Empirical Minimum Bayes Risk Prediction

Vittal Premachandran, Daniel Tarlow, Alan L. Yuille, and Dhruv Batra

Abstract—When building vision systems that predict structured objects such as image segmentations or human poses, a crucial concern is performance under task-specific evaluation measures (e.g., Jaccard Index or Average Precision). An ongoing research challenge is to optimize predictions so as to maximize performance on such complex measures. In this work, we present a simple meta-algorithm that is surprisingly effective – *Empirical Min Bayes Risk*. EMBR takes as input a pre-trained model that would normally be the final product and learns three additional parameters so as to optimize performance on the complex instance-level high-order task-specific measure. We demonstrate EMBR in several domains, taking existing state-of-the-art algorithms and improving performance up to 8 percent, simply by learning three extra parameters. Our code is publicly available and the results presented in this paper can be replicated from our code-release.

Index Terms—Diverse predictions, DivMBest, image segmentation, object segmentation, human pose estimation

1 Introduction

Consider the following problem: given an input image $\mathbf{x}$ and a black-box segmentation model that assigns a score $S(\mathbf{y};\mathbf{x})$ to segmentations $\mathbf{y}$ of the image, choose a segmentation so as to maximize performance on a task-specific evaluation measure. Which segmentation should we output? This work argues that the popular choice of picking the segmentation with the highest score is not necessarily the best decision.

Broadly speaking, the de-facto approach today in computer vision for modeling structured objects (such as segmentations and poses) is to

1) formulate a model where parameters determine a scoring function,
2) choose parameters that optimize performance on a training set, and,
3) predict the configuration that (approximately) maximizes the scoring function at test time.

While this seems like an obvious and reasonable workflow, in this paper, we show how to take models that are the final product of such workflows and improve performance. With little additional effort, we take existing published models and extract additional performance gains of up to 8 percent.

The motivation for our approach comes from Bayesian decision theory, which gives a principled methodology for making decisions in the face of uncertainty and task-specific evaluation measures. The key aspects of Bayesian decision theory are to optimize expected performance on the measure of interest while averaging over uncertainty. Specifically, the prescription of Bayesian decision theory is as follows: (1) learn a model that gives accurate probabilities $\mathrm{P}(\mathbf{y}|\mathbf{x})$, then (2) make predictions so as to minimize expected loss under the learned distribution, i.e., $\hat{\mathbf{y}}^{MBR} = \min_{\hat{\mathbf{y}}} \mathbb{E}_{\mathrm{P}(\mathbf{y}|\mathbf{x})}[\ell(\mathbf{y},\hat{\mathbf{y}})] = \min_{\hat{\mathbf{y}}} \sum_{\mathbf{y}} \mathrm{P}(\mathbf{y}|\mathbf{x})\ell(\mathbf{y},\hat{\mathbf{y}})$, where $\ell(\cdot,\cdot)$ is the task-specific loss function of interest, which assigns loss $\ell(\mathbf{y},\hat{\mathbf{y}})$ to predicting $\hat{\mathbf{y}}$ when the ground truth is $\mathbf{y}$. We refer to this as the *Minimum Bayes Risk* (MBR) predictor. Unfortunately, a direct application of this elegant classical theory to most structured prediction problems is intractable due to computational consideration – specifically, the output space of $\mathbf{y}$ is too large to perform the summation and the minimization. For example, in scenarios such as image segmentation and pose estimation, the output space is exponentially large, making the summation and minimization intractable in general.

We show that it is nonetheless possible to efficiently construct structured predictors that incorporate the key ingredients – (a) incorporating task-specific losses and (b) averaging over uncertainty. Concretely, we will take as input the trained scoring function $S(\mathbf{y};\mathbf{x})$ from a given model and produce a set of M *plausible* candidate solutions $\mathbf{Y_M} = \{\mathbf{y}^1,\ldots,\mathbf{y}^M\}$ along with a probability distribution over these candidates. Decisions are made by employing the MBR predictor but restricting the optimization and summation required to the M candidate solutions. The surprising observation of this work is that this formulation can produce substantial gains over state-of-the-art performance while parameterizing this meta-model with only three parameters –

1) M, the number of candidate solutions,
2) T, which controls the scale (or "temperature") of $S(\cdot)$,
3) λ, which determines the amount of *diversity* to impose when generating the M candidates.

In addition, a further minor improvement in performance can be achieved by introducing robustness to our model, which can simply be attained by the use of a threshold parameter, α.

Crucially, while our method is motivated by decision theory principles and resembles MBR, it is actually an instance of Empirical Risk Minimization (ERM) – our goal is not to approximate Bayes Risk (BR), rather to train a predictor that utilizes ideas from decision theory and performs well on a

- *V. Premachandran and A. Yuille are with the Johns Hopkins University, Baltimore, MD. E-mail: vittalp@jhu.edu, yuille@stat.ucla.edu.*
- *D. Tarlow is with the Microsoft Research. E-mail: dtarlow@microsoft.com.*
- *D. Batra is with Virginia Tech. E-mail: dbatra@vt.edu.*

Manuscript received 29 May 2015; revised 14 Dec. 2015; accepted 22 Feb. 2016. Date of publication 2 Mar. 2016; date of current version 12 Dec. 2016.
Recommended for acceptance by S. Roth.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2016.2537807

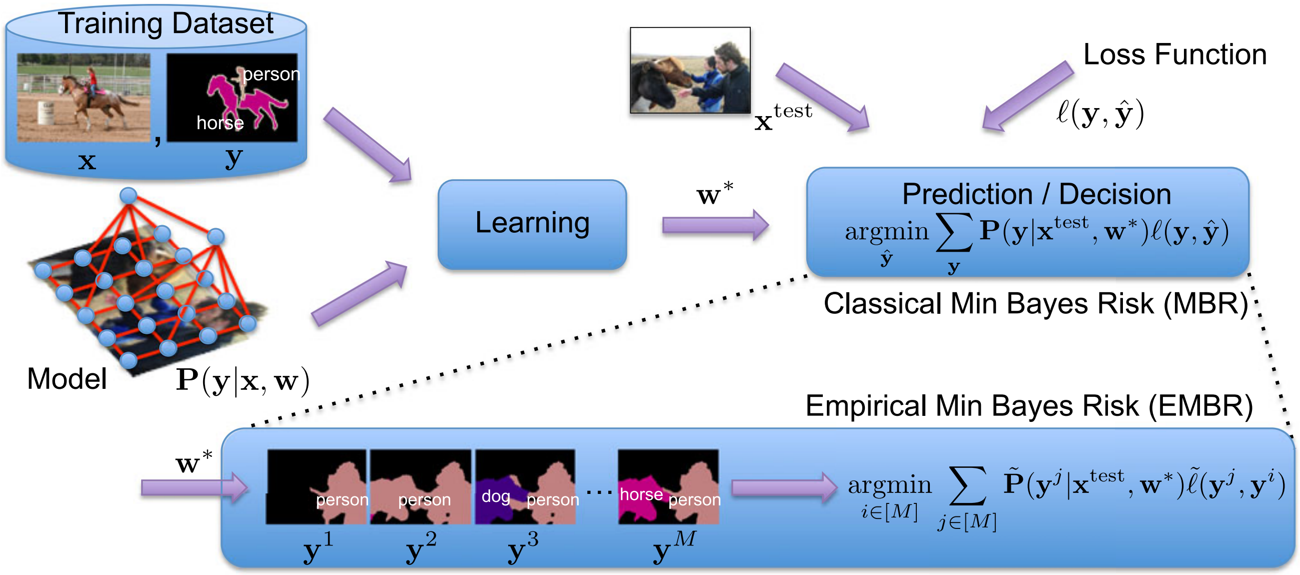

Training Dataset
person
horse
x
y
Model
P(y|x, w)
Learning
w*
x^test
Loss Function
ℓ(y, ŷ)
Prediction / Decision
argmin_ŷ Σ_y P(y|x^test, w*)ℓ(y, ŷ)
Classical Min Bayes Risk (MBR)
Empirical Min Bayes Risk (EMBR)
w*
person
person
dog
person
horse
person
y^1
y^2
y^3
y^M
argmin_{i∈[M]} Σ_{j∈[M]} P̃(y^j|x^test, w*)ℓ̃(y^j, y^i)

Fig. 1. Classical Min Bayes Risk (MBR) versus Empirical Min Bayes Risk (EMBR): Probabilistic reasoning involves (a) learning the parameters of our model from training data, and (b) making predictions or decisions by optimizing *Bayes Risk* or expected loss. We present a meta-algorithm (EMBR) that is motivated by MBR but is instead based on Empirical Risk Minimization principle.

held-out dataset. The three parameters are learned by performing grid search and choosing the setting that minimizes empirical risk. Thus, we call the method Empirical Minimum Bayes Risk (EMBR) prediction. Fig. 1 illustrates Classical Min Bayes Risk (MBR) versus Empirical Min Bayes Risk (EMBR).

Contributions. We develop a simple and efficient meta-algorithm that is inspired by Bayesian decision theory and which inherits some of the improvements in accuracy that the framework promises, but which is computationally efficient, simple to implement, agnostic to the loss function being used, and can be applied to models which have already been trained, *even ones not trained in a probabilistic framework* (e.g., Structured SVMs).

Our experiments on a range of problems –

1) binary foreground-background segmentation,
2) human body pose estimation, and
3) semantic object-category segmentation

show that the proposed approach consistently improves performance over the input model, indicating that this is a simple but effective way of improving performance by incorporating task loss into the prediction procedure. As an example, by applying our methodology to the publicly available pre-trained pose estimation models of Yang and Ramanan [33], we achieved state-of-art accuracies on the PARSE dataset, improving results by about 8 percent.

Preliminary versions of this idea have been presented in [26]. This manuscript presents new discussions and new experiments.

New discussions:

1) We introduce a robust version of the EMBR loss function that is now able to handle the adverse effects of outliers in the proposal set. The preliminary version presented in [26] did not handle this case.
2) We provide a detailed description of the relation between the EMBR predictor and the MAP predictor and list out various cases in which the EMBR predictor degenerates to the MAP predictor.

New experiments:

1) The preliminary version in [26] reported the performance of EMBR while using DivMBest as the proposal generator. In this manuscript, we also report the performance of EMBR when used in conjunction with a different proposal generator, namely, the Perturb-and-MAP [20].
2) We compare the performance of EMBR when the parameters are tuned on two types of metrics; (a) corpus-level metric and (b) instance-level metric on the PASCAL semantic segmentation experiments.
3) We also report results obtained when using a robust version of the EMBR loss that can better-handle the adverse effects of outliers that might be present in the proposal set.
4) We also test the performance of the EMBR predictor on the now state-of-the-art algorithm for pose estimation [7], which makes use of features from a deep network and an Image Dependent Pairwise Relation term in the model.

2 Preliminaries

We begin by establishing notation before reviewing two standard approaches to structured prediction: the probabilistic approach, and the empirical risk minimization approach.

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 $\mathbf{y} \in \mathcal{Y}$, where $\mathbf{y}$ may be a foreground-background segmentation, or location of body poses of a person in the image, or a category-level semantic segmentation. Specifically, let $\mathbf{y} = \{y_1 \ldots y_n\}$ be a set of discrete random variables, each taking value in a finite label set, $y_u \in \mathcal{Y}_u$. In the semantic segmentation experiments, u indexes over the (super-)pixels in the image, and these variables are the labels assigned to each (super-)pixel, i.e., $y_u \in \mathcal{Y}_u = \{\text{sky}, \text{building}, \text{road}, \text{car}, \ldots\}$. In the pose estimation experiments, u indexes over body parts (head, torso, right arm, etc.), and each variable indicates the (discretized) location of the body part in the image plane.

The quality of predictions is determined by a loss function $\ell(\mathbf{y}^{gt}, \hat{\mathbf{y}})$ that denotes the cost of predicting $\hat{\mathbf{y}}$ when the ground-truth is $\mathbf{y}^{gt}$. In the context of semantic segmentation, this loss might be the PASCAL loss [10] $1 - \frac{\text{intersection}}{\text{union}}$ measure, averaged over masks of all categories. In the context of pose estimation, this loss might be the Average Precision of Keypoints (APK) measure proposed by Yang and Ramanan [33].

2.1 Probabilistic Structured Prediction

Score. A common approach to probabilistic structured prediction is to base the model on a *score* function $S(\mathbf{y}; \mathbf{x}, \mathbf{w})$ which assigns a score to configurations $\mathbf{y}$ (later, we will drop the explicit dependence on $\mathbf{w}$ for notational simplicity). The probability of any configuration is given by the Gibbs distribution: $\mathrm{P}(\mathbf{y}|\mathbf{x}), = \frac{1}{\mathcal{Z}} e^{S(\mathbf{y};\mathbf{x})}$, where $\mathcal{Z}$ is the partition function. In this work, we will make minimal assumptions about the structure of the scoring function.

Assumptions. Our key computational assumption is that it is possible to efficiently compute $\mathrm{argmax}_{\mathbf{y}}\, S(\mathbf{y}; \mathbf{x})$ (or a good approximation) using algorithms such as graph cuts or α-expansion, but that it is not possible to compute probabilities $\mathrm{P}(\mathbf{y}|\mathbf{x})$ or expectations. This assumption is fairly typical in modeling structured outputs, and is born out of the difference in hardness of maximization versus summation over exponentially large spaces. For instance, a maximum bipartite matching can be found in $O(n^3)$ time with the Hungarian algorithm [16], but summing over all perfect matchings (i.e., computing the permanent) is #P-complete [31]. The only other assumption that we make is that we can modify unary potentials $\theta_u(y_u)$ and tractably optimize a unary-augmented score function, i.e., $\mathrm{argmax}_{\mathbf{y}}\, S(\mathbf{y}; \mathbf{x}) + \sum_u \theta_u(y_u)$, which allows leveraging methods such as DivMBest [4] and Perturb-and-MAP [20] to find a set of solutions by solving multiple maximization problems where $S(\cdot)$ is augmented with perturbed (either with structured diversity, or with Gumbel noise) unary potentials. Strictly speaking, our approach has no requirement that the scoring function have unary potentials, simply that it be able to perform unary-augmented score maximization. However, for the sake of exposition, it is convenient to assume that unaries are present (initialized to 0 if absent from the model). In practice, all models used in our experiments are composed of two parts; a unary term and a pairwise term.

MAP predictor. The MAP predictor finds the labeling $\mathbf{y}$ that maximizes the probability or score:

$$\mathbf{y}^{MAP} = \underset{\mathbf{y}\in\mathcal{Y}}{\mathrm{argmax}}\ S(\mathbf{y}; \mathbf{x}). \tag{1}$$

This problem is NP-hard in general [27]. Thus, most works focus on exact inference for certain subclasses, e.g., when the graph G is a tree or the scoring function is supermodular, MAP can be computed optimally via highly efficient algorithms – dynamic-programming [23] and max-flow/min-cut [13], [15], respectively.

MBR predictor. The Bayes Risk of predicting $\hat{\mathbf{y}}$ is defined as

$$\mathrm{BR}(\hat{\mathbf{y}}) = \mathbb{E}_{\mathrm{P}(\mathbf{y}|\mathbf{x})}[\ell(\mathbf{y}, \hat{\mathbf{y}})] = \sum_{\mathbf{y}\in\mathcal{Y}} \mathrm{P}(\mathbf{y}|\mathbf{x})\ell(\mathbf{y}, \hat{\mathbf{y}}). \tag{2}$$

This is the expected cost of predicting $\hat{\mathbf{y}}$ under loss function $\ell(\cdot,\cdot)$ when the annotations come from the distribution $\mathrm{P}(\mathbf{y}|\mathbf{x})$. The Minimum Bayes Risk predictor is one that minimizes this expected risk, i.e.,

$$\mathbf{y}^{MBR} = \underset{\hat{\mathbf{y}}\in\mathcal{Y}}{\mathrm{argmin}}\ \mathrm{BR}(\hat{\mathbf{y}}) \tag{3a}$$

$$= \underset{\hat{\mathbf{y}}\in\mathcal{Y}}{\mathrm{argmin}} \sum_{\mathbf{y}\in\mathcal{Y}} \mathrm{P}(\mathbf{y}|\mathbf{x})\ell(\mathbf{y}, \hat{\mathbf{y}}). \tag{3b}$$

Intuitively, MBR assumes that any configuration $\mathbf{y}$ could be the ground-truth annotation with probability given by $\mathrm{P}(\mathbf{y}|\mathbf{x})$, and decides to hedge against uncertainty by minimizing an average loss. Note that the MAP predictor is the MBR predictor when the loss function is 0-1, i.e., assigns zero cost if $\hat{\mathbf{y}}$ is equal to $\mathbf{y}^{gt}$ and constant cost otherwise. Also notice that performing exact MBR prediction is in general "doubly intractable"[1] because the summation and minimization are both over exponentially large choices (e.g., all possible segmentations).

2.2 Empirical Risk Minimization

An alternative inductive principle is Empirical Risk Minimization. In this view, we define a predictor function $f(\mathbf{x}; \mathbf{w})$ that maps an input $\mathbf{x}$ to an output $\mathbf{y}$ and is parameterized by $\mathbf{w}$. The goal is then simply to choose parameters that minimize empirical risk, which is often chosen to be the loss function of interest (if tractable), or some approximation to it (e.g., structured hinge loss).

In a common instantiation in computer vision, the form of $f(\mathbf{x}; \mathbf{w})$ is chosen to *resemble* the MAP predictor,

$$f(\mathbf{x}; \mathbf{w}) = \underset{\mathbf{y}}{\mathrm{argmax}}\, S(\mathbf{y}; \mathbf{x}, \mathbf{w}). \tag{4}$$

That is, $\mathbf{w}$ is used to construct a scoring function $S(\mathbf{y}; \mathbf{x}, \mathbf{w})$, and then the output is set to be the $\mathbf{y}$ that maximizes the scoring function.

Note that in this case, the scoring function can still be exponentiated and normalized to produce a distribution over $\mathbf{y}$, but it will not in general correspond to meaningful beliefs about the values that $\mathbf{y}$ is likely to take on; the sole concern in setting $\mathbf{w}$ is to minimize empirical risk, and we emphasize that there is no reason to believe that a setting of $\mathbf{w}$ that has low empirical risk will also yield a sensible distribution over $\mathbf{y}$. To make this point explicit, when dealing with such probability distributions, which are valid distributions but were not trained to correspond to beliefs about configurations, we will use the notation $\tilde{\mathrm{P}}(\cdot)$.

[1] The term "doubly intractable" comes from analogy to Murray et al. [17] where it is used to mean distributions that involve two exponential sums for which we don't have good dynamic programming solutions. Formally, the MBR problem is instance of 'marginal-MAP' inference, where a subset of variables are marginalized ($\mathbf{y}$ in our notation), and another subset are maximized/minimized ($\hat{\mathbf{y}}$ in our notation). Marginal-MAP queries in general graphical models are NP^{PP}-complete [22], and remain hard even when MAP inference is easy (for instance Marginal-MAP is still NP-hard for trees, and cannot even be effectively approximated).

3 Approach: Empirical MBR

The approach that we take in this paper follows the empirical risk formulation as described in the previous section, but rather than defining the prediction function to resemble the MAP predictor, we define it to resemble an MBR predictor.

As mentioned previously, straightforwardly employing the MBR predictor is intractable because the summation and minimization are both over exponentially large choices of $\mathbf{y} \in \mathcal{Y}$. Our decision that leads to tractability is to restrict both the sum and the minimization to be over a set of M strategically chosen solutions $\mathbf{Y_M} = \{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^M\}$. Specifically, the predictor is defined as follows:

$$\mathbf{y}^{EMBR} = \operatorname*{argmin}_{\hat{\mathbf{y}} \in \mathbf{Y_M}} \sum_{\mathbf{y} \in \mathbf{Y_M}} \tilde{\mathrm{P}}(\mathbf{y}|\mathbf{x}) \tilde{\ell}(\mathbf{y}, \hat{\mathbf{y}}), \tag{5}$$

where we call $\mathbf{y}^{EMBR}$ the Empirical MBR (EMBR) prediction, $\tilde{\mathrm{P}}(\cdot)$ is a probability distribution over the M configurations, and $\tilde{\ell}(\cdot,\cdot)$ is any loss function that is used by the EMBR predictor. While it is natural to set the EMBR-loss to be the same as the task-loss, $\tilde{\ell}(\cdot,\cdot) = \ell(\cdot,\cdot)$, this is not strictly necessary in our formulation. Moreover, in some situations, it may not be desirable (see [29, Section 4.7] for an example from information retrieval), or possible (task-loss might be a *corpus-level loss*, e.g., PASCAL Segmentation criteria can only be computed on a dataset, not an individual image – see Section 5 for more discussions regarding this point).

We describe how to construct the candidate set of solutions, $\mathbf{Y_M}$, and their probabilities, $\tilde{\mathrm{P}}(\cdot)$, in the next subsections. The full EMBR algorithm is given in Algorithm 1.

Algorithm 1. Empirical Minimum Bayes Risk Prediction

```
Input: Score function $S(\mathbf{y};\mathbf{x})$, loss $\tilde{\ell}(\cdot,\cdot)$.
Input: Validation-selected parameters $M, T, \lambda$.
{Multiple Solution Generation: E.g. DivMBest}
for $m \in 1, \ldots, M$ do
    $S^m_{\Delta}(\mathbf{y}) \leftarrow S(\mathbf{y}) + \sum_{u \in \mathcal{V}} \sum_{m'=1}^{m-1} \lambda \cdot$
    $\mathbf{y}^m \leftarrow \mathrm{argmax}_{\mathbf{y}}\, S^m_{\Delta}(\mathbf{y};\mathbf{x})$
end for
{Scores to Probabilities}
for $i \in 1, \ldots, M$ do
    $\tilde{\mathrm{P}}(\mathbf{y}^i|\mathbf{x}) \leftarrow \frac{\exp\{\frac{1}{T} S(\mathbf{y}^i;\mathbf{x})\}}{\sum_{j=1}^{M} \exp\{\frac{1}{T} S(\mathbf{y}^j;\mathbf{x})\}}$
end for
{Prediction}
$i^* = \mathrm{argmin}_{i \in [M]} \sum_{j \in [M]} \tilde{\mathrm{P}}(\mathbf{y}^j|\mathbf{x}) \tilde{\ell}(\mathbf{y}^j, \mathbf{y}^i)$
return $\mathbf{y}^{i^*}$
```

Computation. If we construct a matrix of pairwise losses:

$$L = \begin{bmatrix} \tilde{\ell}(\mathbf{y}^1,\mathbf{y}^1) & \tilde{\ell}(\mathbf{y}^1,\mathbf{y}^2) & \cdots & \tilde{\ell}(\mathbf{y}^1,\mathbf{y}^M) \\ \tilde{\ell}(\mathbf{y}^2,\mathbf{y}^1) & \tilde{\ell}(\mathbf{y}^2,\mathbf{y}^2) & \cdots & \tilde{\ell}(\mathbf{y}^2,\mathbf{y}^M) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\ell}(\mathbf{y}^M,\mathbf{y}^1) & \tilde{\ell}(\mathbf{y}^M,\mathbf{y}^2) & \cdots & \tilde{\ell}(\mathbf{y}^M,\mathbf{y}^M) \end{bmatrix}, \tag{6}$$

and a vector stacking all approximate probabilities $\mathbf{p} = [\tilde{\mathrm{P}}(\mathbf{y}^1|\mathbf{x}) \ldots \tilde{\mathrm{P}}(\mathbf{y}^M|\mathbf{x})]^\top$, then the EMBR predictor,

$$\mathbf{y}^{EMBR} = \operatorname*{argmin}_{\mathbf{y}^i, i \in [M]} \sum_{j \in [M]} \tilde{\mathrm{P}}(\mathbf{y}^j|\mathbf{x}) \tilde{\ell}(\mathbf{y}^j, \mathbf{y}^i), \tag{7}$$

can be obtained by extracting the minimum element of the vector obtained using a single matrix-vector multiplication, $L\mathbf{p}$.

The runtime of EMBR given $\mathbf{Y_M}$ and $\tilde{\mathrm{P}}(\cdot)$ is $O(M^2)$. In our experiments $M \leq 50$ and the cost of making predictions is not a significant cost in the prediction pipeline.

3.1 Converting Scores to Probabilities

As mentioned previously, our approach in this work is to assume access to a scoring function $S(\mathbf{y};\mathbf{x})$, with minimal assumptions on how it is constructed (manually tuned, or learned so that the MAP predictor has low empirical risk).

We transform this scoring function and use it as the basis for defining $\tilde{\mathrm{P}}(\mathbf{y}|\mathbf{x})$ to be used within an EMBR predictor.

Given $S(\cdot)$ and set of candidates $\mathbf{Y_M}$, perhaps the simplest sensible choice for defining $\tilde{\mathrm{P}}(\mathbf{y}|\mathbf{x})$ is as follows:

$$\tilde{\mathrm{P}}(\mathbf{y}|\mathbf{x}) = \frac{\exp \frac{1}{T} S(\mathbf{y};\mathbf{x})}{\sum_{\mathbf{y}' \in \mathbf{Y_M}} \exp \frac{1}{T} S(\mathbf{y}';\mathbf{x})}, \tag{8}$$

where T is a temperature parameter that determines the peakiness of $\tilde{\mathrm{P}}(\cdot)$. Note that this method of converting non-probabilistic model outputs into probabilities has been studied in the unstructured case; notably, Platt [24] suggests passing learned SVM outputs through a sigmoid function that has a parameter that behaves similarly to our temperature (there is also one additional offset parameter in [24]). Other approaches are possible. For example, [36] suggests using isotonic regression, [35] discusses calibrating Naive Bayes and decision tree classifiers, and [18] looks deeper into re-calibrating outputs using the different methods and applying them to different first-stage classifiers. In future work it would be interesting to explore alternative mappings from $S(\cdot)$ to $\tilde{\mathrm{P}}(\cdot)$ inspired by these works.

All that remains is to specify the candidate set $\mathbf{Y_M}$.

3.2 Producing Diverse High-Scoring Candidates

How should the candidate configurations $\mathbf{Y_M}$ be chosen? In order to be useful, the set of points must provide an accurate summary of the score landscape or the Gibbs distribution, i.e., be high-scoring and diverse. Two common techniques for producing multiple solutions in probabilistic models can be broadly characterized as follows: (1) M-best MAP algorithms [3], [19], [34] that find the top M most probable solutions and (2) sampling-based algorithms [2], [25], [30]. Both these groups fall short for our task. M-Best MAP algorithms do not place any emphasis on diversity and tend to produce solutions that differ only by the assignment of a handful of pixels. Sampling-based approaches typically exhibit long wait-times to transition from one mode to another, which is required for obtaining diversity. Previous works [4], [21] have demonstrated experimentally that Gibbs sampling does not work well for the task of generating a diverse set of high-scoring solutions.

While our approach is applicable to any choice of diverse hypothesis generators, we experimented with the DivMBest algorithm of Batra et al. [4], and the Perturb-and-MAP algorithm of Papandreou and Yuille [20]. Alternatives that we did not experiment with but might be worthwhile exploring in future work are Herding [14], [32] and Multiple Choice Learning [11], [12].

For sake of completeness, we briefly describe both DivMBest and Perturb-and-MAP. More details can be found in [4] and [20].

DivMBest. DivMBest finds diverse M-best solutions incrementally. Let $\mathbf{y}^1$ be the best solution (or MAP), $\mathbf{y}^2$ be the second solution found and so on. At each step, the next best solution is defined as the highest scoring state with a minimum degree of "dissimilarity" w.r.t. previously chosen solutions, where dissimilarity is measured under a function $\Delta(\cdot,\cdot)$:

$$\mathbf{y}^M = \underset{\mathbf{y}\in\mathcal{Y}}{\mathrm{argmax}} \quad S(\mathbf{y};\mathbf{x}) \tag{9a}$$

$$s.t. \qquad \Delta(\mathbf{y},\mathbf{y}^m) \geq k_m \quad \forall m \in [M-1]. \tag{9b}$$

In general, this problem is NP-hard and Batra et al. [4] proposed to use the Lagrangian relaxation formed by dualizing the dissimilarity constraints $\Delta(\mathbf{y},\mathbf{y}^m) \geq k_m$:

$$g(\boldsymbol{\lambda}) = \max_{\mathbf{y}\in\mathcal{Y}} \; S_\Delta(\mathbf{y};\mathbf{x}) \doteq S(\mathbf{y};\mathbf{x}) + \sum_{m=1}^{M-1} \lambda_m \left(\Delta(\mathbf{y},\mathbf{y}^m) - k_m\right). \tag{10}$$

Here $\boldsymbol{\lambda} = \{\lambda_m \,|\, m \in [M-1]\}$ is the set of Lagrange multipliers, which determine the weight of the penalty imposed for violating the diversity constraints.

Following [4], we use Hamming diversity, i.e., $\Delta(\mathbf{y},\mathbf{y}^m) = \sum_{u\in\mathcal{V}} [\![y_u \neq y_u^m]\!]$, where $[\![\cdot]\!]$ is 1 if the input condition is true, and 0 otherwise. This function counts the number of nodes that are labeled differently between two solutions. For Hamming dissimilarity, the Δ-augmented scoring function (10) can be written as:

$$S_\Delta(\mathbf{y};\mathbf{x}) = S(\mathbf{y};\mathbf{x}) + \underbrace{\sum_{u\in\mathcal{V}} \left(\sum_{m=1}^{M-1} \lambda_m [\![y_u \neq y_u^m]\!] \right)}_{\text{Augmented Unary Score}}. \tag{11}$$

The second term in the above equation corresponds to augmenting the unary potentials, $\theta_u(y_u)$, that was mentioned above in Section 2.1. Thus, the maximization in Eq. (10) can be performed simply by feeding a perturbed unary term to the algorithm used for maximizing the score (e.g., α-expansion or TRW-S). The dissimilarity constraints, k_m, does not appear in the above equation because it is now replaced by soft diversity terms.

Perturb-and-MAP. Perturb-and-MAP [20] is a framework for generating IID random samples from a Markov Random Field (MRF), which exploits the advancements in energy minimization techniques by avoiding costly MCMC.

Given a scoring function $S(\mathbf{y};\mathbf{x})$, the Perturb-and-MAP model proposes to generate random samples by perturbing the terms constituting the scoring function and finding the configuration that optimizes the perturbed random field. For the Perturb-and-MAP model to coincide with the Gibbs model, the perturbation needs to be performed on the fully expanded unary and binary potentials by adding IID Gumbel noise to them. However, as suggested by the authors of [20], we add IID Gumbel noise only to the unary terms in our experiments. The Perturb-and-MAP solutions can be obtained by optimizing the following configuration

$$\tilde{S}(\mathbf{y};\mathbf{x}) = S(\mathbf{y};\mathbf{x}) + \lambda\epsilon_u(y_u). \tag{12}$$

The term, $\epsilon_u(y_u)$, denotes the IID Gumbel noise,

$$\epsilon_u(y_u) = -\log\left(-\log(s)\right), \tag{13}$$

where, s is a sample from the uniform distribution, and λ is used to denote the perturbation strength (we are overloading the symbol λ to denote both the strength of the Gumbel noise, when used in conjunction with Perturb-and-MAP, and to denote the amount of diversity, when used in conjunction with DivMBest).

Relation to DivMBest. We take a short digression to compare and contrast DivMBest and Perturb-and-MAP. Compare Equations (10) and (12). Notice that the additional terms in DivMBest may also be viewed as a perturbation. However, while Perturb-and-MAP perturbations are IID Gumbel, the DivMBest perturbations depend on the solutions that have been found so far. In other words, in DivMBest, the perturbation is performed only on those parts of the potential table, that correspond to a labelling produced in one of the previous solutions.

3.3 Learning Parameters in EMBR

We assume that the weights $\mathbf{w}$ parameterizing the score function $S(\mathbf{y};\mathbf{x},\mathbf{w})$ are provided as input to our approach, presumably learnt on some training dataset. We follow the recommendation of [4], and use a single λ parameter (so $\lambda_m = \lambda$ for all m). There are three parameters to be tuned in EMBR – λ, T, and M – which are chosen by grid search to maximize task-loss $\ell(\cdot,\cdot)$ on some validation dataset.

We report results with four variants of our approach (and one sensitivity test), corresponding to tuning 1/2/3 parameters:

- *One parameter EMBR*
  - EMBR-($\lambda_M, T=\infty, M$): We set T to ∞ (which corresponds to a uniform distribution over the solutions) and M to a value where the `oracle` curve (accuracy of the most accurate solution in the set) starts to plateau; for the binary segmentation and pose estimation experiments, we set M to 50, and for the PASCAL VOC segmentation experiments, we set M to 30. Thus, λ_M is the only parameter that is optimized via grid-search to maximize EMBR performance at M. We show plots of this variant as a function of the numbers of solutions available at test-time, from 1 to M, however the final result is simply a single number (at $M=50$ or $M=30$) (i.e., we do not optimize test performance over M).
- *Two parameter EMBR*
  - EMBR-(λ_M, T_M, M): We set M to a value where the `oracle` starts to plateau (as above), and both

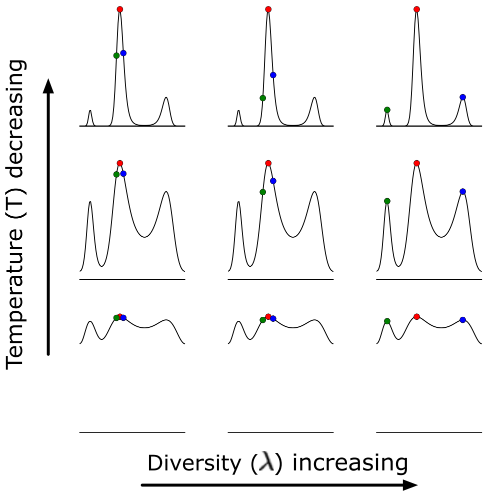


Fig. 2. Illustration of the effect of two of the three parameters in our model. The colored dots represent the chosen candidates $\mathbf{Y_M}$, and the height of the curve at these points illustrates the $\tilde{\mathrm{P}}(\mathbf{y})$ value assigned to each candidate. These parameters are learned via grid search, then predictions are made by using these $\tilde{\mathrm{P}}(\mathbf{y})$ values within a Minimum Bayes Risk predictor.

λ_M and T_M are tuned to maximize EMBR performance at M. Fig. 2 illustrates the effect of varying T and λ on our model.

- EMBR-($\lambda_{m^*}, T = \infty, m^*$): In this case, we set T to ∞ and tune both M and λ.

- *Three parameter EMBR*
  - EMBR-($\lambda_{m^*}, T_{m^*}, m^*$): We tune all three parameters, λ, T and M.
- *Sensitivity analysis*
  - EMBR-(λ_m, T_m): For each $m \in [M]$, we identify the best parameters λ_m and T_m. During test time, we use the appropriate pair of parameters. This curve is reported to show the sensitivity of the method to the choice of M. It is not valid to take the maximum of this curve over test performance, as that would be choosing M to maximize test performance.

4 Experiments

Setup. We tested EMBR on three different problems:

1) Binary (foreground-background) interactive segmentation (Section 4.1), on 100 images from the PASCAL VOC 2010 `val` set.
2) 2D articulated human body pose estimation on,
   a) the PARSE dataset [33] (Section 4.2), and
   b) the Leeds Sports Pose (LSP) [7] (Section 4.2.2), and,
3) Category-level segmentation on PASCAL VOC 2012 Segmentation Challenge dataset [9] (Section 4.3).

These scenarios use very different models & score functions, with different MAP inference algorithms (max-flow/min-cut, dynamic programming, greedy inference), and different high-order task-losses $\ell(\cdot,\cdot)$ (intersection-over-union, APK [33], PASCAL metric [10]). Despite the differences, our approach is uniformly applicable. In two of the models, binary segmentation with supermodular potentials and pose estimation with a tree-structured model, we can compute MAP exactly. Thus, when EMBR outperforms MAP, the cause was not the approximate maximization for MAP. In all three cases, the loss functions in the problem is a high-order loss [28], making exact MBR intractable.

Baselines:

- On all experiments, we compare our approach against the natural baseline of MAP, which simply predicts the highest scoring solution and is indifferent to the setting of T, λ and M.
- In a manner similar to [4], we also report `oracle` accuracies, i.e., the accuracy of the *most accurate* solution in the set $\mathbf{Y_M}$. This forms an upper-bound on the performance of any predictor (including MAP and EMBR) which picks a single solution from the set $\mathbf{Y_M}$.
- In the pose estimation experiments on the PARSE dataset (Section 4.2.1), we also compare against the results of Yang and Ramanan [33], which was the previous state-of-the-art on the PARSE dataset.
- In the pose estimation experiments on the LSP dataset (Section 4.2.2), we also compare against the results of Chen and Yuille [7], which was the previous state-of-the-art on the LSP dataset.

Main theme in results. Due to the greedy nature of DivMBest, EMBR degenerates to MAP at $M = 1$. Our results will show that EMBR consistently and convincingly outperforms the natural baseline of MAP in all experiments. This supports our claim that incorporating the key ideas from decision theory – incorporating task-specific losses and averaging over uncertainty – leads to significant improvements. In the pose estimation experiments, we outperform the state-of-art method of Yang and Ramanan [33] by about 8 percent. It is important to remember that this is all without access to any new features or model – simply by utilizing information about the task loss!

4.1 Binary Segmentation

Model. We replicate the binary segmentation setup from [4], who simulated an interactive segmentation scenario on 100 images from the PASCAL VOC 2010 dataset, and manually provided scribbles on objects contained in them. For each image, a 2-label pairwise CRF on superpixels is set up. At each superpixel, Transductive SVMs are trained on color and texture features, and their outputs are used as node potentials. The edge potentials are contrast-sensitive Potts. This results in a supermodular score function so we can efficiently compute the exact MAP and DivMBest solutions using graph-cuts. 50 images were used for training base model (Transductive SVMs), 25 for learning the EMBR-parameters, and 25 for reporting testing accuracies. The task loss $\ell(\cdot,\cdot)$ and the EMBR loss $\tilde{\ell}(\cdot,\cdot)$ are 1 minus the intersection-over-union of the ground-truth and predicted foreground masks. The images, precomputed features and groundtruth for the 50 images can be downloaded from the project webpage [1].

Results. Fig. 4a shows the performance of EMBR as a function of M while using DivMBest samples. We can see that EMBR-($\lambda_M, T_M, M = 50$) outperforms MAP by about 7 percent (68.87 percent). The 1-Param and 3-Param settings of EMBR perform similar, suggesting that default choices T and M work well.

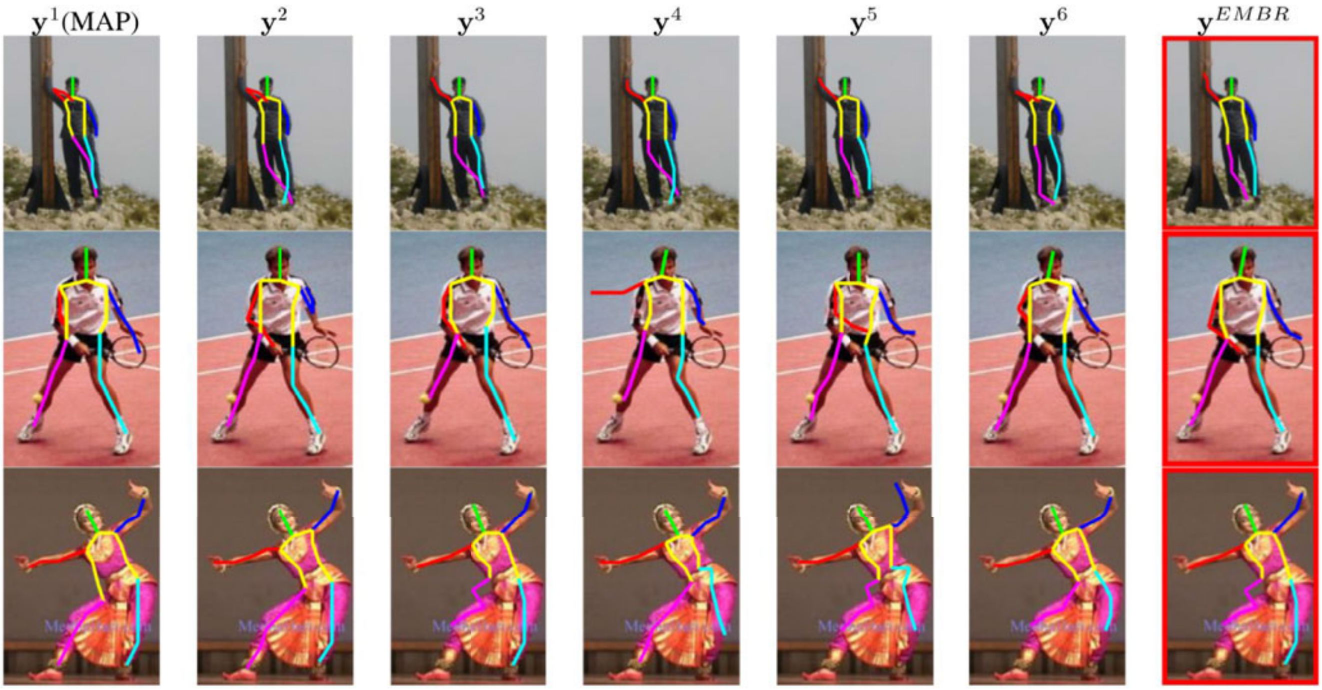


Fig. 3. Qualitative Results: Within each row, the first column corresponds to MAP, the middle columns show the diverse solutions, and the last column shows the EMBR prediction. The top two rows show examples where EMBR selects a better pose than MAP, while the bottom row shows an example where MAP produces a better result. Notice the right hand, and the separation of the legs in the EMBR solution in the first row. In the second row, notice the right hand being correctly detected by EMBR.

Fig. 5a shows the performance of EMBR while using Perturb-and-MAP samples. The improvement that EMBR-PnM provides over MAP is not that significant.

4.2 Pose Estimation

4.2.1 PARSE Dataset

Model. We replicate the setup of Yang and Ramanan [33], whose mixture-of-parts deformable human-body model has demonstrated competitive performance on various benchmarks. The variables in their model are part (head, body, etc.) locations and type. The graph-structure is a tree and (exact) inference is performed by dynamic programming. The loss function most commonly used for this problem is the Percentage of Correct Parts (PCP) [8]. Yang and Ramanan proposed a novel metric for measuring performance called Average Precision of Keypoints [33], which treats each keypoint as a separate detection problem, and measures the average precision in the precision-recall curve for each keypoint. In our experiments, we use PCP as the instance-level loss function used in the EMBR definition $\tilde{\ell}(\cdot,\cdot)$ but choose the best parameters of EMBR by optimizing meanAPK, which is a corpus-level metric. The parameters are chosen via cross-validation on the PARSE test set.

Results. Fig. 4b shows the meanAPK achieved by various methods versus M. We can see that all EMBR variants significantly outperform MAP. Note that in segmentation, the evaluation metric only cares about the predictions, not the scores associated with the predictions. In pose estimation, the precision-recall curve for each keypoint is different based on the score associated with that particular full-body detection. This is why oracle at $M = 1$ (which uses the original scores of [33]) does not perform identical to EMBR at $M = 1$ (which uses the Bayes Gain as the score).

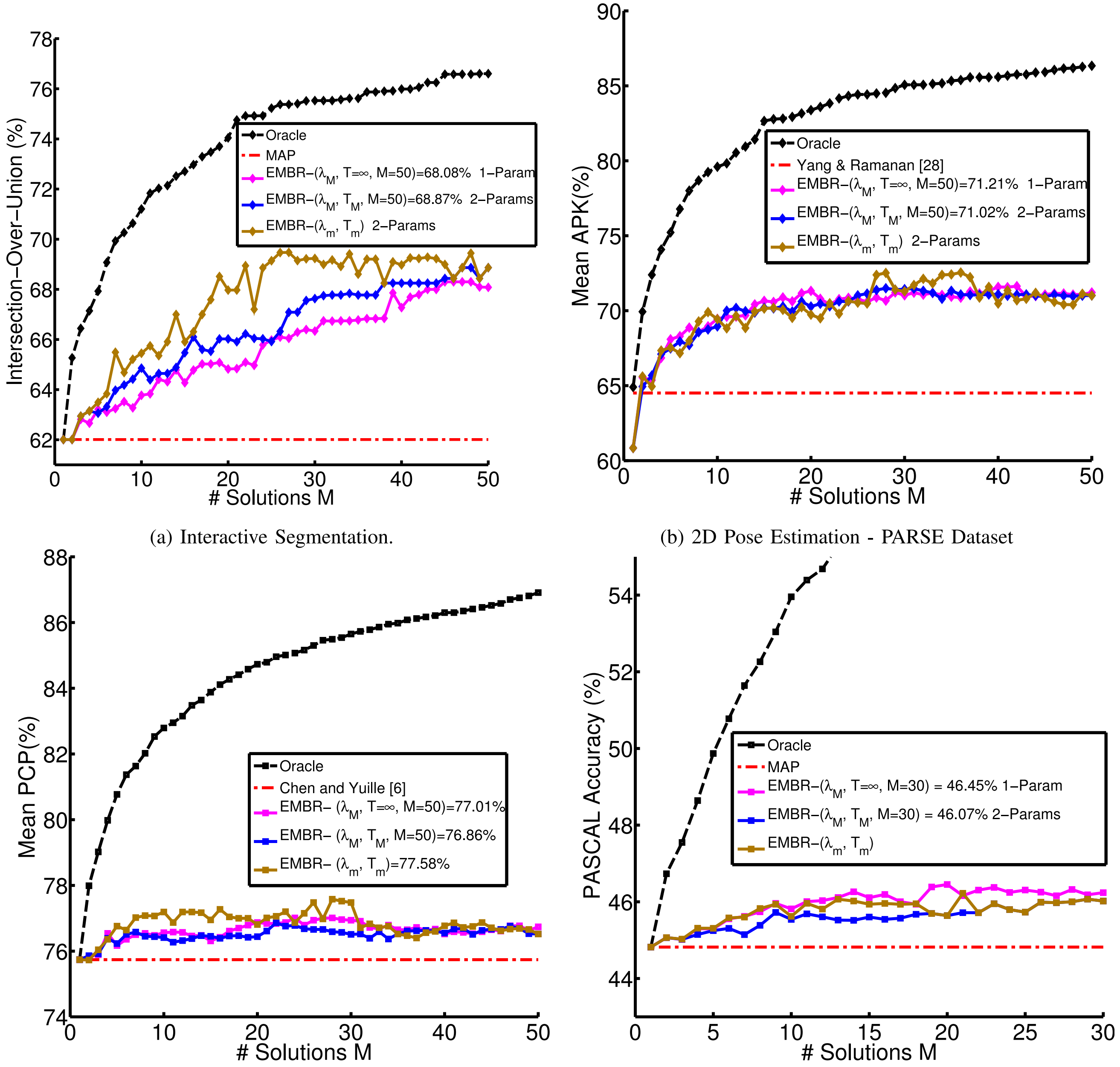
EMBR-($\lambda_M, T_M, M = 50$) achieves a final mean-APK of 71.02 percent, and EMBR-($\lambda_{m*}, T_{m*}, m*$) performs slightly better by achieving 71.53 percent, which is about 7-percentage-point improvement over the previous state of the art of 64.5 percent [33]. Fig. 3 shows some qualitative results.

Fig. 5b shows the performance of EMBR when using the Perturb-and-MAP samples. EMBR-PnM provides a non-trivial improvement over MAP. However, the improvement is not as significant as the improvement produced while using DivMBest samples.

4.2.2 Leeds Sports Pose (LSP) Dataset

Model. We replicate the set up of Chen and Yuille [7], who show state-of-the-art results on the Leeds Sports Pose Dataset. The model is similar to the model used by Yang and Ramanan [33] except for two key differences.

1) *Deep scores:* The authors of [7] train a deep network on small image patches to perform a multi-way classification among one of the many joint configurations in a human body. The evidence obtained from the top layer of the deep network can be used in the unary and pairwise terms of their graphical model.
2) *Image dependent pairwise relational (IDPR) term:* The pairwise terms of the score function, in addition to having the standard deformation term, also has an Image Dependent Pairwise Relational (IDPR) term. The intuition behind having the IDPR term is that one can reliably predict the relative positions of a part's neighbors by *only* observing the local image patch around it.



(a) Interactive Segmentation.

(b) 2D Pose Estimation - PARSE Dataset

(c) 2D Pose Estimation - LSP Dataset

(d) PASCAL VOC `val`.

Fig. 4. Quantitative results using DivMBest: We show the performance of different methods versus M on three different problems. We observe that EMBR consistently and convincingly outperforms the natural baseline of MAP, and in the case of pose estimation, achieves state-of-art results.

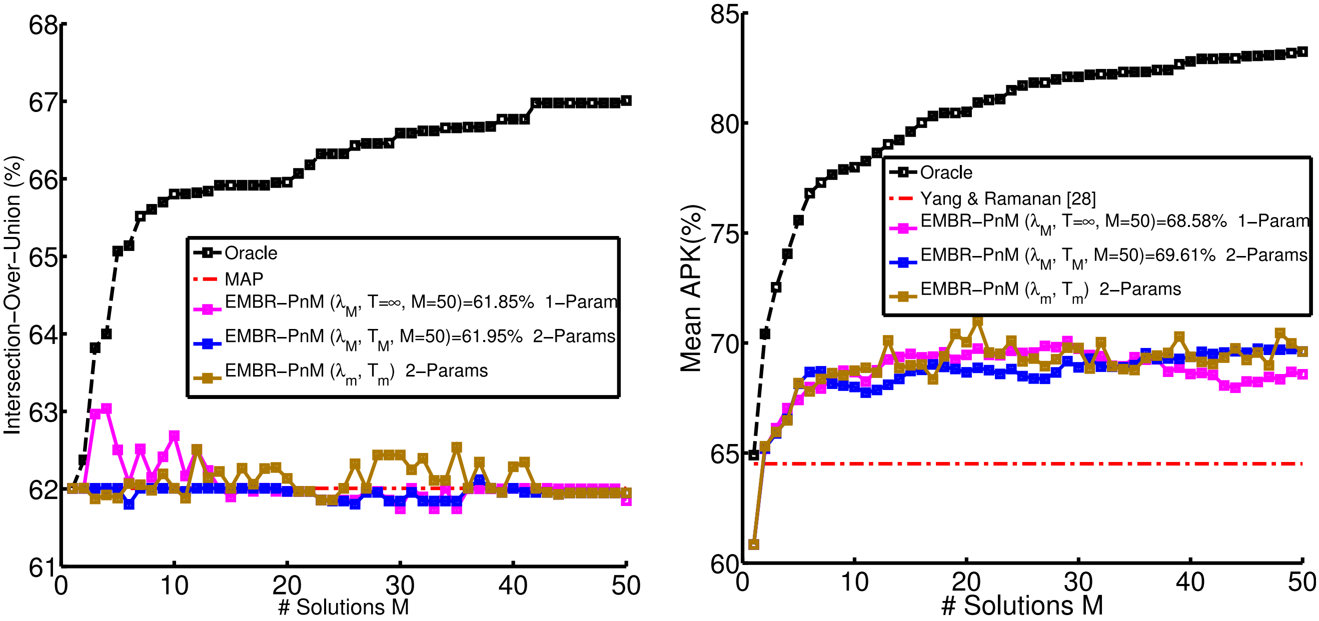
The graph structure is a tree and (exact) inference can be performed using dynamic programming. We use PCP as the EMBR loss function. The best EMBR parameters are chosen by optimizing the mean PCP, which is a corpus-level metric. The parameters are chosen via cross-validation. Since the authors of [7] use PCP to report their results, we also report our results using PCP enabling easy comparison.

Results. The LSP Dataset is a difficult pose estimation dataset since it contains highly articulated human poses that occur in sports scenes. We can see from Fig. 4c, which is a plot of the mean PCP versus M, that EMBR achieves a modest 1.27 percent improvement over the state-of-the-art result on this difficult dataset. The MAP accuracy on this dataset is 75.74 percent and EMBR-$(\lambda_M, T_M, M = 50)$ achieves a final mean-PCP of 77.01 percent. We obtained slightly better MAP accuracy (mentioned above) than the results quoted in the original paper [7] when we retrained their model.

4.3 Category Segmentation on VOC12

Finally, we study the performance of EMBR on category-level segmentation on the PASCAL VOC 2012 dataset, where the goal is to label every pixel with one of 20 object categories or the background.

Model. We build on the CPMC+O2P framework of Carreira et al. [5] – approximately 150 CPMC segments [6] are generated for each image, scored via Support Vector Regressors over second-order pooled features [5], and then greedily pasted. The sum of the scores of the pasted segments is the score of a segmentation, and DivMBest is used



(a) Interactive Segmentation.

(b) 2D Pose Estimation - PARSE Dataset.

Fig. 5. Quantitative results using Perturb-and-MAP: We show the performance of different method versus M on the interactive binary segmentation and the pose estimation problems. We observe that EMBR-PnM does improve upon the natural baseline of MAP (though the improvement is not as much as in the DivMBest case.).

to produce diverse segmentation maps. The task accuracy $(1 - \ell(\cdot,\cdot))$ in this case is the corpus-level Jaccard Index used by PASCAL, averaged over all 21 categories. The EMBR loss $\tilde{\ell}(\cdot,\cdot)$ is 1 minus the instance-level approximation to this corpus-level loss (a discussion about corpus-level loss is presented in Section 5.1).

Results. Fig. 4d shows the PASCAL accuracy as a function of M. This is a difficult problem; EMBR-$(\lambda_{m^*}, T_{m^*}, m^*)$ yields an improvement of only about 1 percent (45.82 percent). The 1-Param and 3-Param settings of EMBR perform similar.

4.4 Robust Loss Functions

The EMBR of a particular sample, $\mathbf{y}^i$, as can be seen from Equation (5), is affected by the loss with respect to every other sample, $\mathbf{y}^j \in \mathbf{Y}_M$, $\forall j \neq i$. Therefore, the Bayes Risk, $BR(\mathbf{y}^i)$, can be adversely affected by outliers. In order to cull the potential unfavorable effects that the outliers might have on the computation of the Bayes Risk, we make use of a robust version of the loss function, defined as,

$$\tilde{\ell}_r(\mathbf{y}^i, \mathbf{y}^j) = \begin{cases} \tilde{\ell}(\mathbf{y}^i, \mathbf{y}^j), & \text{if } \tilde{\ell}(\mathbf{y}^i, \mathbf{y}^j) < \alpha \\ \alpha, & \text{otherwise.} \end{cases} \tag{14}$$

We experimented with the above loss and found it to improve the performance of the EMBR predictor.

- *Binary segmentation:* We get an improvement of 1.7 percent over the 68.87 percent and end up with a final IOU score of 70.57 percent, when using an α of 0.7.
- *Pose estimation - PARSE dataset:* Robust EMBR loss improved the performance of the EMBR predictor by a further 0.8 percent and we end up with a final accuracy of 72.32 percent, when using an α of 0.6.
- *Pose estimation - LSP dataset:* We noticed only minor improvements on this dataset.
- *Category segmentation on VOC12:* We obtained an improvement of 0.41 percent to achieve a final accuracy of 46.23 percent on this experiment, when using an α of 0.5.

All α values reported above were found by performing a grid search.

5 Discussions

5.1 Instance-Level versus Corpus-Level Losses

In a number of settings, for instance PASCAL segmentation, the evaluation criteria is a corpus-level metric, which measures the loss of predictions for an entire dataset, and not a single instance. In its current format, EMBR utilizes only instance-level losses (in the PASCAL experiments, an instance-level approximation to PASCAL loss). However, we can, and do, optimize the EMBR parameters over the corpus-level loss as,

$$(\lambda^*, T^*, m^*) = \underset{\lambda,T,m}{\operatorname{argmin}}\, \ell_D(\{\mathbf{y}_i^{EMBR}\}_1^n, \{\mathbf{y}_{gt}^i\}_1^n). \tag{15}$$

Here, $\ell_D(.,.)$ is the corpus-level loss, $\mathbf{y}_i^{EMBR}$ is the EMBR prediction for the ith image, $\mathbf{y}_{gt}^i$ is the ground-truth corresponding to the ith image, and $\{.\}_1^n$ denotes a set containing n elements.

We also experimented with training the EMBR parameters over the mean instance-level loss and report the performance of the predictor on the corpus-level loss. Table 1 tabulates the results from the various combinations that are possible, i.e., training on either the instance-level loss or the corpus-level loss and testing on the same or the other. Thus, we get four train-test combinations.

The main theme that we found is that training using a corpus-level loss provided a better set of parameters that improved both the corpus-level loss and the instance-level

TABLE 1
This Table Shows the Performance of the EMBR Predictor When the Three Parameters Are Tuned on Both the Corpus-Level Loss and the Instance-Level Loss

Val Train	Corpus (MAP = 44.8%)	Instance (MAP = 57.05%)
Corpus	45.82%	57.23%
Instance	44.81%	57.05%

We can see that tuning on the corpus-level loss always improves the performance of the predictor.

loss on the held-out set, suggesting that corpus-level losses may be intrinsically better for training.

In future, we plan to extend the EMBR predictor itself to naturally handle corpus-level losses.

5.2 Relation to MAP Predictor

The EMBR predictor has interesting relations to the MAP predictor. Under certain conditions, the EMBR predictor degenerates to the standard MAP predictor.

The following cases depend on the type of solution generator. They hold true when $\mathbf{y}^1 = \mathbf{y}^{MAP}$, which is the case when using DivMBest.

- *When $M = 1$:* It is easy to see that in the trivial case when we have just one solution, the EMBR predictor will predict it to be the "best". Since $\mathbf{y}^1 \in \mathbf{Y_M}$ corresponds to the output predicted by the MAP predictor, the EMBR predictor and the MAP predictor agree for the case when $M = 1$.
- *When $M = 2$ and DivMBest solutions are exact:* Consider the case when $M = 2$. The Bayes Risk for the two solutions, $\mathbf{y}^1$ and $\mathbf{y}^2$ can be written as,

$$\begin{aligned} BR(\mathbf{y}^1) &= \tilde{\mathrm{P}}(\mathbf{y}^1|\mathbf{x})\ell(\mathbf{y}^1,\mathbf{y}^1) + \tilde{\mathrm{P}}(\mathbf{y}^2|\mathbf{x})\ell(\mathbf{y}^1,\mathbf{y}^2) \\ &= \tilde{\mathrm{P}}(\mathbf{y}^2|\mathbf{x})\ell(\mathbf{y}^1,\mathbf{y}^2) \end{aligned} \tag{16}$$

  and

$$\begin{aligned} BR(\mathbf{y}^2) &= \tilde{\mathrm{P}}(\mathbf{y}^1|\mathbf{x})\ell(\mathbf{y}^2,\mathbf{y}^1) + \tilde{\mathrm{P}}(\mathbf{y}^2|\mathbf{x})\ell(\mathbf{y}^2,\mathbf{y}^2) \\ &= \tilde{\mathrm{P}}(\mathbf{y}^1|\mathbf{x})\ell(\mathbf{y}^2,\mathbf{y}^1), \end{aligned} \tag{17}$$

  respectively. Since the loss is symmetric, $\mathbf{y}^1$ (i.e., the MAP solution) will *always* be chosen if $\tilde{\mathrm{P}}(\mathbf{y}^2|\mathbf{x}) < \tilde{\mathrm{P}}(\mathbf{y}^1|\mathbf{x})$, which is true when the DivMBest candidates are obtained using exact optimization of the energy function.
- *In the limit when $T \to 0$:* For any $M \geq 1$, when the temperature parameter, T, tends to zero, the EMBR predictor will always predict the MAP solution. From Equation (8), we can see that, when $T \to 0$, the conditional distribution, $\tilde{\mathrm{P}}(\mathbf{y}|\mathbf{x})$, tends to a delta function at $\mathbf{y}^1$,

$$\tilde{\mathrm{P}}(\mathbf{y}^1|\mathbf{x}) = \frac{\exp\frac{1}{T}S(\mathbf{y}^1;\mathbf{x})}{\sum_{\mathbf{y}'\in\mathbf{Y_M}}\exp\frac{1}{T}S(\mathbf{y}';\mathbf{x})} \to 1, \tag{18}$$

  and

$$\tilde{\mathrm{P}}(\mathbf{y}^i|\mathbf{x}) = \frac{\exp\frac{1}{T}S(\mathbf{y}^i;\mathbf{x})}{\sum_{\mathbf{y}'\in\mathbf{Y_M}}\exp\frac{1}{T}S(\mathbf{y}';\mathbf{x})} \to 0 \;\forall\; i \neq 1. \tag{19}$$

  This is true while using proposals where $\mathbf{y}^1 = \mathbf{y}^{MAP}$, causing $S(\mathbf{y}^1;\mathbf{x}) > S(\mathbf{y}^i;\mathbf{x}) \forall i \neq 1$. Therefore, when $T \to 0$, the EMBR predictor ends up picking the MAP solution.
- *When using a 0/1 loss function:* The most interesting case occurs when we use a $0/1$ loss function, $\ell(\cdot,\cdot)$, within the EMBR predictor. The $0/1$ loss function is defined as

$$\ell(\mathbf{y}^i,\mathbf{y}^j) = \begin{cases} 0, & \text{if } i = j \\ 1, & \text{otherwise.} \end{cases} \tag{20}$$

The loss matrix, as given in Equation (6), is a matrix with the diagonal elements being zeros and the off-diagonal elements being ones

$$L = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{bmatrix}. \tag{21}$$

We now show that the EMBR predictor,

$$\mathbf{y}^{EMBR} = \underset{\mathbf{y}^i, i\in[M]}{\operatorname{argmin}} \; L\mathbf{p}, \tag{22}$$

which can be expanded as

$$\mathbf{y}^{EMBR} = \underset{\mathbf{y}^i, i\in[M]}{\operatorname{argmin}} \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathrm{P}}(\mathbf{y}^1|\mathbf{x}) \\ \tilde{\mathrm{P}}(\mathbf{y}^2|\mathbf{x}) \\ \vdots \\ \tilde{\mathrm{P}}(\mathbf{y}^3|\mathbf{x}) \end{bmatrix}, \tag{23}$$

and simplified to

$$\mathbf{y}^{EMBR} = \underset{\mathbf{y}^i, i\in[M]}{\operatorname{argmin}} \begin{bmatrix} \sum_{i\in[M]\setminus\{1\}} \tilde{\mathrm{P}}(\mathbf{y}^i|\mathbf{x}) \\ \sum_{i\in[M]\setminus\{2\}} \tilde{\mathrm{P}}(\mathbf{y}^i|\mathbf{x}) \\ \vdots \\ \sum_{i\in[M]\setminus\{M\}} \tilde{\mathrm{P}}(\mathbf{y}^i|\mathbf{x}) \end{bmatrix}, \tag{24}$$

when using a $0/1$ loss function, will end up selecting the MAP solution. Using the property that $\sum_{i\in[M]} \tilde{\mathrm{P}}(\mathbf{y}^i|\mathbf{x}) = 1$, we can simplify the above optimization function to

$$\mathbf{y}^{EMBR} = \underset{\mathbf{y}^i, i\in[M]}{\operatorname{argmin}} \begin{bmatrix} 1 - \tilde{\mathrm{P}}(\mathbf{y}^1|\mathbf{x}) \\ 1 - \tilde{\mathrm{P}}(\mathbf{y}^2|\mathbf{x}) \\ \vdots \\ 1 - \tilde{\mathrm{P}}(\mathbf{y}^M|\mathbf{x}) \end{bmatrix}. \tag{25}$$

Denoting the vector of ones as $\mathbf{1}$ and the probability vector as $\mathbf{p}$, the above equation can be succinctly written as,

$$\mathbf{y}^{EMBR} = \underset{\mathbf{y}^i, i\in[M]}{\operatorname{argmin}} \; \mathbf{1} - \mathbf{p}. \tag{26}$$

The above optimization function is equivalent to the following optimization,

$$\mathbf{y}^{EMBR} = \underset{\mathbf{y}^i, i\in[M]}{\text{argmax}}\ \mathbf{p}. \tag{27}$$

When

$$\mathbf{y}^1 > \mathbf{y}^i\ \forall\ i\ \neq 1, \tag{28}$$

which is true for DivMBest, $\mathbf{y}^{EMBR}$ ends up being the same as $\mathbf{y}^1$, which is nothing but $\mathbf{y}^{MAP}$, i.e.,

$$\mathbf{y}^{EMBR} = \mathbf{y}^1 = \mathbf{y}^{MAP}. \tag{29}$$

Thus, we have shown that under the special case where the EMBR loss is a 0/1 loss function, the EMBR predictor will end up picking the same solution as the MAP predictor.

5.3 When Is EMBR Expected to Work?

Intuitively speaking, one major requirements of EMBR is that the loss function provide some additional information about the solutions. Therefore, loss functions such as the 0/1 loss function do not help our case (see the final point of Section 5.2). We believe that higher-order loss functions that help quantify semantic differences between the discrete set of solutions will significantly help the EMBR predictor. Moreover, the performance of the EMBR predictor also depends on the quality of the solutions. For the predictor to work well, the solutions should have some shared parts among them so that the loss function can extract additional information from the pairwise differences. Characterizing theoretical requirements for EMBR is a direction for future work.

6 Conclusions

We have described a simple meta-algorithm for making predictions in structured output models that are better suited for a particular task-specific evaluation measure. The primary benefit of the formulation is in its simplicity, efficiency, and strong performance. We believe that the two-stage framework that we operate under is particularly desirable, because it allows researchers to continue to use the techniques that are popular in building models in the first stage without worrying about whether the method will be compatible with a variety of loss functions of interest in the second stage.

Moreover, we have shown the effectiveness of this approach by testing it on multiple computer vision tasks and on various datasets. To facilitate easy reproduction of the results presented in this paper, we have released the source code for all our experiments.[2]

We hope that this work will encourage researchers to define and optimize more complicated evaluation measures, which more accurately reflect the tasks that our vision systems need to accomplish to be useful in the real world.

Acknowledgments

The authors thank Varun Ramakrishna for helpful discussions during early stages of this work. They thank Xianjie Chen for help with using his pose estimation code. VP and AY were partially supported by the National Science Foundation grant CCF-1317376. DB was partially supported by the National Science Foundation under grants IIS-1353694 and IIS-1350553, the Army Research Office YIP Award W911NF-14-1-0180, and the Office of Naval Research grant N00014-14-1-0679. 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 US Government or any sponsor. This work was done while V. Premachandran and A. Yuille were at the University of California, Los Angeles.

References

[1] (2015). Embr project page [Online]. Available: https://filebox.ece.vt.edu/ vittal/embr/index.html

[2] A. Barbu and S.-C. Zhu, "Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1239–1253, Aug. 2005.

[3] D. Batra, "An efficient message-passing algorithm for the m-best MAP problem," in *Proc. 28th Conf. Uncertainty Artif. Intell.*, 2012, pp. 121–130.

[4] D. Batra, P. Yadollahpour, A. Guzman-Rivera, and G. Shakhnarovich, "Diverse M-best solutions in Markov random fields," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 1–16.

[5] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, "Semantic segmentation with second-order pooling," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 430–443.

[6] J. Carreira and C. Sminchisescu, "Constrained parametric mincuts for automatic object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3241–3248.

[7] X. Chen and A. Yuille, "Articulated pose estimation by a graphical model with image dependent pairwise relations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1736–1744.

[8] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari, "Articulated human pose estimation and search in (almost) unconstrained still images," Technical report, ETHZ, 2010.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. (2012). The pascal visual object classes challenge 2012 (VOC2012) Results [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[11] A. Guzman-Rivera, D. Batra, and P. Kohli, "Multiple choice learning: Learning to produce multiple structured outputs," In *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1808–1816.

[12] A. Guzman-Rivera, P. Kohli, D. Batra, and R. Rutenbar, "Efficiently enforcing diversity in multi-output structured prediction," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2014, pp. 284–292.

[13] P. Hammer, "Some network flow problems solved with pseudo-Boolean programming," *Oper. Res.*, vol. 13, pp. 388–399, 1965.

[14] F. Huszár and D. Duvenaud, "Optimally-weighted herding is Bayesian quadrature," *arXiv Preprint arXiv:1204.1664*, 2012.

[15] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.

[16] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, Mar. 1957.

[17] I. Murray, Z. Ghahramani, and D. MacKay, "Mcmc for doubly-intractable distributions," *arXiv Preprint arXiv:1206.6848*, 2012.

[18] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 625–632.

[2] The source code can be downloaded from https://github.com/VittalP/embr.

[19] D. Nilsson, "An efficient algorithm for finding the M most probable configurations in probabilistic expert systems," *Statist. Comput.*, vol. 8, pp. 159–173, 1998. 10.1023/A:1008990218483.

[20] G. Papandreou and A. Yuille, "Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 193–200.

[21] D. Park and D. Ramanan, "N-best maximal decoders for part models," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2627–2634.

[22] J. D. Park and A. Darwiche, "Solving map exactly using systematic search," in *Proc. 19th Conf. Uncertainty Artif. Intell.*, 2002, pp. 459–468.

[23] J. Pearl, "Reverend Bayes on inference engines: A distributed hierarchical approach," in *Proc. Am. Assoc. Artif. Intell. Nat. Conf.*, 1982, pp. 133–136.

[24] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Proc. Adv. Large Margin Classifiers*, 1999, pp. 61–74.

[25] J. Porway and S.-C. Zhu, "C^4: Exploring multiple solutions in graphical models by cluster sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1713–1727, Sep. 2011.

[26] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 1043–1050.

[27] S. E. Shimony, "Finding MAPs for belief networks is NP-hard," *Artif. Intell.*, vol. 68, no. 2, pp. 399–410, Aug. 1994.

[28] D. Tarlow and R. S. Zemel, "Structured output learning with high order loss functions," in *Proc. 15th Int. Conf. Artif. Intell. Statist.*, 2012, pp. 1212–1220.

[29] M. Taylor, J. Guiver, S. Robertson, and T. Minka, "Softrank: Optimizing non-smooth rank metrics," in *Proc. Int. Conf. Web Search Data Mining*, 2008, pp. 77–86.

[30] Z. Tu and S.-C. Zhu, "Image segmentation by data-driven Markov chain Monte Carlo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 657–673, May 2002.

[31] L. Valiant, "The complexity of computing the permanent," *Theoretical Comput. Sci.*, vol. 8, no. 2, pp. 189–201, 1979.

[32] M. Welling, "Herding dynamical weights to learn," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 1121–1128.

[33] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2878–2890, Dec. 2013.

[34] C. Yanover and Y. Weiss, "Finding the M most probable configurations using loopy belief propagation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 289–296.

[35] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 609–616.

[36] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2002, pp. 694–699.

Vittal Premachandran received the PhD degree in computer science from Nanyang Technological University, Singapore. He is currently a postdoctoral researcher at Johns Hopkins University. He was previously a postdoctoral researcher at the University of California, Los Angeles. He was also a visiting researcher at the National University of Singapore. He serves as a reviewer for main conferences and journals. His work has received the Best Student Paper award at the International Conference on Image Processing 2013. His research interests include computer vision, machine learning, and artificial intelligence.

Daniel Tarlow received the PhD degree in computer science from the machine learning group at the University of Toronto in 2013 and is currently a researcher in the Machine Intelligence and Perception group at Microsoft Research in Cambridge, United Kingdom. His research interests are in the application of probabilistic models to problems involving complex structured objects and decision criteria. He serves as a reviewer for the main conferences and journals in machine learning, and he was a coorganizer of the NIPS 2012, 2013, and 2014 Workshops on perturbations, optimization, and statistics. He has received paper awards at Uncertainty in Artificial Intelligence 2011 (Best Student Paper, Runner Up) and Advances in Neural Information Processing Systems 2014 (Outstanding Paper), and holds a research fellowship at Darwin College, University of Cambridge.

Alan L. Yuille received the BA degree in mathematics from the University of Cambridge in 1976. His PhD on theoretical physics, supervised by Prof. S.W. Hawking, was approved in 1981. He was a research scientist in the Artificial Intelligence Laboratory at MIT and the Division of Applied Sciences at Harvard University from 1982 to 1988. He served as an assistant and associate professor at Harvard until 1996. He was a senior research scientist at the Smith-Kettlewell Eye Research Institute from 1996 to 2002. He joined the University of California, Los Angeles, as a full professor with a joint appointment in statistics and psychology in 2002. In 2016, he joined Johns Hopkins University as a Bloomberg distinguished professor in cognitive science and computer science. His research interests include computational models of vision, mathematical models of cognition, and artificial intelligence and neural networks.

Dhruv Batra is an assistant professor at the Bradley Department of Electrical and Computer Engineering at Virginia Tech, where he leads the VT Machine Learning & Perception group. His research interests lie at the intersection of machine learning, computer vision, and AI, with a focus on developing intelligent systems that are able to concisely summarize their beliefs about the world, integrate information and beliefs across different sub-components or 'modules' of AI (vision, language, reasoning) to extract a holistic view of the world, and explain why they believe what they believe. He has published more than 50 scientific articles in top-tier journals and conferences, including CVPR, ICCV, ECCV, NIPS, ICML, UAI, *International Journal of Computer Vision*, etc. Research from his lab has been featured in *Bloomberg Business*, *The Boston Globe*, *MIT Technology Review*, *Newsweek*, and a number of popular press magazines and newspapers. He received the Carnegie Mellon Dean's Fellowship in 2007, two Google Faculty Research Awards in 2013 and 2015, Virginia Tech Teacher of the Week in 2013, Army Research Office (ARO) Young Investigator Program (YIP) award in 2014, the National Science Foundation (NSF) CAREER award in 2014, and Virginia Tech CoE outstanding new assistant professor award in 2015. His research is supported by NSF, ARO, ARL, ONR, DARPA, Amazon, Google, Microsoft, and NVIDIA. Webpage: http://computing.ece.vt.edu/ dbatra