Optimal Decisions from Probabilistic Models: the Intersection-over-Union Case

Sebastian Nowozin
Microsoft Research
Sebastian.Nowozin@microsoft.com

Abstract

A probabilistic model allows us to reason about the world and make statistically optimal decisions using Bayesian decision theory. However, in practice the intractability of the decision problem forces us to adopt simplistic loss functions such as the 0/1 loss or Hamming loss and as result we make poor decisions through MAP estimates or through low-order marginal statistics. In this work we investigate optimal decision making for more realistic loss functions. Specifically we consider the popular intersection-over-union (IoU) score used in image segmentation benchmarks and show that it results in a hard combinatorial decision problem. To make this problem tractable we propose a statistical approximation to the objective function, as well as an approximate algorithm based on parametric linear programming. We apply the algorithm on three benchmark datasets and obtain improved intersection-over-union scores compared to maximum-posterior-marginal decisions. Our work points out the difficulties of using realistic loss functions with probabilistic computer vision models.

1. Introduction

A popular viewpoint on computer vision tasks is to posit them as a probabilistic inference task. The classic recipe is as follows [28, 15, 25], shown in Figure 1: 1. specify a probabilistic model $p(x, z)$ of the quantity of interest z and the observed signal x; 2. observe x and obtain the conditional posterior distribution $p(z|x)$ of the quantity of interest; 3. using the posterior infer summary statistics or decisions.

In the last decade this recipe has been successfully used and adapted. In particular, nowadays models for semantic image segmentation and human pose estimation are often discriminatively and hence conditionally specified as $p(z|x; \theta)$, where θ is a high-dimensional parameter vector [49, 30]. This changes the first step of the recipe, but the 2nd and 3rd steps remain unchanged.

A large body of work is available describing probabilistic models and sophisticated inference procedures, but the decision task (step 3.) is often overlooked. For example, [49] simply output the most likely state $\operatorname{argmax}_z p(z|x)$ as an inference result. Deciding for the most likely state corresponds to a particular choice of *loss function*, namely the *0-1 loss*, $\ell(z, y) = 1_{\{z \neq y\}}$. This loss is unrealistic and does not match the way we assess prediction quality in computer vision benchmarks. Our work fixes this mismatch; we start from decision theory [2, 32]: given our beliefs $p(z|x)$ and a *loss* or *utility function* that measures the benefit of a decision y when the true world state is z, we would like to make *optimal decisions*, that is minimize the expected loss or equivalently maximize expected utility. In this work we consider optimal decision making with the *intersection-over-union utility* (IoU) [11] when using probabilistic models.

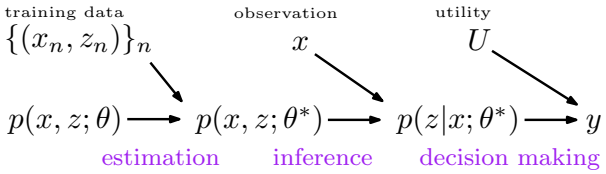

Figure 1. Typical workflow when working with a probabilistic model: 1. *Estimation*: using training data to find a good model within a specified model class, 2. *Inference*: infering a posterior distribution given an observation x, 3. *Decision Making*: using posterior beliefs and a given utility to make optimal decision.

Another method to deal with task-specific loss functions is to directly learn a decision function using empirical risk minimization (ERM) [50], popular in computer vision through the structured SVM. However, there are advantages to maintaining a probabilistic model: first, we can use different loss functions at test time; second, probabilities facilitate combining separate submodels, each of which can be separately designed and trained; and third, there is currently no *consistent* ERM method for structured prediction [26].

1.1. Prior Work

Previous work has investigated empirical risk minimization (ERM) [50] with higher-order loss functions; in computer vision the first work is [4] who showed how to learn with the intersection-over-union loss restricted to bounding boxes. For segmentation Ranjbar et al. [35] approximate the intersection-over-union score for empirical risk minimization, and for the binary case, Tarlow and Zemel [44, 45] propose efficient inference procedures for the *loss-augmented inference problem*. Krähenbühl and Koltun [21] extend [10] to handle higher-order loss functions. Beside the higher-order IoU score other higher-order loss functions such as

label count losses and their approximability with simple models has been investigated by Pletscher and Kohli [33] and Küttel and Ferrari [23]. This prior work puts emphasis in learning with higher-order losses; however, for learning they all use the empirical risk learning objective. Our work is different in that we follow the "classic" recipe and first construct a probabilistic model, then solve the optimal decision task using the higher-order loss function. The only prior work that has considered this optimal decision task is Tarlow and Adams [43], who proposed a greedy algorithm.

Contributions. Our work makes the following contributions. 1. We derive a closed-form statistical approximation, (8), to the intersection-over-union score for the case of conditionally independent beliefs. 2. We propose an algorithm, Alg. 2, for making approximately optimal decisions under the intersection-over-union score. 3. We experimentally evaluate the algorithm on multiple data sets and demonstrate an increase in intersection-over-union score.

2. Problem and Statistical Approximation

We first formalize what it means to make optimal decisions under uncertainty [2].

Problem 1 (Optimal Decision Making)**.** Given a set $\mathcal{Y} = \{1, \dots, K\}$ of classes, an index set $\mathcal{V} = \{1, \dots, n\}$, an utility function $U : \mathcal{Y}^{\mathcal{V}} \times \mathcal{Y}^{\mathcal{V}} \to \mathbb{R}$, and a probability distribution p over the set $\mathcal{Y}^{\mathcal{V}}$, find the optimal decision

$$y^* = \operatorname*{argmax}_{y \in \mathcal{Y}^{\mathcal{V}}} \mathbb{E}_{z \sim p} \left[U(z, y) \right]. \tag{1}$$

Intuitively, (1) optimizes our expected utility under every possibility z, weighted by our beliefs about the state of the world as encoded in p. The problem could be generalized by allowing more general decision domains but for most applications in computer vision (1) is sufficient.

Solving (1) is difficult for three reasons. First, it is an optimization problem over a large set whose size grows exponentially in $|\mathcal{V}|$. Second, the expectation $\mathbb{E}_{z \sim p}[\cdot]$ requires computation of an average over the same large set. Third, the function U may not have enough structure for efficient computation. In practice the tractability of (1) depends on whether we can replace the expectation expression with a simple closed-form solution. For example, in the simple case when U is the negative Hamming loss it decomposes additively over $\mathcal{V}$ so that the expectation commutes and each variable can be optimized separately, yielding the maximum posterior marginal (MPM) solution [25]. This is not the case for more complicated utility functions, as we now illustrate.

2.1. Intersection-over-Union Utility

The intersection-over-union score is a popular benchmark score for semantic segmentation. It has become popular in the computer vision community due to the PASCAL VOC segmentation challenges [11]. It is defined as follows.

Definition 2 (Intersection-over-Union Utility)**.** Given a ground truth assignment $y \in \mathcal{Y}^{\mathcal{V}}$ and a prediction $z \in \mathcal{Y}^{\mathcal{V}}$, the intersection-over-union utility is $U_{\text{iou}} = \frac{1}{K} \sum_{k=1}^{K} U_{\text{iou}}^{(k)}(z, y)$, where

$$U_{\text{iou}}^{(k)}(z, y) = \frac{\sum_{i \in \mathcal{V}} 1_{\{z_i = k \wedge y_i = k\}}}{\sum_{i \in \mathcal{V}} 1_{\{z_i = k \vee y_i = k\}}} \tag{2}$$

is the *per-class intersection-over-union utility* and $1_{\{\text{predicate}\}}$ is the indicator function which is one in case the predicate is true and zero otherwise.

For the PASCAL VOC segmentation benchmark the set $\mathcal{V}$ is the set of all pixels in all test set images. Therefore (2) contains ratios of sums over all pixels and does not decompose over pixels. This property makes (1) difficult to solve. In particular we obtain the following specialization of (1) to the intersection-over-union utility.

$$\mathbb{E}_{z \sim p}[U_{\text{iou}}(z, y)] = \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}_z \left[\frac{\sum_{i \in \mathcal{V}} 1_{\{z_i = k \wedge y_i = k\}}}{\sum_{i \in \mathcal{V}} 1_{\{z_i = k \vee y_i = k\}}} \right]. \tag{3}$$

Equation (3) contains an expectation of a ratio, which does not have a closed form solution. We now show how this expectation can be approximated using techniques from asymptotic statistics. In particular, we have the following result due to Rice [36, 37].

Proposition 3 (Rice [36])**.** *Let S and T be two real-valued random variables with finite moments of all order and with $P(T = 0) = 0$ and $\mathbb{E}T \neq 0$. Then*

$$\mathbb{E}\left[\frac{S}{T}\right] = \frac{\mathbb{E}S}{\mathbb{E}T} + \sum_{j=1}^{\infty} \Psi_j, \tag{4}$$

where

$$\Psi_j = (-1)^j \frac{(\mathbb{E}S)\langle^j T\rangle + \langle S,^j T\rangle}{(\mathbb{E}T)^{j+1}}. \tag{5}$$

Here $\langle^j T\rangle$ denotes the j'th central moment of T so that $\langle^1 T\rangle = 0$ and we write $\langle S,^j T\rangle = \mathbb{E}[(S - \mathbb{E}S)(T - \mathbb{E}T)^j]$.

In applications of this result the infinite sum is often truncated. Similar but less general results are discussed in [39] and [42, Section 4.10], and a discussion of expectations of ratios is given by Heijmans [17].

To apply Proposition 3 to our expectation (3) we define $S_k = \sum_{i \in \mathcal{V}} 1_{\{z_i = k \wedge y_i = k\}}$ and $T_k = \sum_{i \in \mathcal{V}} 1_{\{z_i = k \vee y_i = k\}}$ for each $k \in \mathcal{Y}$ so that we have $\mathbb{E}[U_{\text{iou}}^{(k)}(z, y)] = \mathbb{E}[S_k / T_k]$. We are interested in expanding $\mathbb{E}[S_k / T_k] = \mathbb{E}[S_k] / \mathbb{E}[T_k] + \sum_{j=1}^{\infty} \Psi_j$, so we need to compute $\mathbb{E}[S_k]$, $\mathbb{E}[T_k]$, and Ψ_j.

For general p these expectations do not have simple closed-form solutions, but we can make progress

if we assume conditional independence, *i.e.* $p(z|x) = \prod_{i\in\mathcal{V}} p_i(z_i|x)$, where x would be an observed image. Assuming conditional independence is a natural modeling assumption in many probabilistic models in computer vision and does not imply *unconditional* independence between different z_i's. For example, conditional independence is assumed when using random forests [40] to predict pixel marginals in semantic segmentation. Under this assumption we have the following result, as also derived by [43, 21].

Proposition 4. *For the above definition of S_k and T_k and for a conditionally independent distribution $p(z) = \prod_{i\in\mathcal{V}} p_i(z_i)$ we have, as a function of the decision y,*

$$\mathbb{E}[S_k] = \sum_{i\in\mathcal{V}} p_i(k)\, 1_{\{y_i=k\}}, \tag{6}$$

$$\mathbb{E}[T_k] = \sum_{i\in\mathcal{V}} (1_{\{y_i=k\}} + p_i(k)\, 1_{\{y_i\neq k\}}). \tag{7}$$

We give the proof in the supplementary materials. We now have simple closed-form expressions for $\mathbb{E}[S_k]$ and $\mathbb{E}[T_k]$. Let us examine the additional expansion terms Ψ_j. For the first term Ψ_1 we have the following result.

Proposition 5. *For the intersection-over-union utility $\mathbb{E}[S_k/T_k]$ we have $\Psi_1 = 0$ in expansion (4).*

We give the proof in the supplementary materials. It is possible to analyze the higher order terms Ψ_j, $j \geq 2$, but we stop here because Proposition 5 already implies a strong guarantee on the quality of the approximation. In particular by standard results for the delta method, [42, Section 4.3], we have asymptotically for $n \to \infty$ that

$$\mathbb{E}\left[\frac{S_k}{T_k}\right] = \frac{\mathbb{E}S_k}{\mathbb{E}T_k} + O(n^{-1}). \tag{8}$$

Together with (6) and (7) this yields a closed-form approximation to $\mathbb{E}[S_k/T_k]$. Note that in a typical computer vision application n will be large ($n \gg 10^3$) and therefore the approximation error will be small. This guarantee also explains the empirical success of this approximation as used by [43, 21] where the approximation was derived heuristically and described as "surrogate" and "relaxation". We now examine the guarantee (8) experimentally.

2.2. Experimental Validation of the Approximation

We perform the following simulation experiment. For each of $N = 5000$ binary variables we sample a probability distribution p_i from a Dirichlet prior with uniform parameter $\alpha = 0.2$. We then select $n \in \{100, 200, \ldots, N\}$ and perform a Monte Carlo evaluation of $\mathbb{E}[S_1/T_1]$, $\mathbb{E}[S_1]$, and $\mathbb{E}[T_1]$, where the expectation is evaluated by simulating $y_i \sim p_i$ and $z_i \sim p_i$ one hundred thousand times. We then compare $\mathbb{E}[S_1/T_1]$ with $\mathbb{E}[S_1]/\mathbb{E}[T_1]$ and plot n against the error on a log-log plot, see Figure 2.

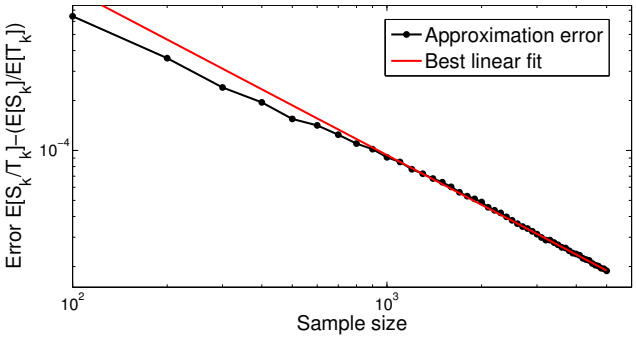

Figure 2. Approximation error of (8) as a function of n. Note the log-log axes. The results are obtained using Monte Carlo simulation of (8). The predicted $O(n^{-1})$ behavior of the error is clearly visible and the fitted slope coefficient is $-0.9954 \approx -1$.

Because the theory predicts a $O(n^{-1})$ behavior of the error we fit a line to the second half of the error observations. The slope of the line agrees with an $O(n^{-1})$ error.

3. Method

We now develop a method for making optimal decisions under the intersection-over-union utility function. By specializing (1) to the intersection-over-union utility and by using approximation (8) with the closed form solutions (6) and (7) we obtain the following problem, as in [43].

Problem 6 (Approximately Optimal Decision Making under the Intersection-over-Union Utility)**.** Given marginal beliefs p_i over K classes, find the prediction by solving

$$\begin{aligned} \max_{\lambda} \quad & \frac{1}{K}\sum_{k=1}^{K} \frac{\sum_{i\in\mathcal{V}} p_i(k)\, \lambda_{i,k}}{\sum_{i\in\mathcal{V}} [p_i(k) + (1-p_i(k))\, \lambda_{i,k}]} \\ \text{sb.t.} \quad & \sum_{k=1}^{K} \lambda_{i,k} = 1, \qquad i \in \mathcal{V}, \\ & \lambda_{i,k} \in \{0,1\}, \qquad i \in \mathcal{V}, \quad k \in \mathcal{Y}. \end{aligned} \tag{9}$$

Here $\lambda_{i,k}$ is an indicator variable, selecting for each y_i one label $\lambda_{i,k} = 1$ so that $y_i = k$. Problem 6 is a sum of ratios of affine functions. The number of ratios equals the number K of classes in the problem. This type of optimization problem is known in the optimization community by different names: *sum-of-ratio linear fractional program* [38], *multiple ratio hyperbolic 0-1 programming problem* [46], and *generalized linear fractional programs* [6]. For binary variables these problems are known to be NP-hard in general even for a single ratio. But if we relax the constraint $\lambda_{i,k} \in \{0,1\}$ to the interval $\lambda_{i,k} \in [0,1]$, the known results are surprising: a sum containing only a single ratio is solvable in polynomial time by the Charnes-Cooper transformation [7], a sum of two ratios is also solvable in polynomial-time [20] and is known to be pseudo-convex under restrictive conditions [6, Section 7.5] that do

not apply in our case. For three and more ratios the problem remains hard even in the relaxed domain [38].

One sensible approach to solving (9) is proposed in [43], where the authors use *greedy* local search, iteratively changing one variable at a time to improve the objective. The method is simple to implement, efficient, and maintains a feasible solution at all times; we describe it in the supplementary materials. However, in many optimization problems—for example discrete energy minimization problems [18]—such a simple greedy approach can be outperformed by approaches that make use of the problem structure. This is generally the case when the problem has an intrinsic complexity leading to local optima where the greedy method could get stuck in. Our proposed method uses the problem structure more globally but is based on solving a relaxation and therefore could lead to non-integral solutions. The question which method is preferable is then an empirical question of how intrinsically complex the objective (9) really is. If it is sufficiently simple the greedy method may work better, but if it is complex we may see the more global method to work better.

Our approach to solving (9) will be to iteratively improve a solution y by optimizing over blocks of variables corresponding to pairs of labels, holding all other variables fixed. Doing so yields a tractable subproblem where only two fractions appear in (9) and we use Konno's algorithm [20] to solve the corresponding relaxed subproblem.

3.1. Optimizing over Two Classes

The strategy to optimize over a large subset of variables depending on the current candidate solution has been used in the α-β-swap graphcut algorithm [5] and is also used for solving a number of hard combinatorial problems in the framework of very large scale neighborhood search (VLSN) [1], [30, Section 4.5.1]. Applying it to (9) we obtain the following subproblem restricted to two classes.

Problem 7 (Two Class Problem)**.** For two distinct classes $k_1, k_2 \in \mathcal{Y}$, and a reference labeling $y \in \mathcal{Y}^{\mathcal{V}}$, let $\mathcal{W} \subseteq \mathcal{W}_{k_1 k_2} = \{i \in \mathcal{V} : y_i = k_1 \vee y_i = k_2\}$, $\mathcal{W} \neq \emptyset$, that is, an arbitrary non-empty subset of the variables currently labeled k_1 or k_2. To find the optimal labeling restricted to the set $\mathcal{W}$, we set $\lambda_{i,k_1} = \mu_i$, $\lambda_{i,k_2} = 1 - \mu_i$, and solve

$$\begin{aligned} \max_{\mu} \quad & \frac{a_0 + \sum_{i\in\mathcal{W}} a_i\mu_i}{b_0 + \sum_{i\in\mathcal{W}} b_i\mu_i} + \frac{c_0 + \sum_{i\in\mathcal{W}} c_i\mu_i}{d_0 + \sum_{i\in\mathcal{W}} d_i\mu_i}, \qquad (10) \\ \text{sb.t.} \quad & \mu_i \in \{0,1\}, \qquad i \in \mathcal{W}, \end{aligned}$$

where from the terms involving k_1 and k_2 in (9) we have

$$\begin{aligned} a_0 &= \textstyle\sum_{i\in\mathcal{V}\setminus\mathcal{W}} p_i(k_1)\, 1_{\{y_i=k_1\}}, \\ a_i &= p_i(k_1), \quad i \in \mathcal{W}, \\ b_0 &= \textstyle\sum_{i\in\mathcal{V}} p_i(k_1) + \sum_{i\in\mathcal{V}\setminus\mathcal{W}} (1 - p_i(k_1))\, 1_{\{y_i=k_1\}}, \\ b_i &= 1 - p_i(k_1), \quad i \in \mathcal{W}, \\ c_0 &= \textstyle\sum_{i\in\mathcal{V}\setminus\mathcal{W}} p_i(k_2)\, 1_{\{y_i=k_2\}} + \sum_{i\in\mathcal{W}} p_i(k_2), \\ c_i &= -p_i(k_2), \quad i \in \mathcal{W}, \\ d_0 &= |\mathcal{W}| + \textstyle\sum_{i\in\mathcal{V}\setminus\mathcal{W}} \left(p_i(k_2) + (1 - p_i(k_2))\, 1_{\{y_i=k_2\}}\right), \\ d_i &= p_i(k_2) - 1, \quad i \in \mathcal{W}. \end{aligned}$$

To solve Problem 7 we adapt the algorithm of Konno et al. [20]. This algorithm is based on first relaxing $\mu_i \in [0,1]$, then transforming (10) by means of the Charnes-Cooper transformation [7] followed by a parametric simplex method for linear programming [8].

The Charnes-Cooper transformation defines a set u_i of variables and one additional variable u_0 and introduces the coupling constraints $u_0 = 1/(d_0 + \sum_{i\in\mathcal{W}} d_i\mu_i)$, and $u_i = \mu_i u_0$. Problem 7 can now be rewritten in the new variables as a sum of a ratio and a linear function as follows.

$$\begin{aligned} \max_{u_0,u} \quad & \frac{a_0 u_0 + \sum_{i\in\mathcal{W}} a_i u_i}{b_0 u_0 + \sum_{i\in\mathcal{W}} b_i u_i} + c_0 u_0 + \sum_{i\in\mathcal{W}} c_i u_i, \\ \text{sb.t.} \quad & d_0 u_0 + \sum_{i\in\mathcal{W}} d_i u_i = 1, \\ & u_i \le u_0, \quad i \in \mathcal{W}, \\ & u_0 \ge 0, \quad u_i \ge 0, \quad i \in \mathcal{W}. \end{aligned}$$

To get rid of the final ratio we introduce an auxiliary parameter $\xi = b_0 u_0 + \sum_{i\in\mathcal{W}} b_i u_i$, and obtain the following family of parametric linear programs.

$$\begin{aligned} \mathcal{P}(\xi) = \frac{1}{\xi}\Big\{ \max_{u_0,u} \quad & \xi c_0 u_0 + \xi \sum_{i\in\mathcal{W}} c_i u_i + a_0 u_0 + \sum_{i\in\mathcal{W}} a_i u_i, \\ \text{sb.t.} \quad & b_0 u_0 + \sum_{i\in\mathcal{W}} b_i u_i = \xi, \qquad (11) \\ & d_0 u_0 + \sum_{i\in\mathcal{W}} d_i u_i = 1, \\ & u_i \le u_0, \quad i \in \mathcal{W}, \qquad (12) \\ & u_0 \ge 0, \quad u_i \ge 0, \quad i \in \mathcal{W} \Big\}. \end{aligned}$$

The algorithm of Konno et al. [20] is an efficient parametric linear programming method to globally maximize $\mathcal{P}$ by sweeping ξ from the minimum possible value to the maximum possible value, tracing the optimal solution for every value of ξ. The solution path is a sequence of quadratic functions in ξ and we visualize a typical example in Figure 3. Compared to standard parametric programming for linear programs [3, Section 5.5] as used in computer vision [19], problem $\mathcal{P}(\xi)$ is more challenging because the parameter ξ appears both in the objective function as well as in the constraint (11). As a result, if we want to increase ξ while maintaining an optimal solution, we need to perform either a primal simplex update or a dual simplex update [8].

3.2. Implementation

Algorithm 1 describes the main loop of the algorithm. The objective function—we show an example in Figure 3—is defined between ξ_{min} and ξ_{max} and composed of piecewise quadratic functions in ξ. The algorithm starts with

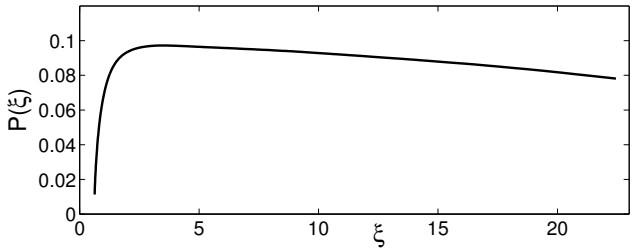

Figure 3. Typical example of function $\mathcal{P}(\xi)$ to be maximized in $\xi \in [\xi_{\text{min}}, \xi_{\text{max}}]$ by the parametric simplex method. $\mathcal{P}(\xi)$ is piecewise quadratic, in this case comprised of 441 segments.

Algorithm 1 Konno's algorithm [20] for (10)

1: **function** PARASIMPLEX($a_0, a_i, b_0, b_i, c_0, c_i, d_0, d_i$)
2: $\xi_{\text{min}} \leftarrow b_0/d_0$
3: $\xi_{\text{max}} \leftarrow (b_0 + \sum_{i\in\mathcal{W}} b_i)/(d_0 + \sum_{i\in\mathcal{W}} d_i)$
4: $i^* \leftarrow \text{argmax}_{i\in\mathcal{W}}(a_i + \xi_{\text{min}} c_i)$
5: $\mathcal{B} \leftarrow \{u_0, u_{i^*}\} \cup \{s_i : i \in \mathcal{W}\}$ ▷ feasible basis
6: $\mathcal{B} \leftarrow$ PRIMALSIMPLEX($\mathcal{B}$) ▷ optimal basis
7: $\xi \leftarrow \xi_{\text{min}}$
8: **while** $\xi < \xi_{\text{max}}$ **do**
9: $(\underline{\xi}, \overline{\xi}, \text{type}, v) \leftarrow$ COMPUTERANGE($\mathcal{B}$)
10: $\hat{\xi} \leftarrow \text{argmax}_{\xi\in[\underline{\xi},\overline{\xi}]} \mathcal{P}(\xi)$ ▷ 1D quadratic
11: Compute maximizer $\mu(\hat{\xi})$
12: $\xi \leftarrow \overline{\xi}$ ▷ next interval
13: **if** type = primal **then**
14: $\mathcal{B} \leftarrow$ PRIMALSIMPLEX($\mathcal{B}, v$) ▷ v enters $\mathcal{B}$
15: **else if** type = dual **then**
16: $\mathcal{B} \leftarrow$ DUALSIMPLEX($\mathcal{B}, v$) ▷ v leaves $\mathcal{B}$
17: **end if**
18: **end while**
19: **return** global maximizer $\mu(\xi^*)$ of $\mathcal{P}$
20: **end function**

$\xi = \xi_{\text{min}}$ and in each iteration increases ξ such that the next interval defining another quadratic function is reached (lines 12). Within each interval we obtain the closed form solution (line 10) and keep track of the global optimal solution ξ^*. We transform (12) to the standard slack form $u_i - u_0 + s_i = 0$, where $s_i \geq 0$ is a slack variable.

Implementing Algorithm 1 is challenging because both a primal and dual simplex method need to be implemented [8, 3]. With care a matrix-free implementation can be achieved by deriving closed-form solutions to two linear systems related to the constraint system defining $\mathcal{P}(\xi)$ and by using a non-standard update strategy to the reduced costs [48]. Our optimized C++ source code is available from the author's homepage. Despite our optimizations to the implementation the overall runtime of Algorithm 1 remains $O(n^2)$. We show typical runtimes in Fig. 4 and observe that they are empirically independent of the coefficients.

3.3. Optimizing (9) by Local Search

Our proposed method is fast enough to optimize predictions for a single image. Unfortunately, because the current use of IoU in segmentation benchmarks applies to the entire test data set the quadratic runtime prohibits us to optimize over the decision variables of all images simultaneously. We therefore divide the set of all variables into random subsets of a specified size. Each subset is optimized sequentially and because each problem corresponds to a neighborhood around the current solution we can guarantee monotonic improvement of the objective (9). The overall procedure is shown in Algorithm 2. We use a subproblem size $M = 2000$ for the experiments. For a subproblem size of $M = 1$ Algorithm 2 becomes the greedy method of [43].

Algorithm 2 Large Neighborhood Local Search for (9)

function OPTIMIZEIOU($p, \mathcal{V}, K$, iters, M)
  $y_i \leftarrow \text{argmax}_{k=1,\dots,K} p_i(k)$ ▷ initialize with MAP
  for $t = 1, \dots,$ iters **do**
    $S \leftarrow$ RANDOMSHUFFLE($(1, 2, \dots, K)$)
    for $j = 1, \dots, \lfloor K/2 \rfloor$ **do** ▷ in parallel
      $k_1 \leftarrow S(2j-1), \quad k_2 \leftarrow S(2j)$
      $\mathcal{W}_{k_1k_2} \leftarrow (i \in \mathcal{V} : y_i = k_1 \vee y_i = k_2)$
      $\mathcal{W}_{k_1k_2} \leftarrow$ RANDOMSHUFFLE($\mathcal{W}_{k_1k_2}$)
      for $r = 0, 1, \dots, \lfloor |\mathcal{W}_{k_1k_2}|/M \rfloor$ **do**
        $\mathcal{W} \leftarrow \mathcal{W}_{k_1k_2}(rM + 1, \dots, (r+1)M)$
        Compute $a_0, a_i, b_0, b_i, c_0, c_i, d_0, d_i$
        $\mu \leftarrow$ PARASIMPLEX(a, b, c, d)
        for $i \in \mathcal{W}$ **do**
          $y_i \leftarrow k_1$ if $\mu_i \geq \frac{1}{2}$, k_2 otherwise
        end for
      end for
    end for
  end for
  return approximately optimal decision y
end function

4. Experiments and Results

We now validate the key contribution; that is, we would like to demonstrate that given the same marginal posteriors our method can optimize the intersection-over-union score. We use three baselines: the greedy method [43], the simple

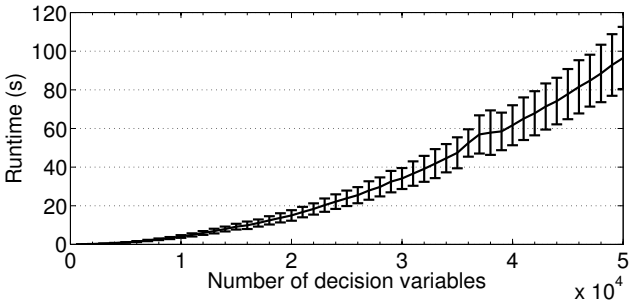

Figure 4. $O(n^2)$ runtime of Algorithm 1 as a function of n with one unit standard deviation over 10 replications with randomly chosen problem coefficients. For $n = 2000$, the size we adopt in Algorithm 2, we have a runtime of 156ms on a single core of an Intel Xeon E5-1650 3.20GHz CPU.

MAP decision (RF-MAP/MPM), which in our case is identical to the maximum posterior marginal (MPM) [25], and a simple "inverse weighted" MAP method (iwMAP). For this method we sum for each class k our total beliefs over all variables, as $\gamma_k = \sum_{i \in \mathcal{V}} p_i(k)$, then compute the inverse-weighted beliefs $\tilde{p}_i(k) = (p_i(k)/\gamma_k)/(\sum_\ell p_i(\ell)/\gamma_\ell)$. We then compute the MAP decision using $\tilde{p}$. The intuition is that the reweighting makes our new total beliefs over all classes uniform and hence can give more probability mass to apriori less likely classes. For our methods we also report the *believed* accuracy and IoU score. The believed IoU score is simply our objective (9), and the believed accuracy is $(\sum_{i \in \mathcal{V}} p_i(y_i))/|\mathcal{V}|$. We run Algorithm 2 for 30 and 60 iterations to obtain approximately optimal decisions RF-IoU-opt30 and RF-IoU-opt60 under the IoU utility.

For all experiments we use marginals obtained using a random forest applied to each pixel [41]. We use the decision tree implementation released by the authors of [31] and augment the features used with simple Histogram-of-Gradients, integral image, and image location features but otherwise use the default settings. We use three semantic scene segmentation data sets: LabelMeFacade [13], Stanford Background [16], and PASCAL VOC 2012 [11]. Although we downscale images for optimizing our decision objective, we always assess accuracy and intersection-over-union score on the original full resolution images.

4.1. LabelMeFacade Dataset

The LabelMeFacade data set [13] uses nine semantic labels describing facade elements. There are 845 test images which we downscale by a factor of 0.125 for a total of 4,429,952 decision variables. The baseline results from random forests in Table 1 are comparable in accuracy (71.28%) with the state-of-the-art (67.33%), but yield an improved IoU score after optimization (IoU-opt60, IoU-greedy).

Method	Acc. belief/actual	IoU belief/actual
RF [13]	- / 49.06	- / -
ICF [14]	- / 60.68	- / -
ICFHGWS+ [14]	- / 67.33	- / -
RF-MAP/MPM	78.69 / **71.28**	34.79 / 31.74
RF-iwMAP	55.30 / 53.76	32.35 / 32.01
RF-IoU-greedy [43]	75.75 / 69.60	38.75 / **35.96**
RF-IoU-opt30	75.60 / 69.47	38.67 / 35.91
RF-IoU-opt60	75.72 / 69.56	38.70 / 35.91

Table 1. LabelMeFacade results (845 test images) [13].

4.2. Stanford Background Dataset

The Stanford background dataset [16] uses 8 semantic classes and 715 images; the standard setup is five-fold cross validation. We produce full resolution marginal posteriors for each of the five test-folds and then downscale the images by a factor of 0.25 for a total of 3,377,600 decision variables. The results are reported in Table 2. Our multiclass accuracy (74.7%) is roughly comparable with the state-of-the-art (81.9% in [24]). Among our results the MAP/MPM decision has the highest accuracy and the IoU-opt60/IoU-greedy decisions have the highest IoU score.

Method	Acc. belief/actual	IoU belief/actual
Gould et al. [16]	- / 76.4	- / -
Kumar/Koller [22]	- / 79.42	- / -
Lempitsky et al. [24]	- / **81.90**	- / -
Tighe/Lazebnik [47]	- / 77.5	- / -
Farabet et al. [12]	- / 81.4	- / -
RF-MAP/MPM	75.75 / **74.70**	50.42 / 50.51
RF-iwMAP	69.21 / 69.90	47.67 / 49.59
RF-IoU-greedy [43]	75.15 / 74.58	51.79 / **52.36**
RF-IoU-opt30	75.14 / 74.56	51.78 / 52.34
RF-IoU-opt60	75.14 / 74.58	51.79 / **52.36**

Table 2. Five fold cross-validation results (715 images) for the Stanford Background dataset [16].

4.3. PASCAL VOC 2012 Dataset

The PASCAL VOC semantic segmentation benchmark [11] is a challenging segmentation dataset with 21 classes. We train a random forest on the "train" subset and produce a posterior for "val", and also train on the "trainval" subset and produce a posterior for "test". We downscale the posterior with a factor or 0.125 to obtain 4,104,672 decision variables for "val" (1449 images) and 4,123,073 decision variables for "test" (1456 images). The results are shown in Table 3 and Table 4. For this challenging dataset we do not come close to the state-of-the-art performance; however, on "val" the IoU score achieved by the MAP/MPM decision is improved from 3.51% to 11.08%. Because both decisions are obtained from the same posterior marginals this increase is directly attributable to our algorithm. As with the previous datasets the best accuracy is achieved by the MAP/MPM decision (RF-MAP/MPM), the best IoU score by the IoU-decision (RF-IoU-greedy).

5. Discussions

The results demonstrate that methods which explicitly optimize for the IoU performance outperform methods that are unaware of the utility function. However, the simple greedy local search method [43] is surprisingly good, slightly outperforming our global method. This is an indication that the objective (9) is simple and does not have many local optima. For practical purposes one can therefore use the more efficient greedy method. Note that this behaviour is different to what is observed in discrete energy minimization problems [18] where greedy methods like iterated conditional modes (ICM) are regularly outperformed

Method	Acc. belief/actual	IoU belief/actual
RF-MAP/MPM	80.47 / **73.33**	3.82 / 3.51
RF-iwMAP	21.00 / 24.61	4.09 / 6.68
RF-IoU-greedy [43]	68.43 / 69.17	7.60 / **11.65**
RF-IoU-opt30	65.98 / 66.95	7.20 / 10.68
RF-IoU-opt60	68.03 / 68.63	7.38 / 11.08

Table 3. PASCAL VOC 2012 validation set results (1449 images).

Method	Acc. belief/actual	IoU belief/actual
BONN O2PCPMC	— / —	— / 47.0
NUS	— / —	— / **47.3**
UVA CRF	— / —	— / 11.3
RF-MAP/MPM	84.25 / —	4.01 / 3.61
RF-iwMAP	23.35 / —	4.02 / 7.53
RF-IoU-greedy [43]	73.63 / —	7.67 / **12.47**
RF-IoU-opt30	71.27 / —	7.27 / 11.49
RF-IoU-opt60	72.92 / —	7.58 / 12.27

Table 4. PASCAL VOC 2012 test set results (1456 images).

by more global methods. Further analysis of the objective (9) is needed to explain this observation.

More generally our work raises a number of issues when using probabilistic models in computer vision and some specific points with the IoU utility.

Computational tractability of higher-order utility/loss functions. We have assumed conditionally independent beliefs p_i for each variable $i \in \mathcal{V}$. Arguably this is a strong simplifying assumption. Yet, even with this assumption the decision problem (Problem 6) remains a hard combinatorial optimization problem. It is reasonable to assume that for more complex models the task would remain at least as hard. Given this intractability, it is then interesting to note that in the recent "marginal-based learning" framework of [10] it is possible to do gradient-based parameter optimization using the intersection-over-union utility as demonstrated in [21]. Although the framework is based on *empirical risk minimization* (ERM) it retains the probabilistic inference method as internal component. It may be the case that ERM is better suited for tractably learning with higher-order loss functions because the hard part—handling the higher-order loss—can be handled during training, whereas test-time prediction remains efficient [45, 33].

The importance of calibrated probabilities. We have seen in the experimental results that our believed accuracy and believed intersection-over-union scores deviate from the actual scores. Could it be that we have "overfitted" our decisions by optimizing (1)? No, this is not possible: we already have beliefs p, and we merely make the best decision under what we believe. The explanation for the observed deviation is that our probabilities p are not perfectly *calibrated*. That is, we may believe that a variable y_i is in a certain state with a certain probability, but we systematically over- or underestimate this probability. As a consequence the expectation (1) does not correctly estimate the consequences of our decisions, as also observed in [43, 34].

How can we improve the calibration of our posterior probabilities? If our probabilistic model class is sufficiently accurate then known results guarantee that we will eventually be well-calibrated [9]. However, in realistic computer vision applications our model is *misspecified* and even when using Bayesian inference there is a systematic miscalibration [27]. Then a more pragmatic approach to calibration may be to use bagging and *recalibration* methods, [29].

Sampling theory and decomposability. In the VOC segmentation challenge [11] the IoU utility is evaluated on the confusion matrix of the *entire* test data set. We argue against this on the basis that the data set was created by sampling a set of images at random from a large population of images on the internet. Therefore one unit of observation is a single image and the utility function should decompose along these independent units. Applying the IoU utility on the entire data set is not meaningful: if you do, your decisions on a particular image can potentially be improved by considering future *independent* observables. On the other hand, applying the IoU utility to individual images is meaningful.

6. Conclusion

In this work we have studied the intersection-over-union score, a popular higher-order utility function used in image segmentation benchmarks. Starting with decision theory we proposed a statistical approximation and algorithm for making approximately optimal decisions under the IoU utility. The experiments have confirmed improved results. We hope to stimulate further research into learning and optimal decision making with higher-order loss functions.

Acknowledgements. The author thanks Danny Tarlow and Christoph Lampert for discussions and feedback.

References

[1] R. K. Ahuja, Ö. Ergun, J. B. Orlin, and A. P. Punnen. A survey of very large-scale neighborhood search techniques. In *Proc. Workshop on Discrete Opt.* Elsevier, 2002.

[2] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, 1985.

[3] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

[4] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *ECCV*, 2008.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell*, 23(11):1222–1239, 2001.

[6] A. Cambini and L. Martein. *Generalized Convexity and Optimization: Theory and Applications*. Springer, 2008.

[7] A. Charnes and W. W. Cooper. Programming with linear fractional functionals. *Naval Research Logistics Quarterly*, 9:181–186, 1962.

[8] V. Chvátal. *Linear Programming*. WH Freeman, 1983.

[9] A. P. Dawid. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 77:605–613, 1982.

[10] J. Domke. Learning graphical model parameters with approximate marginal inference. *IEEE Trans. Pattern Anal. Mach. Intell*, 35(10):2454–2467, 2013.

[11] M. Everingham, L. V. Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL voc2012 challenge results.

[12] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell*, 35(8):1915–1929, 2013.

[13] B. Fröhlich, E. Rodner, and J. Denzler. A fast approach for pixelwise labeling of facade images. In *ICPR*, 2010.

[14] B. Fröhlich, E. Rodner, and J. Denzler. Semantic segmentation with millions of features: Integrating multiple cues in a combined random forest approach. In *ACCV*, 2012.

[15] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell*, 6(6):721–741, 1984.

[16] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009.

[17] R. Heijmans. When does the expectation of a ratio equal the ratio of expectations? *Statistical Papers*, 40, 1999.

[18] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, J. Lellmann, N. Komodakis, and C. Rother. A comparative study of modern inference techniques for discrete energy minimization problems. In *CVPR*. IEEE, 2013.

[19] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *ICCV*, 2007.

[20] H. Konno, Y. Yajima, and T. Matsui. Parametric simplex algorithm for solving a special class of nonconvex minimization problems. *J. Glob. Opt.*, 1:65–81, 1991.

[21] P. Krähenbühl and V. Koltun. Parameter learning and convergent inference for dense random fields. In *ICML*, 2013.

[22] P. M. Kumar and D. Koller. Efficiently selecting regions for scene understanding. In *CVPR*, 2010.

[23] D. Küttel and V. Ferrari. Learning to approximate global shape priors for figure-ground segmentation. In *British Machine Vision Conference (BMVC)*, 2013.

[24] V. S. Lempitsky, A. Vedaldi, and A. Zisserman. Pylon model for semantic segmentation. In *NIPS*, 2011.

[25] J. L. Marroquin, S. K. Mitter, and T. A. Poggio. Probabilistic solutions of ill-posed problems in computational vision. *J. of the Am. Stat. Assoc.*, 82(397):293, 1987.

[26] D. McAllester. Generalization bounds and consistency for structured labeling. In *Predicting Structured Data*. MIT Press, 2007.

[27] U. K. Müller. Risk of Bayesian inference in misspecified models, and the sandwich covariance matrix. *Econometrica*, 81(5):1805–1849, 2013.

[28] D. Mumford and A. Desolneux. *Pattern Theory: The Stochastic Analysis of Real-World Signals*. CRC, 2010.

[29] A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *ICML*, 2005.

[30] S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3-4):185–365, 2011.

[31] S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao, and P. Kohli. Decision tree fields. In *ICCV*, 2011.

[32] G. Parmigiani and L. Inoue. *Decision Theory: Principles and Approaches*. John Wiley and Sons, Ltd., 2009.

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

[34] P. Pletscher, S. Nowozin, P. Kohli, and C. Rother. Putting MAP back on the map. In *DAGM*, 2011.

[35] M. Ranjbar, G. Mori, and Y. Wang. Optimizing complex loss functions in structured prediction. In *ECCV*, 2010.

[36] S. H. Rice. The expected value of the ratio of correlated random variables. (unpublished note).

[37] S. H. Rice. A stochastic version of the price equation reveals the interplay of deterministic and stochastic processes in evolution. *BMC Evolutionary Biology*, 8(1), 2008.

[38] S. Schaible and J. Shi. Fractional programming: the sum-of-ratios case. In M. Fukushima and Y. Yuan, editors, *Proc. 2nd Japanese-Sino Opt. Meeting*, pages 219–230, 2003.

[39] H. Seltman. Approximations for mean and variance of a ratio. (unpublished note).

[40] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

[41] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[42] C. G. Small. *Expansions and Asymptotics for Statistics*. CRC Press, 2010.

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

[44] D. Tarlow and R. S. Zemel. Big and tall: Large margin learning with high order loss. In *CVPR Workshop on Inference in Graphical Models with Structured Potentials*, 2011.

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

[46] M. Tawarmalani, S. Ahmed, and N. V. Sahinidis. Global optimization of 0–1 hyperbolic programs. *Journal of Global Optimization*, 24:385–416, 2002.

[47] J. Tighe and S. Lazebnik. Superparsing - scalable nonparametric image parsing with superpixels. *International Journal of Computer Vision*, 101(2):329–349, 2013.

[48] J. A. Tomlin. On pricing and backward transformation in linear programming. *Math. Prog.*, 6:42–47, 1974.

[49] Z. Tu, X. Chen, A. L. Yuille, and S. C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of Computer Vision*, 63(2):113–140, 2005.

[50] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2nd edition, 2000.