Optimizing Expected Intersection-over-Union with Candidate-Constrained CRFs

Faruk Ahmed
Université de Montréal
faruk.ahmed@umontreal.ca

Daniel Tarlow
Microsoft Research
dtarlow@microsoft.com

Dhruv Batra
Virginia Tech
dbatra@vt.edu

Abstract

We study the question of how to make loss-aware predictions in image segmentation settings where the evaluation function is the Intersection-over-Union (IoU) measure that is used widely in evaluating image segmentation systems. Currently, there are two dominant approaches: the first approximates the Expected-IoU (EIoU) score as Expected-Intersection-over-Expected-Union (EIoEU); and the second approach is to compute exact EIoU but only over a small set of high-quality candidate solutions. We begin by asking which approach we should favor for two typical image segmentation tasks. Studying this question leads to two new methods that draw ideas from both existing approaches. Our new methods use the EIoEU approximation paired with high quality candidate solutions. Experimentally we show that our new approaches lead to improved performance on both image segmentation tasks.

1. Introduction

The goal of a "good" evaluation metric for a vision task is to quantify the (dis)similarity of a proposed solution w.r.t. the ground truth in a perceptually meaningful way. For the task of semantic image segmentation (labeling each pixel in an image with a class label), one popular metric is the Jaccard Index or Intersection-over-Union (IoU) measure [8], computed between the binary masks of a predicted segmentation and ground-truth, averaged over all categories.

Motivation and Challenge. A number of works have argued [7, 8] that IoU is an evaluation metric better correlated with human judgement than alternatives. Unfortunately, it is a *high order loss* [9, 19, 23] making it difficult to train models to optimize performance under this measure. In general, the recourse has been to optimize a simpler loss amenable to tractable training, such as the Hamming loss.

Recently, there has been renewed interest [13, 18, 20, 21] in using tools from Bayesian Decision Theory (BDT). Specifically, let $\mathbf{x}$ be an image, $\mathbf{y}$ be a segmentation, $\mathrm{P}(\mathbf{y}|\mathbf{x})$ be the conditional probability learned by a model (*e.g.*, a Conditional Random Field (CRF)), and $\ell(\cdot,\cdot)$ be a loss function. BDT provides an elegant recipe for making decisions based on the principle of minimum expected loss or Minimum *Bayes Risk* (MBR):

$$\mathbf{y}^{\mathrm{MBR}} = \operatorname*{argmin}_{\hat{\mathbf{y}}\in\mathcal{Y}} \underbrace{\sum_{\mathbf{y}\in\mathcal{Y}} \ell(\hat{\mathbf{y}},\mathbf{y})\mathrm{P}(\mathbf{y}|\mathbf{x})}_{\text{Expected Loss / Bayes Risk}} . \quad (1)$$

Goal. The goal of this paper is to study tractable techniques for such decision-theoretic predictions under the IoU measure. In most models of practical interest, this task is intractable because it requires both an enumeration and a search over an exponentially-large output space $\mathcal{Y}$ (*e.g.*, the space of all possible segmentations of an image).

How is this done today? There are two main approaches, each making a different set of assumptions to achieve tractability on the summation and minimization:

1. **Approximating the loss $\ell(\cdot,\cdot)$:** A recent pair of papers [18, 21] have proposed an approximation for the IoU measure and a greedy heuristic for optimizing it [1]. The key idea is to compute Expected-Intersection-over-Expected-Union (EIoEU) as opposed to Expected-Intersection-over-Union (EIoU). As we explain later, the former factorizes, while the latter does not.

2. **Approximating the distribution $\mathrm{P}(\cdot)$:** [20] proposed a method for exactly optimizing Expected-IoU, but not under $\mathrm{P}(\cdot)$; instead under a delta-approximation of $\mathrm{P}(\cdot)$ having support only at a collection of appropriately chosen M candidate solutions[2]. In this case, the optimization is a simple enumeration over the candidates.

Contributions and Overview. First, we show that despite seeming complementary and disparate on surface, the two approximations in existence today are related under a par-

[1] [18] propose other heuristics for optimizing EIoEU in their work, but it was shown that the greedy heuristic was the best performing.

[2] The method of [20] is actually not specific to IoU, and can be applied to any loss function of interest.

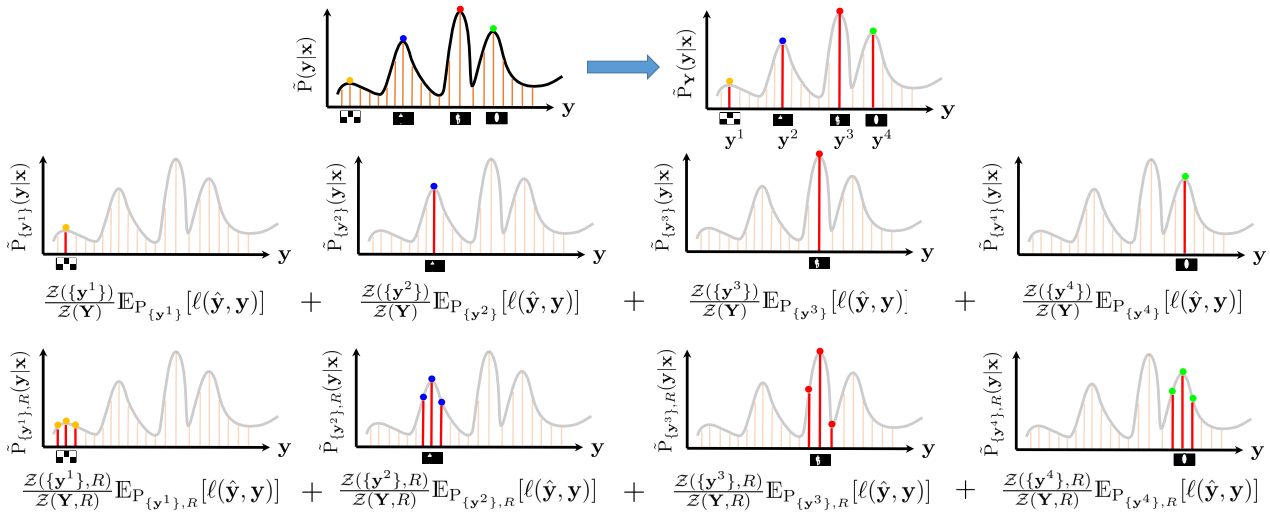


Figure 1: (*top row*) [20] proposed an approximate MBR predictor which computes exact EIoU over a set of candidate solutions $\mathbf{Y} = \{\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3, \mathbf{y}^4\}$. (*middle row*) We show that this is equivalent to averaging EIoU over distributions with single point support on the candidate solutions and applying the approximation of [18, 21] (EIoEU), which performs better than applying the approximation over the entire solution space. (*bottom row*) This leads us to suggest averaging EIoEU over Hamming-constrained distributions. We show how to implement such distributions using cardinality potentials.

	Search	Distribution
Nowozin [18]	greedy	CRF
Premachandran *et al*. [20]	`enum`	delta
`CRF-EIoEU+enum` (ours)	`enum`	CRF
`C`^{3}`RF-EIoEU+enum` (ours)	`enum`	C^{3}RF

Table 1: Comparison of the search heuristics used by the methods we discuss and the distributions they reason over.

ticular view, which we then use as a starting point for developing methods that interpolate the two. Specifically, we show that approximating the distribution $\mathrm{P}(\cdot)$ with delta functions (as in [20]) and computing EIoU is equivalent to averaging EIoEU (as in [18, 21]) over multiple distributions each with a single delta function. As we explain in Section 4, this follows from a simple observation that the EIoEU approximation is exact for a delta distribution. Although straightforward in hindsight, this connection improves our current understanding of the literature.

Inspired by this connection, we ask the following question – can we combine the best of both approaches? Specifically, can we combine the idea of using EIoEU with the idea of optimizing EIoU for a set of candidate solutions?

We propose two ways of performing this combination:

1. **CRF-EIoEU with Candidates:** We utilize the EIoEU approximation [18, 21] to estimate EIoU with marginals over the entire distribution, but only for the set of candidate solutions from [20]. Essentially, this replaces the greedy search in [18, 21] with a search over a small number of candidates from [20] (`enum`). We find that this combination outperforms the greedy search of [18, 21], which suggests that the search strategy of [20] is a more practical heuristic.

2. **C^{3}RF-EIoEU with Candidates**: Summarizing a distribution with a collection of delta functions seems particularly harsh, and it is natural to expect that for some problems and distributions, an intermediate level of summarization might be more useful to reason over. We propose reasoning over a distribution that has non-zero support only in Hamming regions around the candidate set of solutions, which we call a Candidate-Constrained CRF (C^{3}RF). We show how to implement such candidate-constrained distributions via cardinality potentials [11, 22], and find that this generalization leads to further improvements in performance on the task of semantic segmentation on PASCAL VOC 2012.

In Table 1, we summarize the search heuristics adopted by the various methods and the distributions over which they compute EIoU.

2. Related Work

We already discussed two approaches in the introduction for MBR prediction under IoU. Most previous work on loss-aware decision making [3, 16, 25] for general loss functions is applicable to unstructured prediction – binary or multi-class classification – and not structured prediction. One exception is [20], which is applicable to general (high-order) losses, due to the approximation over a set of candidate solutions. [20] can be viewed as a special case of our approach that arises when the Hamming radius in the C^{3}RF is set to 0 (*i.e*., the model is a mixture of delta distributions at each of the candidates). We will show in the experiments that the

richer set of distributions that arise from the C^3RF model allow us to improve over this baseline.

Other methods that use candidate solutions followed by loss-aware decision making have been explored in the Machine Translation (MT) community. MBR decoding for MT was first introduced in [14], where it was used to predict the best translation from a set of N-best candidate translations [1, 10]. Subsequent work [15, 24] showed efficient ways to perform MBR prediction from a larger pool of candidate solutions in the statistical MT setting.

The other relevant line of work comes from variational inference techniques for mixture models [4, 12, 17]. There are two main differences in our approach. First, rather than *learn* the mixture components, we simply fix the centers of the components to the candidate solutions. This simplifies the inference task. Second, we make use of hard mixture assignments, but note that softer choices could potentially be used instead, and that is a topic for future exploration.

3. Background

We begin by establishing our notation, and reviewing background on probabilistic structured prediction.

Basic Notation. For any positive integer n, let $[n]$ be shorthand for the set $\{1, 2, \ldots, n\}$. Given an input image $\mathbf{x} \in \mathcal{X}$, our goal is to make a prediction about output variables $\mathbf{y} \in \mathcal{Y}$, where $\mathbf{y}$ may be a figure-ground segmentation, or a category-level semantic segmentation. Specifically, let $\mathbf{y} = \{y_1 \ldots y_n\}$ be a set of discrete random variables, each taking values in a finite label set, $y_i \in Y_i$. In semantic segmentation, i indexes over the (super-)pixels in the image, and these variables are labels assigned to each (super-)pixel, *i.e.* $Y_i = \{\text{cat, boat, cow}, \ldots\}$. For a set $F \subseteq [n]$, we use y_F to denote the tuple $\{y_i \mid i \in F\}$, and Y_F to be the cartesian product of the individual label spaces $\times_{i \in F} Y_i$.

3.1. Conditional Random Fields and Factor Graphs

Conditional Random Fields (CRFs) are probabilistic models that represent conditional probabilities $\mathrm{P}(\mathbf{y}|\mathbf{x})$ in a compact manner via factor graphs. Let $G = (\mathcal{V}, \mathcal{E})$ be a bipartite graph with two kinds of vertices – variable $i \in [n]$ and factor nodes $F \subseteq [n]$. Each factor holds a local compatibility function, measuring the score of the variables in its scope: $\theta_F : Y_F \to \mathbb{R}$. An edge $\{i, F\} \in \mathcal{E}$ indicates that variable y_i participates in the factor function $\theta_F(\cdot)$, *i.e.*, $i \in F$.

The score for any configuration $\mathbf{y}$ is given by $S(\mathbf{y}|\mathbf{x}) = \sum_F \theta_F(y_F)$, and the corresponding probability is given by the Gibbs distribution: $\mathrm{P}(\mathbf{y}|\mathbf{x}) = \frac{1}{\mathcal{Z}} \exp S(\mathbf{y}|\mathbf{x})$, where $\mathcal{Z}$ is the partition function or the normalization constant.

If we wish to make a prediction from our model, we employ a predictor, which converts $\mathrm{P}(\mathbf{y}|\mathbf{x})$ into a prediction $\hat{\mathbf{y}}$.

In the next two sections we shall briefly review the two approaches towards performing Bayesian decision making under the IoU measure.

3.2. Empirical MBR

Premachandran *et al*. [20] tackle the intractability of the MBR predictor by simply restricting the solution space to a small set of candidate solutions. Specifically, given a set of M candidate solutions $\mathbf{Y} = \{\mathbf{y}^1, ..., \mathbf{y}^M\}$, a loss function $\ell(.,.)$, and the Gibbs distribution $\mathrm{P}(.)$ corresponding to the scoring function, they proposed an 'Empirical MBR' (or EMBR) predictor:

$$\mathbf{y}^{\texttt{Delta-EIoU+enum}} = \operatorname*{argmin}_{\hat{\mathbf{y}} \in \mathbf{Y}} \sum_{\mathbf{y} \in \mathbf{Y}} \ell(\hat{\mathbf{y}}, \mathbf{y}) \tilde{\mathrm{P}}(\mathbf{y}|\mathbf{x}) \tag{2}$$

where $\tilde{\mathrm{P}}(\mathbf{y}|\mathbf{x}) = \frac{\exp\left(S(\mathbf{y})\right)}{\sum_{\mathbf{y}' \in \mathbf{Y}} \exp\left(S(\mathbf{y}')\right)}$ is the re-normalized distribution over the M candidate solutions. In this paper, we refer to the predictor as `Delta-EIoU+enum`, because of the way the distribution is assumed to be summarized by a set of delta functions at the candidate solutions, and expected loss is enumerated for each of the candidate solutions.

The candidate solutions are acquired by using the same setup as [20] to generate DivMBest solutions [2].

3.3. EIoEU

For a ground-truth segmentation $\mathbf{y}$ and a predicted segmentation $\hat{\mathbf{y}}$, where each variable can take K possible classes $y_i \in \{1, \cdots, K\}\ \forall i \in \mathcal{V}$, the Intersection-over-Union measure is given by

$$\ell(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{K} \sum_{k=1}^{K} \frac{\sum_{i \in \mathcal{V}} \mathbb{1}\{\hat{y}_i = k \wedge y_i = k\}}{\sum_{i \in \mathcal{V}} \mathbb{1}\{\hat{y}_i = k \vee y_i = k\}}. \tag{3}$$

This definition is actually a gain function and not a loss, hence it will be maximized. [18, 21] proposed approximating the Expected-IoU by Expected-Intersection-over Expected-Union (EIoEU):

$$\mathbb{E}_{\mathrm{P}}[\ell(\hat{\mathbf{y}}, \mathbf{y})] = \sum_{\mathbf{y} \in \mathcal{Y}} \ell(\hat{\mathbf{y}}, \mathbf{y}) \mathrm{P}(\mathbf{y}|\mathbf{x}) \tag{4}$$

$$= \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}_{\mathrm{P}}\Big[\frac{\sum_{i \in \mathcal{V}} \mathbb{1}\{\hat{y}_i = k \wedge y_i = k\}}{\sum_{i \in \mathcal{V}} \mathbb{1}\{\hat{y}_i = k \vee y_i = k\}}\Big] \tag{5}$$

$$\approx \frac{1}{K} \sum_{k=1}^{K} \frac{\mathbb{E}_{\mathrm{P}}[\sum_{i \in \mathcal{V}} \mathbb{1}\{\hat{y}_i = k \wedge y_i = k\}]}{\mathbb{E}_{\mathrm{P}}[\sum_{i \in \mathcal{V}} \mathbb{1}\{\hat{y}_i = k \vee y_i = k\}]}, \tag{6}$$

which can now be written simply as a function of single variable (*i.e.* pixel) marginals p^i:

$$\frac{1}{K} \sum_{k=1}^{K} \frac{\sum_{i \in \mathcal{V}} \mathbb{1}\{\hat{y}_i = k\} p^i(k)}{\sum_{i \in \mathcal{V}} \left(\mathbb{1}\{\hat{y}_i = k\} + p^i(k) \mathbb{1}\{\hat{y}_i \neq k\}\right)} = f(\mathcal{P}, \hat{\mathbf{y}}) \tag{7}$$

where $\mathcal{P} = \{p^i(y_i|\mathbf{x})\}_{i,y_i} = \left\{\sum_{\mathbf{y}':\mathbf{y}'_i=y_i} \mathrm{P}(\mathbf{y}'|\mathbf{x})\right\}_{i,y_i}$ is the set of marginals. [18, 21] suggest a greedy heuristic for finding the solution with optimum EIoEU.

4. Relating EMBR and EIoEU

We now show how the EIoEU approximation [18, 21] is related to the EMBR predictor (2) of [20].

Given a set of solutions $\mathbf{Y}$, let us define $\mathcal{Z}(\mathbf{Y}) = \sum_{\mathbf{c}\in\mathbf{Y}} \exp\left(S(\mathbf{c}|\mathbf{x})\right)$ as the normalization constant for the multiple-delta distribution over these solutions. For example, $\mathcal{Z}(\{\mathbf{c}\}) = \exp\left(S(\mathbf{c}|\mathbf{x})\right)$ is a special case for a single delta distribution. With this notation, we can express the multiple-delta distribution used by [20] as:

$$\mathrm{P}_{\mathbf{Y}}(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{c}\in\mathbf{Y}} \frac{\mathcal{Z}(\{\mathbf{c}\})}{\mathcal{Z}(\mathbf{Y})} \mathbb{1}\{\mathbf{y}=\mathbf{c}\}. \tag{8}$$

Plugging this definition of $\mathrm{P}(.)$ into the expression for EIoU, we have

$$\begin{aligned}\mathbb{E}_{\mathrm{P}_{\mathbf{Y}}}[\ell(\hat{\mathbf{y}},\mathbf{y})] &= \sum_{\mathbf{y}\in\mathcal{Y}} \ell(\hat{\mathbf{y}},\mathbf{y}) \sum_{\mathbf{c}\in\mathbf{Y}} \frac{\mathcal{Z}(\{\mathbf{c}\})}{\mathcal{Z}(\mathbf{Y})} \mathbb{1}\{\mathbf{y}=\mathbf{c}\} \\ &= \sum_{\mathbf{c}\in\mathbf{Y}} \frac{\mathcal{Z}(\{\mathbf{c}\})}{\mathcal{Z}(\mathbf{Y})} \sum_{\mathbf{y}\in\mathcal{Y}} \ell(\hat{\mathbf{y}},\mathbf{y}) \mathbb{1}\{\mathbf{y}=\mathbf{c}\}, \end{aligned} \tag{9}$$

which can be interpreted as a weighted average of EIoU under M distributions, each with support on a single solution, *i.e.* $\mathrm{P}_{\{\mathbf{c}\}}(\mathbf{y}) = \mathbb{1}\{\mathbf{y}=\mathbf{c}\}, \forall \mathbf{c}\in\mathbf{Y}$.

If we apply the EIoEU approximation (7) to the inner sum, the expression turns into:

$$\sum_{\mathbf{c}\in\mathbf{Y}} \frac{\mathcal{Z}(\{\mathbf{c}\})}{\mathcal{Z}(\mathbf{Y})} \frac{1}{K} \sum_{k=1}^{K} \frac{\sum_{i\in\mathcal{V}} \mathbb{1}\{\hat{y}_i=k\} p^i_{\mathbf{c}}(k)}{\sum_{i\in\mathcal{V}} \left(\mathbb{1}\{\hat{y}_i=k\} + p^i_{\mathbf{c}}(k)\mathbb{1}\{\hat{y}_i\neq k\}\right)}. \tag{10}$$

Since each delta distribution $\mathrm{P}_{\{\mathbf{c}\}}$ has support only on solution $\mathbf{c}$, the marginals are also delta functions $p^i_{\mathbf{c}}(k) = \mathbb{1}\{\mathbf{c}_i = k\}$. Substituting these marginals above converts EIoEU into EIoU – *i.e.* the EIoEU approximation is exact for a single delta distribution. Thus, we derive the same predictor as (2).

5. Our Proposed Generalizations

This connection between the EMBR predictor [20] and the EIoEU approximation [18, 21] leads us to consider combining ideas from both.

Optimize EIoEU only for candidate solutions: As shown in Sec. 4, the framework of [20] can be viewed as optimizing EIoEU over a summarized delta distribution. It is natural to extend this to optimizing EIoEU over the entire distribution for the set of candidate solutions. We call this predictor `CRF-EIoEU+enum`.

Reason over an intermediate level of summarization: We consider reasoning over an intermediate level of summarization of the distribution, with the delta approximation at one extreme and the original distribution at the other. Just as the EMBR method averages loss over a collection of delta distributions, we shall average EIoU over a collection of distributions that have support only in Hamming neighborhoods of the candidate solutions.

5.1. Candidate-Constrained CRFs

In analogy to the definition of $\mathrm{P}_{\{\mathbf{c}\}}$, let $\mathrm{P}_{\{\mathbf{c}\},R}(\mathbf{y}|\mathbf{x}) \propto \mathbb{1}\{\Delta^{\text{Ham}}(\mathbf{y},\mathbf{c}) \leq R\} \exp\left(S(\mathbf{y}|\mathbf{x})\right)$ be the distribution constrained to have support only within a Hamming ball of radius R centered at a candidate solution $\mathbf{c}$, where $\Delta^{\text{Ham}}(\mathbf{y},\mathbf{c}) = \sum_i \mathbb{1}\{y_i \neq c_i\}$ is the Hamming distance function.

Recalling that $\mathbf{Y} = \{\mathbf{y}^1, \ldots, \mathbf{y}^M\}$ is the set of candidate solutions, we define the candidate constrained distribution, which is parameterized by the candidate set $\mathbf{Y}$ and a radius R, to be

$$\mathrm{P}_{\mathbf{Y},R}(\mathbf{y}|\mathbf{x}) \propto \sum_{\mathbf{c}\in\mathbf{Y}} \mathbb{1}\{\Delta^{\text{Ham}}(\mathbf{y},\mathbf{c}) \leq R\} \exp\left(S(\mathbf{y}|\mathbf{x})\right), \tag{11}$$

which can be rewritten as follows:

$$\mathrm{P}_{\mathbf{Y},R}(\mathbf{y}|\mathbf{x}) \propto \sum_{\mathbf{c}\in\mathbf{Y}} \mathcal{Z}(\{\mathbf{c}\},R) \mathrm{P}_{\{\mathbf{c}\},R}(\mathbf{y}|\mathbf{x}), \tag{12}$$

where $\mathcal{Z}(\mathbf{Y},R) = \sum_{\mathbf{y}\in\mathcal{Y}} \sum_{\mathbf{c}\in\mathbf{Y}} \mathbb{1}\{\Delta^{\text{Ham}}(\mathbf{y},\mathbf{c}) \leq R\} \exp\left(S(\mathbf{y}|\mathbf{x})\right)$ is the normalizing constant (intuitively $\mathcal{Z}(\mathbf{c},R)$ is the mass of the candidate solution $\mathbf{c}$). Following the same steps as in Sec. 4, we can derive an expression for average EIoU over the individual Hamming-constrained distributions:

$$\begin{aligned}\mathbb{E}_{\mathrm{P}_{\mathbf{Y},R}}[\ell(\hat{\mathbf{y}},\mathbf{y})] &= \sum_{\mathbf{y}\in\mathcal{Y}} \ell(\hat{\mathbf{y}},\mathbf{y}) \mathrm{P}_{\mathbf{Y},R}(\mathbf{y}|\mathbf{x}) \\ &= \sum_{\mathbf{c}\in\mathbf{Y}} \frac{\mathcal{Z}(\{\mathbf{c}\},R)}{\mathcal{Z}(\mathbf{Y},R)} \sum_{\mathbf{y}\in\mathcal{Y}} \ell(\hat{\mathbf{y}},\mathbf{y}) \mathrm{P}_{\{\mathbf{c}\},R}(\mathbf{y}|\mathbf{x}) \\ &= \sum_{\mathbf{c}\in\mathbf{Y}} \frac{\mathcal{Z}(\{\mathbf{c}\},R)}{\mathcal{Z}(\mathbf{Y},R)} f(\mathcal{P}_{\{\mathbf{c}\},R}, \hat{\mathbf{y}}). \end{aligned} \tag{13}$$

It's easy to notice that if we assume a constant IoU for $\hat{\mathbf{y}}$, and set it to $\ell(\hat{\mathbf{y}},\mathbf{c})$ for all $\mathbf{y}$, the second step results in the following formula:

$$\mathbb{E}^{\text{const}}_{\mathrm{P}_{\mathbf{Y},R}}[\ell(\hat{\mathbf{y}},\mathbf{y})] = \sum_{\mathbf{c}\in\mathbf{Y}} \frac{\mathcal{Z}(\{\mathbf{c}\},R)}{\mathcal{Z}(\mathbf{Y},R)} \ell(\hat{\mathbf{y}},\mathbf{c}) \underbrace{\sum_{\mathbf{y}\in\mathcal{Y}} \mathrm{P}_{\{\mathbf{c}\},R}(\mathbf{y}|\mathbf{x})}_{=1}$$

$$= \sum_{\mathbf{c} \in \mathbf{Y}} \frac{\mathcal{Z}(\{\mathbf{c}\}, R)}{\mathcal{Z}(\mathbf{Y}, R)} \ell(\hat{\mathbf{y}}, \mathbf{c}) \tag{14}$$

This expression looks similar to EMBR, but with masses in place of exponentiated scores. This has an intuitive interpretation: the local probability mass of a solution can potentially be more informative than its score.

In the next section, we describe how we overcome the technical challenges of computing masses and marginals in the Hamming-constrained distributions.

6. Estimating Masses and Marginals in Hamming-Constrained Distributions

In this section, we describe how we estimate mass of a candidate solution along with the Hamming-constrained marginals given a factor graph, the candidate solution $\mathbf{c}$, and a given bin radius R.

To enfore the Hamming constraint, we add a higher-order potential (HOP) to the factor graph, contributing a score θ_{HOP}, that clamps probabilities of all solutions lying outside the Hamming ball of radius R to 0. Thus, the probability of a solution $\mathbf{y}$ in the constrained distribution takes the form

$$\mathrm{P}_{\{\mathbf{c}\},R}(\mathbf{y}) \propto \exp\left\{\sum_{F} \theta_F(y_F) + \theta_{\text{HOP}}(\mathbf{y}; \mathbf{c}, R)\right\}, \tag{15}$$

where the HOP, θ_{HOP}, is defined as follows:

$$\theta_{\text{HOP}}(\mathbf{y}; \mathbf{c}, R) = \begin{cases} -\infty, & \text{if } \Delta^{\text{Ham}}(\mathbf{y}, \mathbf{c}) > R \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

6.1. Imposing Hamming Constraints with the HOP

Since Hamming distance from a solution $\mathbf{c}$ is just the number of nodes disagreeing with $\mathbf{c}$, we make use of cardinality potentials [11] to impose Hamming constraints in our model.

In [22], an efficient method for imposing arbitrary cardinality potentials in a sum-product message passing setting was introduced. A binary tree is constructed on top of nodes on which some arbitrary cardinality potential is intended to be imposed. Intermediate nodes in the tree compute the sum of cardinalities of their children, such that at the root node, we can compute the beliefs for the number of leaf nodes that are ON (see Fig. 2a). An arbitrary cardinality potential can be imposed on the root node, and passing messages down to the leaves will give us revised beliefs.

Hamming distance in binary graphs. For graphs with nodes taking binary labels, the Hamming distance between solutions $\mathbf{y}$ and $\mathbf{c}$ is the sum of the count of OFF bits in $\mathbf{y}$ where there are ON bits in $\mathbf{c}$ (denoted by C_0^1 in Fig. 2b) and the count of ON bits in $\mathbf{y}$ where there are OFF bits in $\mathbf{c}$ (denoted by C_1^0). As shown in Fig. 2b, we impose the required cardinality potential to disallow solutions outside the Hamming radius, by constructing the binary tree such that the two subtrees beneath the root node compute the two counts C_0^1 and C_1^0, and the root node sums these counts. Thus, the root node gives us the total count of disagreements between $\mathbf{y}$ and $\mathbf{c}$. If we now set the cardinality potential to 0 when the count is $\leq R$, and $-\infty$ otherwise, we can impose a hard constraint that only allows $\mathbf{y}$ within radius R of $\mathbf{c}$.

Hamming distance in multilabel graphs. We first expand the multi-label graph into an equivalent binary graph. A node taking K labels in the multi-label graph corresponds to K nodes in the binary graph. For every such expanded set of nodes, we impose a 1-of-K potential that forces only one of the nodes to be ON. We construct such a potential by once again using the cardinality tree from [22] such that the only allowed cardinality for any subset of K nodes is 1.

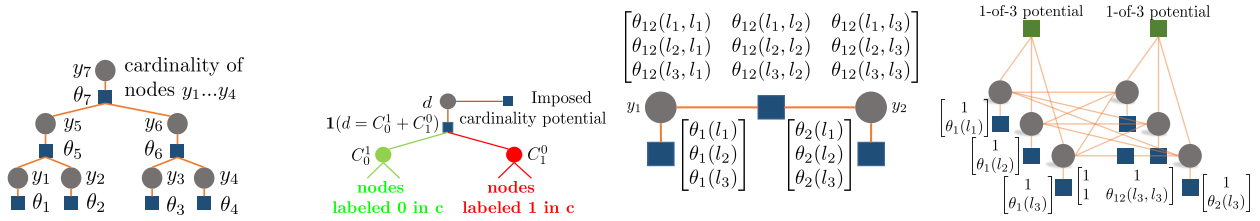
For computing Hamming distance from a solution $\mathbf{c}$ in this model, we only need to compute the number of OFF nodes among the set of nodes that are ON in $\mathbf{c}$, since that gives us the number of disagreements between the current state of the graph and $\mathbf{c}$. In a similar fashion as with the binary case, we impose a Hamming constraint that disallows solutions outside the Hamming ball centered at $\mathbf{c}$.

Hamming constrained marginals and masses: The partition function of this constrained CRF is the mass of the solution $\mathbf{c}$. To estimate the partition function, we first perform sum-product message passing on the HOP-augmented factor graph thus obtaining node and factor marginals. Let $N(i)$ be the number of factors with an edge to node i, $\mu_i(y_i)$ be the node marginal of node i, $\mu_F(y_F)$ be the factor marginal for factor F, and $\theta_F(y_F)$ be the potential for factor F. Then the standard Bethe estimator for the partition function [26] is given as follows:

$$\log \mathcal{Z} = \sum_{i \in \mathcal{V}} (N(i) - 1) \left[\sum_{y_i \in Y_i} \mu_i(y_i) \log \mu_i(y_i)\right] - \sum_{F \in \mathcal{F}} \sum_{y_F \in Y_F} \mu_F(y_F) \left(\log \theta_F(y_F) + \log \mu_F(y_F)\right) \tag{17}$$

For tree-structured graphs, sum-product message passing provides exact marginals, and the Bethe approximation is exact. There are no guarantees in loopy graphs, but results are often reasonable in practice; in the experiments section we compare estimates returned by the Bethe estimator to a sampling-based baseline on small problems and find the Bethe estimator to be efficient in terms of accuracy.

The set of factors in the graph involve the factors in the cardinality tree. The largest factor in the graph (assuming #labels $\leq$ #superpixels) would belong to the cardinality tree, and would be of $O(n^2)$ size (where n is the number



(a) (b) (c) (d)

Figure 2: (a) The state of the root node (y_7) is the cardinality of all the leaf nodes; (b) The root node d counts the Hamming distance from $\mathbf{c}$; (c) A simple 2 node multi-label graph (each node takes 3 labels $\{l_1, l_2, l_3\}$); (d) 1-of-K potentials are imposed binary nodes corresponding to the same multi-label variable to ensure that exactly one of them is ON. To avoid clutter, not all factors have not been shown.

of nodes/superpixels), so message passing operations are $O(n^2)$ costly.

7. Experiments

We first perform experiments on synthetic data to confirm that our estimated masses are a good approximation to the true masses. Then, we evaluate the predictors we discussed on two segmentation datasets: Binary (foreground-background) segmentation and category-level semantic segmentation on PASCAL VOC 2012 `val` [8]. In each case, we compare the following predictors:

1. `MAP` (over $\mathcal{Y}$): The Maximum a Posteriori prediction is a natural baseline for MBR methods. It is straightforward to show that the MAP predictor is an MBR predictor assuming 0-1 loss rather than the task loss.

2. `CRF-EIoEU+greedy` (over $\mathcal{Y}$): Greedy optimization of EIoEU. [3]

3. `CRF-EIoEU+enum` (over $\mathbf{Y}$): Optimization of EIoEU via enumeration over the candidate set.

4. `Delta-EIoU+enum` (over $\mathbf{Y}$): Optimization of EIoU (averaged over multiple delta distributions) via enumeration over the candidate set. This is the approach of [20] (2).

5. `Mass-EIoU+enum` (over $\mathbf{Y}$): Optimization of EIoU (averaged over multiple delta distributions) via enumeration over the candidate set; but the averaging uses masses of candidates instead of their scores (14).

6. `C`^{3}`RF-EIoEU+enum` (over $\mathbf{Y}$): Optimization of EIoEU (averaged over Hamming ball constrained distributions) via enumeration over the candidate set (13).

The parameters involved for the models are: an implicit temperature parameter, T, associated with the Gibbs distribution; a *diversity* parameter, λ, used for computing the set of candidate solutions as in [2]; a radius parameter, R, for the constrained distributions. The best parameters are chosen via cross-validation for each of the six approaches. Approach 2 only learns T, approaches 3 and 4 learn λ and T, and approaches 5 and 6 learn λ, T, and R.

It should be noted that the experimental setups we use from [20] are not designed to be well-calibrated, *i.e.* they are not trained to encode meaningful beliefs over the space of solutions (which is implicitly assumed in applying BDT). We believe this is a more practical scenario, since it is typically hard to learn a well-calibrated model for problems like segmentation in a structured prediction setting.

7.1. Estimating masses for toy graphs

As a sanity check, we used our method for computing mass around solutions on small toy $N \times N$ grids, where true masses can be computed via brute-force enumeration. The unary and pairwise log-potentials are randomly set in the interval $(-\infty, 0]$. We also sample a random (small) radius value ranging from 1 to $\sqrt{N}$. Fig. 3a shows absolute error in log-mass estimation, averaged across 10 runs, for Bethe and uniform sampling in the Hamming ball. We can see that the Bethe approximation works very well, while sampling takes a large number of samples, even in such a small graph.

7.2. Interactive Binary Segmentation

We use the foreground-background setup from [2] – 100 images were taken from the PASCAL VOC 2010 dataset, and manually annotated with scribbles for objects of interest.

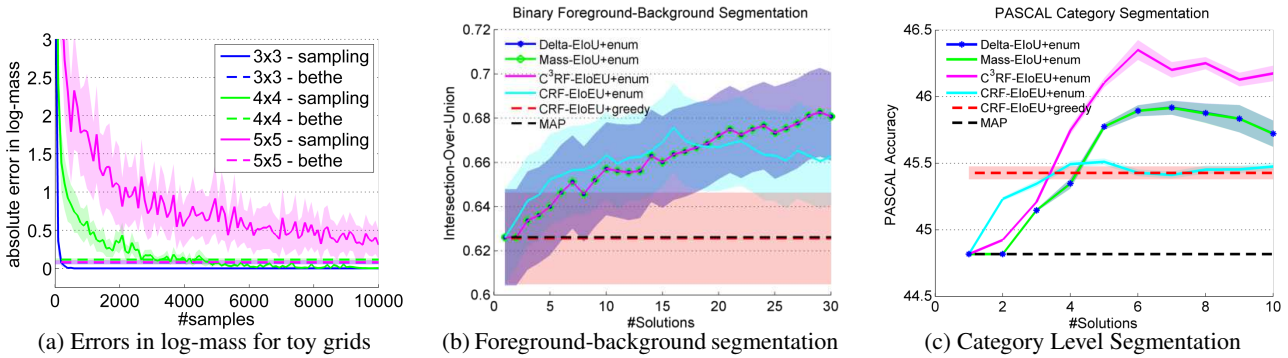
CRF construction: For every superpixel in the image, outputs of Transductive SVMs are used as node potentials, which along with constrast sensitive Potts edge potentials provide the binary CRF to be used for segmentation.

Candidate Solutions: We run DivMBest on the CRF to acquire a set of 30 diverse solutions per test image.

Evaluation: 50 of these images were used to train the SVM parameters, and Leave-One-Out cross-validation is performed on the remaining 50 images. The validation objective is the performance at $M = 30$ for 49 images.

Results: In Fig. 3b, we show how the various predictors perform. The shading around the curves indicate standard error bars – *i.e.* standard deviation/$\sqrt{\#\text{folds}}$ for the 50 Leave-one-out curves. We observe that `Delta-EIoU+enum`, `Mass-EIoU+enum`, and `C`^{3}`RF-EIoEU+enum` perform exactly

[3] We use the code from `http://tinyurl.com/os7coq8` for implementing this optimization.



(a) Errors in log-mass for toy grids (b) Foreground-background segmentation (c) Category Level Segmentation

Figure 3: (a) Errors for the grids are approximately the same for the Bethe approximation. Sampling requires more samples with increasing grid size to provide a good estimate. (b) All EMBR predictors improve significantly upon MAP and the greedy optimization. `Mass-EIoU+enum` and `C3RF-EIoEU+enum` opt for a radius of 0 and reduce to `Delta-EIoU+enum`. `CRF-EIoEU+enum` performs relatively poorly. (c) `C3RF-EIoEU+enum` performs the best, with a $\sim 1.4\%$ improvement over MAP. Again, `Mass-EIoU+enum` performs similarly as `Delta-EIoU+enum`, and `CRF-EIoEU+enum` performs relatively poorly.

the same: 5.45% above MAP. This is because `Mass-EIoU+enum` and `C3RF-EIoEU+enum` both pick a Hamming radius of 0 during cross-validation, and reduce to `Delta-EIoU+enum`. `CRF-EIoEU+enum` appears to perform well at the start, however, at $M = 30$, it performs poorly: 3.63% above MAP. `CRF-EIoEU+greedy` [18] performs slightly worse than `MAP` but within error bars.

7.3. PASCAL Category Segmentation

We evaluate our predictors on the task of category segmentation - label each pixel with one of a set of categories - on the PASCAL VOC 2012 `val` set.

CRF Construction: Multi-label pairwise CRFs are constructed on superpixels of images from PASCAL VOC 2012 `train` and `val`. The node potentials are outputs of category-specific regressors, which are trained on `train` using [5]. Edge potentials are multi-label Potts.

Candidate Solutions: We use solutions generated as in [20] - CPMC segments [6] scored by Support Vector Regressors over second-order pooled features [5] are greedily pasted. We observed that DivMBest solutions have duplicates, so we pick the first 10 unique solutions from a set of 50 DivMBest solutions. In the few cases where there aren't 10 unique solutions among the 50 diverse solutions, we allow duplicates to remain, the intuition being that since these solutions are repeatedly picked by the DivMBest algorithm, they represent exceptionally strong beliefs of the model, and allowing these duplicates to remain would assign them a larger weight.

Evaluation: The standard PASCAL evaluation criteria is the corpus-level IoU averaged over all 21 categories. Since we are performing instance level predictions, and then evaluating at a corpus level, the loss function used in all predictors is instance level loss, but the parameters are chosen by cross-validation on corpus-level loss.

We perform 10 fold cross-validation on `val`. To acquire a clearer sense of variance, we perform the 10 fold cross-val for 5 different random permutations of the corpus.

Results: Fig. 3c compares performances of the various predictors. The shading around the curves indicates standard error bars across the 5 different permutations. We again observe that `Delta-EIoU+enum` and `Mass-EIoU+enum` perform very similarly - 0.9% above MAP. `CRF-EIoEU+greedy` [18] performs 0.62% above `MAP`, but is the poorest among the other predictors.

Again, `CRF-EIoEU+enum` is relatively promising at the start, but performs poorest at $M = 10$: 0.68% above `MAP`. `C3RF-EIoEU+enum`, performs the best, with a 1.35% improvement over `MAP`, and 0.45% improvement over `Delta-EIoU+enum`, with no overlap in error bars.

Note that in our current setup, there may be overlap in Hamming balls across the clamped distributions. We also ran a set of experiments where we shrink radii around each solution till there is no overlap. We found that this results in reduced performance ($\approx 0.1\%$ below MAP).

7.4. Discussion

We now discuss take-away messages from our experiments.

Search heuristic We notice that `CRF-EIoEU+enum` consistently outperforms `CRF-EIoEU+greedy`, which suggests that optimizing EIoU for a set of candidate solutions with a high "oracle accuracy" (the accuracy of the best performing solution in the set) as performed by [20] is a more practical heuristic. We also observe that the average EIoEU score achieved by the greedy heuristic is 0.0739 while `enum` reaches a higher score of 0.0746 using the same set of marginals.

`Mass-EIoU+enum` reduces to `Delta-EIoU+enum`: We note that cross-validation picks `Delta-EIoU+enum` over `Mass-EIoU+enum` almost all of the time (by opting for a Ham-

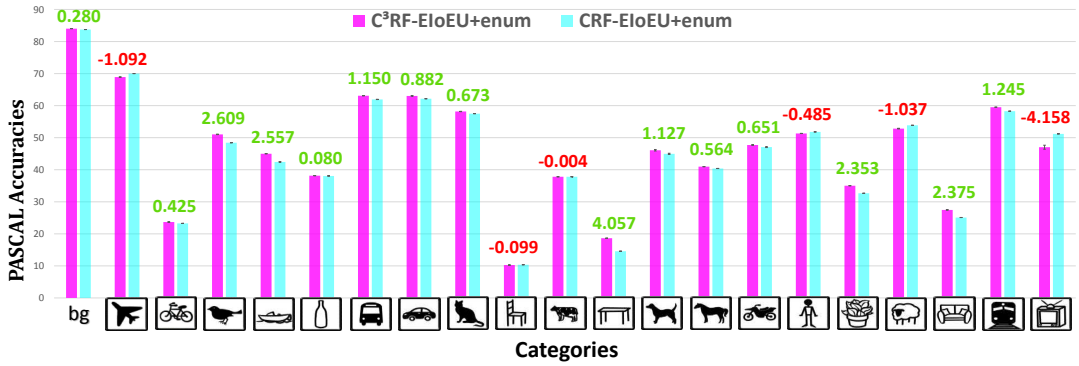


Figure 4: Comparing classwise accuracies for C³RF-EIoEU+enum and CRF-EIoEU+enum. The numbers on top of the bars indicate the difference in accuracy between C³RF-EIoEU+enum and CRF-EIoEU+enum.

ming radius of 0). We hypothesize that this is because Mass-EIoU+enum makes a crude assumption that worsens the approximation quality with increasing radii. As we showed in Sec. 5.1, the Mass-EIoU+enum predictor essentially assumes that the IoU of all solutions in a Hamming ball are the same. Since we are labeling superpixels, flipping even a small number of superpixels typically results in IoUs that are significantly different from the solution we form the Hamming ball around, especially since we evaluate IoU on a pixel level.

C³RF-EIoEU+enum vs. Delta-EIoU+enum: Given that Delta-EIoU+enum is less time-consuming than C³RF-EIoEU+enum, an accuracy/time trade-off has to be made when selecting one over the other. It is natural to consider increasing the number of solutions for Delta-EIoU+enum to see whether the performance difference between Delta-EIoU+enum and C³RF-EIoEU+enum is bridged at a larger M. Loopy BP takes ∼8 seconds per Hamming-constrained distribution. Fortunately, this can be conducted in parallel across the candidate solutions. The optimization for EIoU takes roughly the same time for both Delta-EIoU+enum and C³RF-EIoEU+enum (around 10 seconds). We run Delta-EIoU+enum for the same duration of time that it takes C³RF-EIoEU+enum to run for 10 solutions, and find that Delta-EIoU+enum does not reach higher performance than C³RF-EIoEU+enum at $M = 10$ (Fig. 5).

8. Conclusion

We have shown that combining ideas from [18, 21] and [20] about performing loss-aware structured prediction with the IoU measure leads to improved performance on two image segmentation tasks. A natural question to ask in light of the analysis presented in [18] is why there is room for improvement over the greedy optimization of EIoEU. We can attribute some of the gains to our observation that the candidate solutions do a better job of optimizing the EIoEU objective. However, we speculate that there could be an

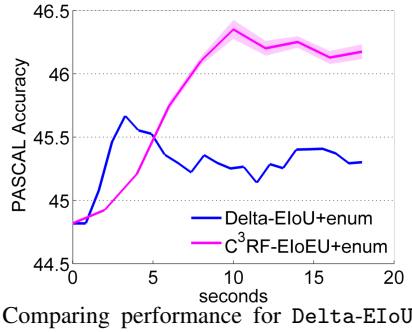


Figure 5: Comparing performance for Delta-EIoU+enum and C³RF-EIoEU+enum across time.

additional effect that comes from the fact that the models we work with have not been trained to produce calibrated probabilities (though we emphasize that there is a degree of calibration happening within our hyperparameter search loop). While one could insist upon only using models with calibrated probabilities, the reality is that doing so requires significant sacrifice in terms of the richness of the models that can then be used. We operate under the assumption that we are not willing to make this sacrifice. Interestingly, as we show, there is still room for ideas from classical Bayesian decision theory albeit applied in an irreverent manner - rather than strictly adhering to the prescription of BDT, it is beneficial to mix modern decision theoretic algorithms with heuristics that work well across vision tasks.

Acknowledgements. We thank Sebastian Nowozin for making his code publicly available. This work was done while FA was an intern at VT. This work was partially supported by a National Science Foundation CAREER award, an Army Research Office YIP award, and the Office of Naval Research grant N00014-14-1-0679, all awarded to DB. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government or any sponsor.

References

[1] D. Batra. An Efficient Message-Passing Algorithm for the M-Best MAP Problem. In *Uncertainty in Artificial Intelligence*, 2012.

[2] D. Batra, P. Yadollahpour, A. Guzman-Rivera, and G. Shakhnarovich. Diverse M-Best Solutions in Markov Random Fields. In *ECCV*, 2012.

[3] J. Berger. *Statistical decision theory and Bayesian analysis*. Springer, New York, NY [u.a.], 2. ed edition, 1985.

[4] G. Bouchard and O. Zoeter. Split variational inference. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, pages 57–64, 2009.

[5] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, pages 430–443, 2012.

[6] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010.

[7] G. Csurka, D. Larlus, F. Perronnin, and F. Meylan. What is a good evaluation measure for semantic segmentation? In *24th British Machine Vision Conference (BMVC)*, Sept 2013.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/?q=node/874.

[9] A. C. Gallagher, D. Batra, and D. Parikh. Inference for order reduction in markov random fields. In *CVPR*, 2011.

[10] K. Gimpel, D. Batra, C. Dyer, and G. Shakhnarovich. A systematic exploration of diversity in machine translation. In *EMNLP*, 2013.

[11] R. Gupta, A. A. Diwan, and S. Sarawagi. Efficient inference with cardinality-based clique potentials. In *Proceedings of the 24th international conference on Machine learning*, pages 329–336. ACM, 2007.

[12] T. Jaakkola and M. Jordan. Improving the mean field approximation via the use of mixture distributions. In M. Jordan, editor, *Learning in Graphical Models*, volume 89 of *NATO ASI Series*, pages 163–173. Springer Netherlands, 1998.

[13] P. Krähenbühl and V. Koltun. Parameter learning and convergent inference for dense random fields. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[14] S. Kumar and W. Byrne. Minimum bayes-risk decoding for statistical machine translation. Johns Hopkins Univ. Baltimore MD, Center for Language and Speech Processing (CLSP), 2004.

[15] S. Kumar, W. Macherey, C. Dyer, and F. Och. Efficient minimum error rate training and minimum bayes-risk decoding for translation hypergraphs and lattices. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 1. Association for Computational Linguistics, 2009.

[16] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. Technical report, Cambridge, MA, USA, 1987.

[17] T. Minka and J. Winn. Gates: A graphical notation for mixture models. In *In Proceedings of NIPS*, volume 21, pages 1073–1080, 2008.

[18] S. Nowozin. Optimal decisions from probabilistic models: the intersection-over-union case. In *CVPR*, 2014.

[19] P. Pletscher and P. Kohli. Learning low-order models for enforcing high-order statistics. *AISTATS*, 2012.

[20] V. Premachandran, D. Tarlow, and D. Batra. Empirical Minimum Bayes Risk Prediction: How to extract an extra few% performance from vision models with just three more parameters. In *CVPR*, 2014.

[21] D. Tarlow and R. Adams. Revisiting uncertainty in graph cut solutions. In *CVPR*, 2012.

[22] D. Tarlow, K. Swersky, R. S. Zemel, R. P. Adams, and B. J. Frey. Fast exact inference for recursive cardinality models. In *UAI*, 2012.

[23] D. Tarlow and R. S. Zemel. Structured output learning with high order loss functions. In *AISTATS*, 2012.

[24] R. W. Tromble, S. Kumar, F. Och, and W. Macherey. Lattice minimum bayes-risk decoding for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008.

[25] A. Wald. Statistical decision functions. *Ann. Math. Statist.*, 20(2):165–205, 06 1949.

[26] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Trans. on Information Theory*, 51(7), July 2005.