ISSN 00390526.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

Do, T.-M.-T. and Artieres, T. Neural conditional random fields. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, 5 2010.

Domke, J. Learning graphical model parameters with approximate marginal inference. volume 35, pp. 2454–2467. IEEE, 2013.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.

Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Comput. Vis. Image Underst.*, 106(1):59–70, April 2007. ISSN 1077-3142. doi: 10.1016/j.cviu.2005.09.012.

Gane, A., Hazan, T., and Jaakkola, T. S. Learning with maximum a-posteriori perturbation models. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, pp. 247–256, 2014.

Gelfand, A. E. *Bottom-Up Approaches to Approximate Inference and Learning in Discrete Graphical Models DISSERTATION*. PhD thesis, UNIVERSITY OF CALIFORNIA, IRVINE, 2014.

Gould, S., Fulton, R., and Koller, D. Decomposing a scene into geometric and semantically consistent regions. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.

Hazan, T. and Jaakkola, T. S. On the partition function and random maximum a-posteriori perturbations. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012.

Hazan, T., Maji, S., Keshet, J., and Jaakkola, T. Learning efficient random maximum a-posteriori predictors with non-decomposable loss functions. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 1887–1895. Curran Associates, Inc., 2013.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

Kirillov, A., Schlesinger, D., Forkel, W., Zelenin, A., Zheng, S., Torr, P. H. S., and Rother, C. Efficient likelihood learning of a generic CNN-CRF model for semantic segmentation. *CoRR*, abs/1511.05067, 2015.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc., 2012.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1.

Lauritzen, S. L. *Graphical models*. Clarendon Press, 1996.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010.

Lindsay, B. G. Composite likelihood methods. *Contemporary mathematics*, 80(1):221–39, 1988.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. *CVPR*, November 2015.

McFadden, D. et al. Conditional logit analysis of qualitative choice behavior. 1973.

Meshi, O., Sontag, D., Jaakkola, T., and Globerson, A. Learning efficiently with approximate inference via dual losses. International Machine Learning Society, 2010.

Papandreou, G. and Yuille, A. L. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In Metaxas, D. N., Quan, L., Sanfeliu, A., and Gool, L. J. V. (eds.), *ICCV*, pp. 193–200. IEEE, 2011. ISBN 978-1-4577-1101-5.

Peng, J., Bo, L., and Xu, J. Conditional neural fields. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 22*, pp. 1419–1427. Curran Associates, Inc., 2009.

Poon, H. and Domingos, P. Sum-product networks: A new deep architecture. In *Uncertainty in Artificial Intelligence*, pp. 337?346, 2011.

Srivastava, N., Salakhutdinov, R., and Hinton, G. Fast inference and learning for modeling documents with a deep boltzmann machine.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

Stoyanov, V. and Eisner, J. Minimum-risk training of approximate crf-based nlp systems. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 120–130. Association for Computational Linguistics, 2012.

Sutton, C. A. and McCallum, A. Piecewise training for structured prediction. *Machine Learning*, 77(2-3):165–194, 2009.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Tarlow, D., Adams, R. P., and Zemel, R. S. Randomized optimum models for structured prediction. In Lawrence, N. D. and Girolami, M. (eds.), *AISTATS*, volume 22 of *JMLR Proceedings*, pp. 1221–1229. JMLR.org, 2012.

Toshev, A. and Szegedy, C. Deeppose: Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659, 2013.

Wainwright, M. J. Estimating the wrong graphical model: Benefits in the computation-limited setting. *Journal of Machine Learning Research*, 7:1829–1859, 2006.

Xiang, R. and Neville, J. On the mismatch between learning and inference for single network domains.

Xu, X. and Reid, N. On the robustness of maximum composite likelihood estimate. *Journal of Statistical Planning and Inference*, 141(9):3047–3054, 2011.

Yellott, J. I. The relationship between luce's choice axiom, thurstone's theory of comparative judgment, and the double exponential distribution. *Journal of Mathematical Psychology*, 15 (2):109 – 144, 1977. ISSN 0022-2496.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision (ICCV)*, 2015.