Empirical Minimum Bayes Risk Prediction: How to extract an extra few% performance from vision models with just three more parameters

Vittal Premachandran*
National University of Singapore
vittal@comp.nus.edu.sg

Daniel Tarlow
Microsoft Research
dtarlow@microsoft.com

Dhruv Batra
Virginia Tech
dbatra@vt.edu

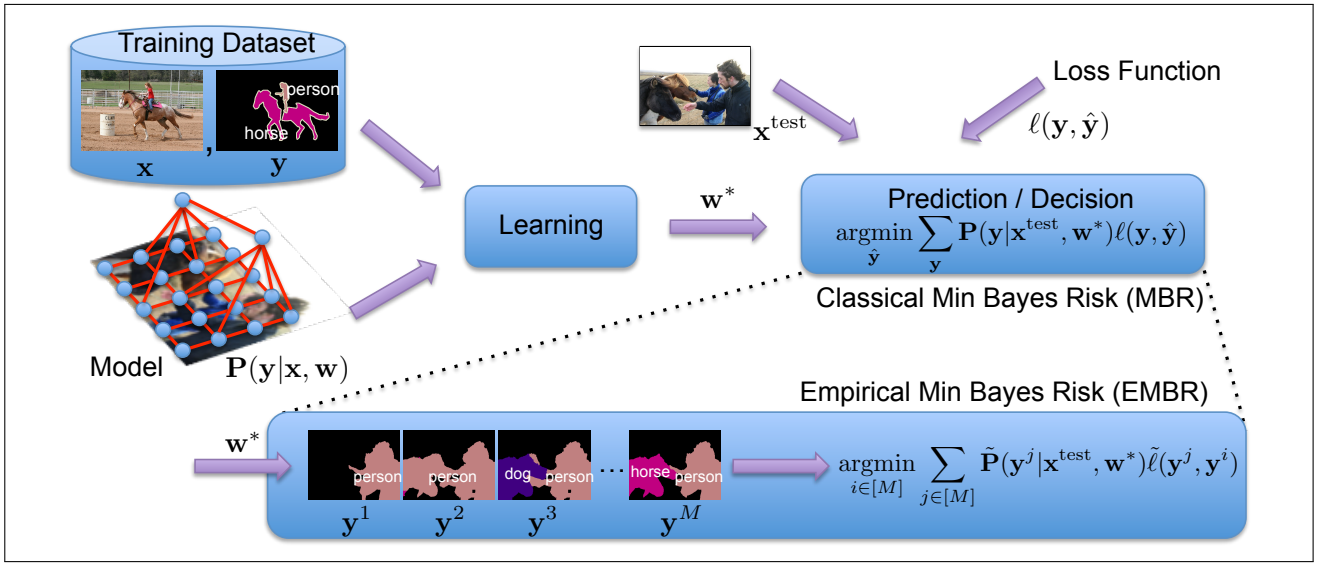


Figure 1. Classical Min Bayes Risk (MBR) vs. Empirical Min Bayes Risk (EMBR): Probabilistic reasoning involves (a) learning the parameters of our model from training data, and (b) making predictions or decisions by optimizing *Bayes Risk* or expected loss. We present a meta-algorithm (EMBR) that is motivated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

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 high-order task-specific measure. We demonstrate EMBR in several domains, taking existing state-of-the-art algorithms and improving performance up to ∼7%, simply with three extra parameters.*

*Part of the work was done when author was a student at Nanyang Technological University.

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 & poses) is to (a) formulate a model where parameters determine a scoring function, (b) choose parameters that optimize performance on a training set, and (c) 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 $\sim$7%.

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. We refer to this as the *Minimum Bayes Risk* (MBR) predictor.

While a direct application of Bayesian decision theory to most structured prediction problems is not feasible due to computational considerations, we show that it is nonetheless possible to efficiently construct structured predictors that incorporate the key ingredients – incorporating task-specific losses and 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 $\mathbf{Y} = \{\mathbf{y}^1, \ldots, \mathbf{y}^M\}$ 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 (see Fig. 2 for an illustration) –

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.

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, rather to train a predictor that utilizes ideas from decision theory and performs well on a 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.

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).

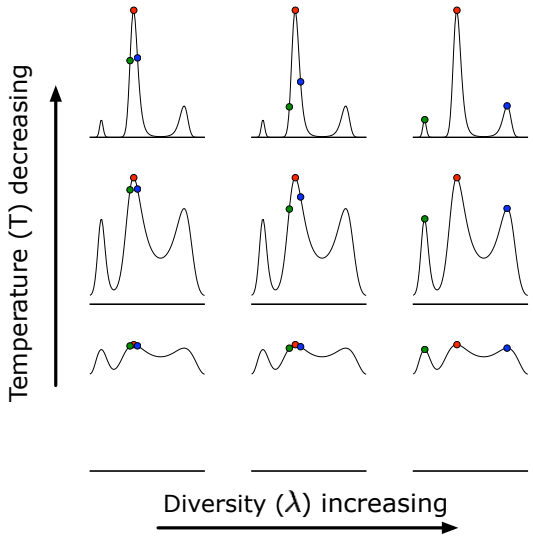


Figure 2: Illustration of the effect of two of the three parameters in our model. The colored dots represent the chosen candidates $\mathbf{Y}$, and the height of the curve at these points illustrates the $\tilde{\mathrm{P}}(\mathbf{y})$ value assigned to each candidate. These parameters are learned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

Our experiments on a range of problems – binary foreground-background segmentation, human body pose estimation, and 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 [29], we achieved state-of-art accuracies on the PARSE dataset, improving results by $\sim$7%.

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.* $y_u \in \mathcal{Y}_u = \{\text{sky, building, road, car}, \ldots\}$. 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, u.

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 VOC [8] $1 - \frac{\text{intersection}}{\text{union}}$ 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 [29].

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. We make minimal assumptions about the structure of the scoring function.

Assumptions. Our key computational assumption is that it is possible to efficiently compute $\operatorname{argmax}_{\mathbf{y}} S(\mathbf{y}; \mathbf{x})$ (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 disparity in hardness of maximization vs. summation over exponentially large spaces. For instance, a maximum bipartite matching can be found in $O(n^3)$ time with the Hungarian algorithm [14], but summing over all perfect matchings (*i.e.* computing the permanent) is #P-complete [26]. 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.* $\operatorname{argmax}_{\mathbf{y}} S(\mathbf{y}; \mathbf{x}) + \sum_u \theta_u(y_u)$, which allows leveraging methods such as DivMBest [3] to find a diverse set of solutions by solving multiple maximization problems where $S(\cdot)$ is augmented with diversity-encouraging unary potentials.

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

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

This problem is NP-hard in general [22]. 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 [19] and max-flow/min-cut [11, 13], respectively.

MBR Predictor. The Bayes Risk of predicting $\hat{\mathbf{y}}$ is defined as $\mathrm{BR}(\hat{\mathbf{y}}) = \mathbb{E}_{\mathrm{P}(\mathbf{y}|\mathbf{x})}[\ell(\mathbf{y}, \hat{\mathbf{y}})] = \sum_{\mathbf{y} \in \mathcal{Y}} \mathrm{P}(\mathbf{y}|\mathbf{x}) \ell(\mathbf{y}, \hat{\mathbf{y}})$. 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 (MBR) predictor is one that minimizes this expected risk, *i.e.*

$$\mathbf{y}^{MBR} = \operatorname*{argmin}_{\hat{\mathbf{y}} \in \mathcal{Y}} \; \mathrm{BR}(\hat{\mathbf{y}}) \tag{2a}$$

$$= \operatorname*{argmin}_{\hat{\mathbf{y}} \in \mathcal{Y}} \; \sum_{\mathbf{y} \in \mathcal{Y}} \mathrm{P}(\mathbf{y}|\mathbf{x}) \ell(\mathbf{y}, \hat{\mathbf{y}}). \tag{2b}$$

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 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 (ERM). 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. 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 $\tilde{\mathrm{P}}(\cdot)$.

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 $\mathbf{Y} = \{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^M\}$. Specifically, the predictor is defined as follows:

$$\mathbf{y}^{EMBR} = \operatorname*{argmin}_{\hat{\mathbf{y}} \in \mathbf{Y}} \sum_{\mathbf{y} \in \mathbf{Y}} \tilde{\mathrm{P}}(\mathbf{y}|\mathbf{x}) \tilde{\ell}(\mathbf{y}, \hat{\mathbf{y}}) \tag{3}$$

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 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 [24, 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).

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

Computation. If we construct a matrix of pairwise losses:

$$L = \begin{bmatrix} \tilde{\ell}(\mathbf{y}^1, \mathbf{y}^1) & \tilde{\ell}(\mathbf{y}^1, \mathbf{y}^2) & \cdots & \tilde{\ell}(\mathbf{y}^1, \mathbf{y}^M) \\ \tilde{\ell}(\mathbf{y}^2, \mathbf{y}^1) & \tilde{\ell}(\mathbf{y}^2, \mathbf{y}^2) & \cdots & \tilde{\ell}(\mathbf{y}^2, \mathbf{y}^M) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\ell}(\mathbf{y}^M, \mathbf{y}^1) & \tilde{\ell}(\mathbf{y}^1, \mathbf{y}^2) & \cdots & \tilde{\ell}(\mathbf{y}^M, \mathbf{y}^M) \end{bmatrix}, \tag{4}$$

and a vector stacking all approximate probabilities $\mathbf{p} = [\tilde{\mathrm{P}}(\mathbf{y}^1|\mathbf{x}) \ldots \tilde{\mathrm{P}}(\mathbf{y}^M|\mathbf{x})]^T$, then the EMBR predictor can be expressed with a single matrix-vector multiplication:

$$\mathbf{y}^{EMBR} = \operatorname*{argmin}_{\mathbf{y}^i, i \in [M]} \sum_{j \in [M]} \tilde{\mathrm{P}}(\mathbf{y}^j|\mathbf{x}) \tilde{\ell}(\mathbf{y}^j, \mathbf{y}^i) \tag{5a}$$

$$= \operatorname*{argmin}_{\mathbf{y}^i, i \in [M]} L\mathbf{p} \tag{5b}$$

The runtime of EMBR given $\mathbf{Y}$ 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.

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, λ.
{DivMBest}
for $m \in 1, \ldots, M$ **do**
  $S^m_\Delta(\mathbf{y}) \leftarrow S(\mathbf{y}) + \sum_{u \in \mathcal{V}} \sum_{m'=1}^{m-1} \lambda \cdot [\![y_u \neq y_u^{m'}]\!]$
  $\mathbf{y}^m \leftarrow \operatorname{argmax}_{\mathbf{y}} S^m_\Delta(\mathbf{y}; \mathbf{x})$
end for
{Scores to Probabilities}
for $i \in 1, \ldots, M$ **do**
  $\tilde{\mathrm{P}}(\mathbf{y}^i|\mathbf{x}) \leftarrow \frac{\exp\left\{\frac{1}{T} S(\mathbf{y}^i; \mathbf{x})\right\}}{\sum_{j=1}^{M} \exp\left\{\frac{1}{T} S(\mathbf{y}^j; \mathbf{x})\right\}}$
end for
{Prediction}
$i^* = \operatorname{argmin}_{i \in [M]} \sum_{j \in [M]} \tilde{\mathrm{P}}(\mathbf{y}^j|\mathbf{x}) \tilde{\ell}(\mathbf{y}^j, \mathbf{y}^i)$
return $\mathbf{y}^{i^*}$

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

$$\tilde{\mathrm{P}}(\mathbf{y}|\mathbf{x}) = \frac{\exp \frac{1}{T} S(\mathbf{y}; \mathbf{x})}{\sum_{\mathbf{y}' \in \mathbf{Y}} \exp \frac{1}{T} S(\mathbf{y}'; \mathbf{x})}, \tag{6}$$

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 [20] 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 [20]). Other approaches are possible. For example, [32] suggests using isotonic regression, [31] discusses calibrating Naive Bayes and decision tree classifiers, and [15] 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. Interestingly, our experimental results suggest that this choice is not crucial.

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

3.2. Producing Diverse High-Scoring Candidates

How should the candidate configurations $\mathbf{Y}$ 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 [2,16,30] that find the top M most probable solutions and (2) sampling-based algorithms [1, 21, 25]. 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 [3,18] 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*. [3], and the Perturb-and-MAP algorithm [17]. Alternatives that we did not experiment with but might be worthwhile exploring in future work are Herding [12, 27] and Multiple Choice Learning [9, 10]. This paper presents results only with DivMBest, which performed best; comparisons to [17] can be found in the supplemental document (available from the author webpages).

For sake of completeness, we briefly describe DivMBest. More details can be found in [3]. 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 $\Delta(\cdot,\cdot)$:

$$\mathbf{y}^M = \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}} \quad S(\mathbf{y};\mathbf{x}) \tag{7a}$$

$$s.t. \qquad \Delta(\mathbf{y},\mathbf{y}^m) \geq k_m \quad \forall m \in [M-1]. \tag{7b}$$

In general, this problem is NP-hard and Batra *et al*. [3] proposed to use the Lagrangian relaxation formed by dualizing the dissimilarity constraints $\Delta(\mathbf{y},\mathbf{y}^m) \geq k_m$:

$$f(\boldsymbol{\lambda}) = \max_{\mathbf{y}\in\mathcal{Y}} \; S_\Delta(\mathbf{y};\mathbf{x}) \doteq S(\mathbf{y};\mathbf{x}) + \sum_{m=1}^{M-1} \lambda_m \; (\Delta(\mathbf{y},\mathbf{y}^m) - k_m) \tag{8}$$

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

Following [3], we use Hamming (or weighted 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 (8) can be written as:

$$S_\Delta(\mathbf{y};\mathbf{x}) = S(\mathbf{y};\mathbf{x}) + \underbrace{\sum_{u\in\mathcal{V}} \left(\sum_{m=1}^{M-1} \lambda_m [\![y_u \neq y_u^m]\!] \right)}_{\text{Augmented Unary Score}}. \tag{9}$$

Thus, the maximization in (8) 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).

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 [3], 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 λ_M and T_M are tuned to maximize EMBR performance at M.
  - 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:

- Binary (foreground-background) interactive segmentation (Section 4.1),
- 2D articulated human body pose estimation on the

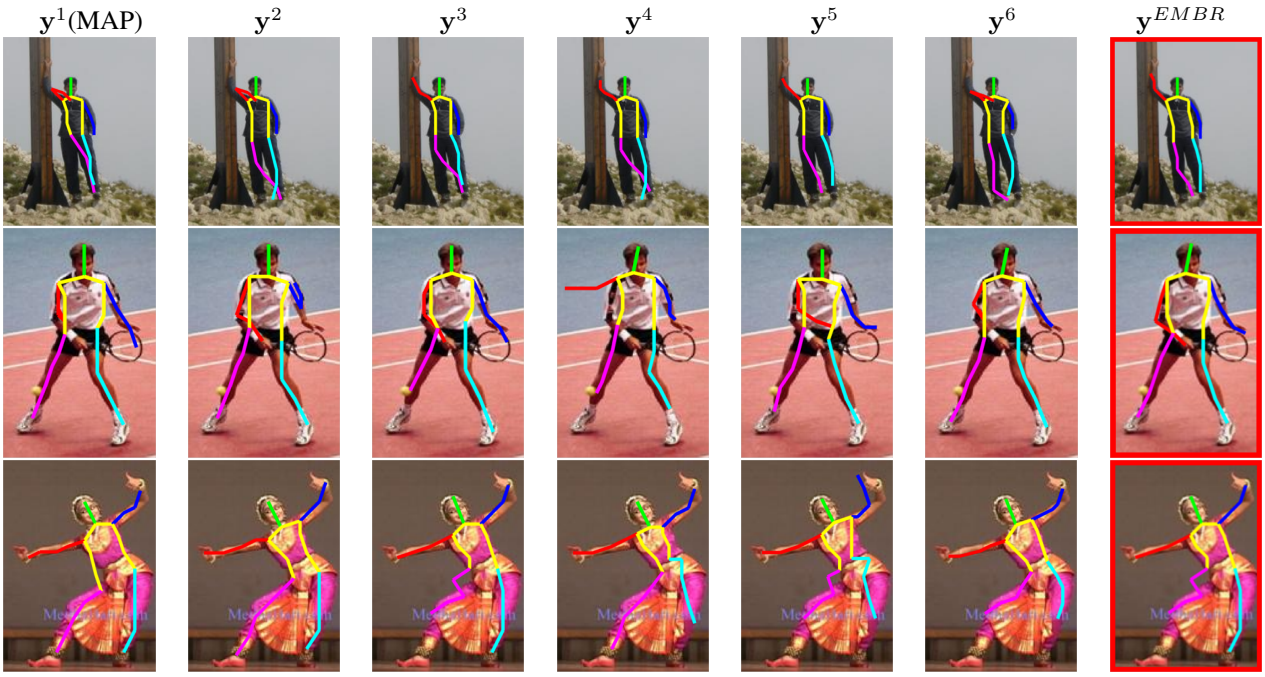


Figure 3: Qualitative Results: Within each row, the first column corresponds to MAP, the middle columns show the diverse solutions, and the last column shows the EMBR prediction. The top two rows show examples where EMBR selects a better pose than MAP, while the bottom row shows an example where MAP 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.

PARSE dataset [29] (Section 4.2), and
- Category-level segmentation on PASCAL VOC 2012 Segmentation Challenge dataset [7] (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 [29], PASCAL metric [8]). 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 [23], making exact MBR intractable.

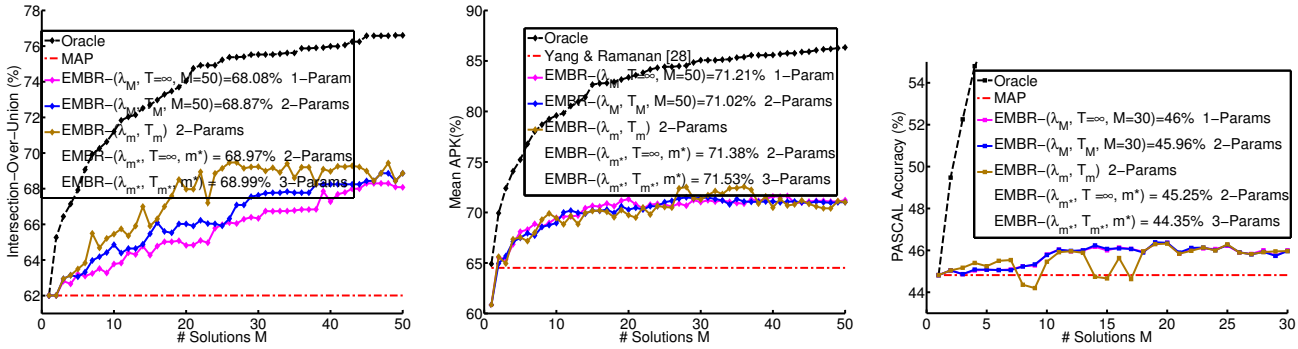
Baselines. On all experiments, we compare our approach against the natural baseline of MAP, which simply predicts the highest scoring solution and is indifferent to the setting of T, λ and M. In a manner similar to [3], we also report `oracle` accuracies, *i.e.* the accuracy of the *most accurate* solution in the set $\mathbf{Y}$. This forms an upper-bound on the performance of any predictor (including MAP and EMBR) which picks a single solution from the set $\mathbf{Y}$. In the pose estimation experiments (Section 4.2), we also compare against the results of Yang and Ramanan [29], which was the previous state-of-the-art on the PARSE dataset.

In our experiments, DivMBest consistently outperformed Perturb & MAP, so we only report results for DivMBest in this section. Perturb & MAP results along with a discussion of its performance appear in the supplementary materials.

Main theme in results. 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 [29] by $\sim 7\%$. 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 [3], 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



(a) Interactive Segmentation. (b) 2D Pose Estimation. (c) PASCAL VOC `val`.

Figure 4: Quantitative Results: We show the performance of different method vs 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.

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.

Results. Fig. 4a shows the performance of EMBR as a function of M. Due to the greedy nature of DivMBest, EMBR degenerates to MAP at $M = 1$. We can see that EMBR-$(\lambda_M, T_M, M = 50)$ outperforms MAP by $\sim 7\%$ (68.87%). The 1 and 3 parameter EMBRs perform similar, suggesting that default choices T and M work well.

4.2. Pose Estimation

Model. We replicate the setup of Yang and Ramanan [29], 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) [6]. Yang and Ramanan proposed a novel metric for measuring performance called Average Precision of Keypoints (APK) [29], 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 APK achieved by various methods versus M. We can see that all EMBR variants significantly outperform MAP[1]). EMBR-DivMBest-$(\lambda_M, T_M, M = 50)$ achieves a final mean-APK of 71.02%, and EMBR-$(\lambda_{m*}, T_{m*}, m*)$ performs slightly better by achieving 71.53%, which is a $\sim$ 7-percentage-point improvement over the previous state of the art of 64.5% [29].

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*. [4] – approximately 150 CPMC segments [5] are generated for each image, scored via Support Vector Regressors over second-order pooled features [4], and then greedily pasted. The sum of the scores of the pasted segments is the score of a segmentation, and DivMBest is used 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.

Results. Fig. 4c shows the PASCAL accuracy as a function of M. This is a difficult problem; EMBR-DivMBest-$(\lambda_M, T_M, M = 30)$ yields an improvement of only $\sim 1\%$ (45.96%). The 1 and 3 parameter EMBRs perform similar. Note that this is 2.2% below the current state of art [28] on VOC Segmentation, which also uses DivMBest solutions, but unlike us, makes use of significantly more sophisticated features to perform the re-ranking of solutions. In future, we plan to explore EMBR re-ranking with the features of [28].

5. Discussion

Instance-level vs Corpus-level Losses. In a number of settings, for instance PASCAL segmentation, the evalua-

[1]Note that in segmentation, the evaluation metric only cares about the predictions, not the scores associated with the predictions. In pose, 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 [29]) does not perform identical to EMBR at $M = 1$ (which uses the Bayes Gain as the score).

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

When is EMBR expected to work? 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. 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.

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.

Acknowledgements: DB was partially supported by the National Science Foundation under Grant No. 1353694, and the Army Research Office YIP Award W911NF-14-1-0180. We thank Varun Ramakrishna for helpful discussions during early stages of this work.

References

[1] A. Barbu and S.-C. Zhu. Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:1239–1253, August 2005. 4

[2] D. Batra. An Efficient Message-Passing Algorithm for the M-Best MAP Problem. In *Uncertainty in Artificial Intelligence*, 2012. 4

[3] D. Batra, P. Yadollahpour, A. Guzman-Rivera, and G. Shakhnarovich. Diverse M-Best Solutions in Markov Random Fields. In *ECCV*, 2012. 3, 5, 6

[4] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, pages 430–443, 2012. 7

[5] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010. 7

[6] 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. 7

[7] 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/challenges/VOC/voc2012/workshop/index.html. 6

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2):303–338, June 2010. 3, 6

[9] A. Guzman-Rivera, D. Batra, and P. Kohli. Multiple Choice Learning: Learning to Produce Multiple Structured Outputs. In *Proc. NIPS*, 2012. 5

[10] A. Guzman-Rivera, P. Kohli, D. Batra, and R. Rutenbar. Efficiently enforcing diversity in multi-output structured prediction. In *AISTATS*, 2014. 5

[11] P. Hammer. Some network flow problems solved with pseudo-boolean programming. *Operations Research*, 13:388–399, 1965. 3

[12] F. Huszár and D. Duvenaud. Optimally-weighted herding is bayesian quadrature. *arXiv preprint arXiv:1204.1664*, 2012. 5

[13] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004. 3

[14] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38, March 1957. 3

[15] A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632. ACM, 2005. 4

[16] D. Nilsson. An efficient algorithm for finding the M most probable configurations in probabilistic expert systems. *Statistics and Computing*, 8:159–173, 1998. 10.1023/A:1008990218483. 4

[17] G. Papandreou and A. Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *ICCV*, pages 193–200, Nov. 2011. 5

[18] D. Park and D. Ramanan. N-best maximal decoders for part models. In *ICCV*, 2011. 5

[19] J. Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *AAAI*, 1982. 3

[20] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances In Large Margin Classifiers*, 1999. 4

[21] J. Porway and S.-C. Zhu. C^4: Exploring multiple solutions in graphical models by cluster sampling. *PAMI*, 33(9):1713 –1727, 2011. 4

[22] S. E. Shimony. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, 68(2):399–410, August 1994. 3

[23] D. Tarlow and R. S. Zemel. Structured output learning with high order loss functions. In *AISTATS*, 2012. 6

[24] M. Taylor, J. Guiver, S. Robertson, and T. Minka. Softrank: optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 77–86. ACM, 2008. 4

[25] Z. Tu and S.-C. Zhu. Image segmentation by data-driven Markov Chain Monte Carlo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:657–673, May 2002. 4

[26] L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189 – 201, 1979. 3

[27] M. Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1121–1128. ACM, 2009. 5

[28] P. Yadollahpour, D. Batra, and G. Shakhnarovich. Discriminative re-ranking of diverse segmentations. In *CVPR*, 2013. 7

[29] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. *PAMI*, To Appear. 2, 3, 6, 7

[30] C. Yanover and Y. Weiss. Finding the M most probable configurations using loopy belief propagation. In *NIPS*, 2003. 4

[31] B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *In Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, 2001. 4

[32] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 694–699. ACM, 2002. 4