Optimizing Complex Loss Functions in Structured Prediction

Mani Ranjbar, Greg Mori, Yang Wang

School of Computing Science
Simon Fraser University, Canada

Abstract. In this paper we develop an algorithm for structured prediction that optimizes against *complex* performance measures, those which are a function of false positive and false negative counts. The approach can be directly applied to performance measures such as F_β score (natural language processing), intersection over union (image segmentation), Precision/Recall at k (search engines) and ROC area (binary classifiers). We attack this optimization problem by approximating the loss function with a piecewise linear function and relaxing the obtained QP problem to a LP which we solve with an off-the-shelf LP solver. We present experiments on object class-specific segmentation and show significant improvement over baseline approaches that either use simple loss functions or simple compatibility functions on VOC 2009.

1 Introduction

Solving challenging vision problems such as image understanding, image segmentation, and video retrieval arguably requires the use of "complex" structured models – those incorporating relationships between multiple input and output entities. Evidence for this comes from state-of-the-art approaches to the aforementioned problems. For example, Hoiem et al. [1] formulate image understanding models that tie together object locations, camera parameters, and surfaces. Blaschko and Lampert [2] localize objects using an efficient solution to a structured output regression model. Desai et al. [3] learn models for simultaneously detecting all objects in an image. Non-max suppression and contextual object co-occurrence statistics are learned in a discriminative fashion. Image segmentation is a canonical example of structured labeling problem (e.g. [4–6]).

For many of these problems the natural performance measures are also "complex" – ones that do not decompose into a simple sum of individual terms measured over each output entity. Examples of such measures are object detection scores that penalize for multiple detections on a single true positive (e.g PASCAL VOC [7]) and region labeling or object segmentation scores that penalize for over and under labeling or segmentation (e.g. intersection / union score). Typical methods for solving these problems learn parameters against other performance measures, e.g. Hamming loss for segmentation, and then apply post-processing techniques (e.g. non-maximum suppression in object detection) to address the

structure in the performance measure. Instead, in this paper we develop an algorithm for linking these two together and formulate learning as jointly considering the complex, structured relationships between output variables in the model and in the learning objective.

The main contribution of this paper is developing a general algorithm for addressing this type of learning problem with complex models and those complex loss functions which are a function of false positive and false negative counts. We specifically apply it to image segmentation, but note that the algorithm can be applied more broadly. We experiment with a standard Markov Random Field (MRF) segmentation model that contains both unary terms for labeling pixels and pairwise terms on the labels of neighbouring pixels. We show that learning the parameters to this model under an objective directly tied to the performance measure significantly improves performance relative to baseline algorithms on the PASCAL VOC Segmentation Challenge.

2 Previous Work

A wide range of learning algorithms exist. Despite technical differences, all of these approaches rely on a performance measure to define what is a "good" result. Based on the complexity of the performance measure, two general approaches to optimize it are imaginable, formulate the learning problem to directly optimize this measure, or approximate this measure with a simpler one and try to optimize it aiming to indirectly optimize the original complex performance measure. We will call the former "direct optimization" and the latter "indirect optimization".

Due to the complexity of some performance measures, e.g., average precision and intersection over union, many state-of-the-art approaches in different challenges exploit an indirect optimization. Looking at PASCAL VOC challenge 2009 [7], for example, average precision and intersection over union are defined as performance measures for detection and segmentation tasks respectively, but methods for both tasks use indirect optimizations for solving these problems.

Structured prediction has become popular in computer vision. Taskar et al. [8] and Tsochantaridis et al. [9] have the same formulation for structured prediction using a max-margin criterion. Both of them need to solve the "most violated constraint" [9], or loss augmented inference [8] in each iteration of their gradient descent to find the optimal parameters. They assume the loss function is decomposable and therefore solving for the most violated constraint is as hard as doing the inference without the loss function, which is assumed to be tractable. Joachims [10] proposed an approach to efficiently compute the most violated constraint when the loss function is not decomposable, but limited the underlying model by allowing only simple compatibility functions, those which involve only a single input and output. In this paper we provide an algorithm for structured prediction with a complex compatibility function that optimizes against complex performance measures, those which are a function of false positive and false negative counts.

3 Background

To create a foundation for the proposed approach, we start with an overview of our learning formulation. Next, we discuss the two common approaches, one based on a simple loss function with a complex compatibility function and the other with complex loss function and simple compatibility function. We call a loss function simple if it can be decomposed into loss on individual training samples. Likewise, a compatibility function is called simple if it only depends on a single sample point and its ground-truth label. Finally, we propose a framework to incorporate certain complex loss functions and complex compatibility functions in structured prediction.

3.1 Problem Formulation

The goal of our learning problem is defined as finding a function $h \in \mathcal{H}$ from the hypothesis space $\mathcal{H}$ given training samples $\mathrm{S} = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N))$ that optimizes the expected prediction performance on the new samples S' of size n'.

$$R^{\Delta}(h) = \int \Delta((h(\boldsymbol{x}'_1), h(\boldsymbol{x}'_2), \ldots, h(\boldsymbol{x}'_{n'})), (y'_1, y'_2, \ldots, y'_{n'}))dPr(S'). \quad (1)$$

In general, the loss function Δ cannot be decomposed into a linear combination of a loss function δ over individual samples. But, for simplicity, most discriminative learning algorithms (e.g. SVM) assume decomposibility and i.i.d. samples, which allows for rewriting Eq. 1 as

$$R^{\Delta}(h) = R^{\delta}(h) = \int \delta(h(\boldsymbol{x}'), y')dPr(\boldsymbol{x}', y'). \quad (2)$$

Instead of solving the estimated risk in Eq. 2, learning algorithms approximate that with empirical risk $\hat{R}^{\delta}$ defined as

$$\hat{R}^{\delta}(h) = \frac{1}{n}\sum_{i=1}^{N} \delta(h(\boldsymbol{x}_i), y_i). \quad (3)$$

For non-decomposable loss functions, such as F_1 score or intersection over union, optimizing Eq. 2 does not provide the desired answer. Rather, we are interested in finding an algorithm that can directly optimize the empirical risk based on the sample loss,

$$\hat{R}^{\Delta}_{S}(h) = \Delta((h(\boldsymbol{x}_1), h(\boldsymbol{x}_2), \ldots, h(\boldsymbol{x}_n)), (y_1, y_2, \ldots, y_n)). \quad (4)$$

Note that finding an $h \in \mathcal{H}$ that optimizes Eq. 4 for an arbitrary loss function Δ can be computationally challenging.

3.2 Structured Prediction Learning

For non-decomposable loss functions, one can reformulate the SVM based on the idea of multivariate prediction [10]. Instead of having a mapping function $h : \mathcal{X} \rightarrow \mathcal{Y}$ from a single example $\boldsymbol{x}$ to its label y, where $\boldsymbol{x} \in \mathcal{X}$ and $y \in \{-1, +1\}$, we look at all examples at once and try to learn a mapping function $\bar{h} : \mathcal{X} \times \cdots \times \mathcal{X} \rightarrow \bar{\mathcal{Y}}$, where $\bar{\mathcal{Y}} \in \{-1, +1\}^N$. We define $\bar{\boldsymbol{x}} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$, and $\boldsymbol{y} = (y_1, \ldots, y_N)$.

We can define the best labeling using a linear discriminant function

$$\bar{h}(\bar{\boldsymbol{x}}) = \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \boldsymbol{w}^T \Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}'). \tag{5}$$

Here, function Ψ measures the compatibility of the data points and their assigned labels. If we define the Ψ function as a simple form

$$\Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}') = \sum_{i=1}^{N} y_i' \boldsymbol{x}_i, \tag{6}$$

that only depends on individual training points and their labels, the optimal labeling sequence is

$$\arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \boldsymbol{w}^T \Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}') = \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \sum_{i=1}^{N} y_i' \boldsymbol{w}^T \boldsymbol{x}_i = (h(\boldsymbol{x}_1), \ldots, h(\boldsymbol{x}_N)), \tag{7}$$

which is exactly the same as the optimal labeling in SVM.

One way of incorporating a loss function Δ in SVM formulation is *Margin Rescaling*[9],

$$\min_{\boldsymbol{w}, \xi \geq 0} \|\boldsymbol{w}\|^2 + C\xi \tag{8}$$

$$s.t.\ \forall \boldsymbol{y}' \in \bar{\mathcal{Y}} \backslash \boldsymbol{y},\ \boldsymbol{w}^T \left[\Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}) - \Psi(\bar{\boldsymbol{x}}, \bar{y}')\right] \geq \Delta(\boldsymbol{y}, \boldsymbol{y}') - \xi \tag{9}$$

Similar to the original SVM formulation, ξ in Eq. 8 is an upper bound on $\Delta(\bar{h}(\bar{\boldsymbol{x}}), \boldsymbol{y})$[10].

The guarantee for convergence in polynomial time, the potential for incorporating complex loss functions in the objective and good performance in practice are the most important reasons why structured prediction has garnered much attention in computer vision recently.

In the standard approaches for solving Eq. 8, the output vector, $\tilde{\boldsymbol{y}}$, corresponding to the most violated constraint should be found repeatedly [9],

$$\tilde{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \Delta(\boldsymbol{y}, \boldsymbol{y}') + \boldsymbol{w}^T \Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}'). \tag{10}$$

Finding $\tilde{\boldsymbol{y}}$ is computationally challenging given an arbitrary loss function, $\Delta(\boldsymbol{y}, \boldsymbol{y}')$, and compatibility function, $\Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}')$. However, solving Eq. 10 in two special cases has been shown to be efficient. We categorize these approaches based on the simplicity of their Δ and Ψ functions. We call a loss function simple if it can be decomposed into individual training samples. Likewise, a compatibility function is called simple if it decomposes over single sample points and their ground-truth labels.

3.3 Simple Δ, Complex Ψ

Optimizing the parameters of a MRF structure when the loss function can be decomposed into the loss of individual samples falls into this category. One popular application in this category is foreground-background segmentation with Hamming loss, which is defined as

$$\Delta_H = \sum_i \mathbb{1}_{[y_i \neq y'_i]}. \tag{11}$$

Szummer et al. [6] have employed this formulation and reported promising results for interactive segmentation.

Decomposibility of the loss function results in a MRF form for Eq. 10, because the loss function can be treated as another unary term that adds up to the unary terms of the compatibility function. Assuming binary labels, this MRF can be solved efficiently using graphcut.

The advantage of this approach is to exploit pairwise connections, but it is only tractable for decomposable loss functions.

3.4 Complex Δ, Simple Ψ

The other special case presented by Joachims [10], is when the Ψ function has a simple form of

$$\Psi(\bar{\boldsymbol{x}}, \boldsymbol{y'}) = \sum_{i=1}^{N} y'_i \boldsymbol{x}_i. \tag{12}$$

If the loss function, Δ, is just a function of true positive (TP), false positive (FP) and false negative (FN), then there are at most $N_p \times N_n$ distinct loss values, where N_p and N_n represent the number of positive and negative training examples, respectively. Hence, Eq. 10 can be solved by iterating over all loss values and maximizing $\boldsymbol{w}^T \Psi(\boldsymbol{x}, \boldsymbol{y'})$ subject to the value of TP, FP and FN [10].

Unlike the approach of Szummer et al. [6], many standard accuracy measures that lead to non-decomposable loss functions, such as F_β score (natural language processing), intersection over union (image segmentation), Precision/Recall at k (web search engines) and ROC area (binary classifiers) can be directly optimized by this approach. However, this method cannot benefit from the pairwise interactions of training samples, which are shown to be advantageous in many applications, such as object detection [3] and scene interpretation [1].

4 Proposed Approach: Solving Complex Δ, Complex Ψ

Discussing the advantages and shortcomings of the previous methods, we now propose an approach to directly optimize certain complex loss functions in a Markov network. Here, we can optimize non-decomposable accuracy measures, such as F_β and intersection over union and still be able to benefit from pairwise interactions between training points.

We choose to follow the general framework of Structural$_{\text{SVM}}$ [9], shown in Eq. 8. Solving Eq. 8 requires finding the most violated constraint (Eq. 10) at

each iteration and modifying the parameter vector $\boldsymbol{w}$ accordingly [9]. We propose a novel method to efficiently solve for an approximate most violated constraint for certain non-decomposable loss functions in presence of pairwise terms in the compatibility function, Ψ.

We can summarize the proposed approach as

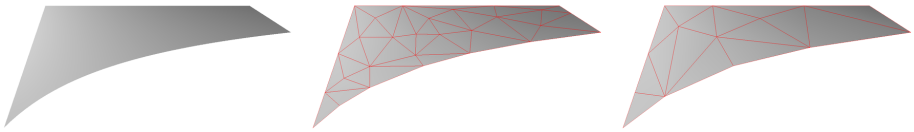
1. Replacing the original non-decomposable loss function with a piecewise linear approximation,
2. Writing the problem of finding the most violated constraint as a quadratic program,
3. Converting the quadratic program to a linear program and solve the relaxed problem.

4.1 Piecewise Linear Approximation

Many standard accuracy measures, including the one presented in the previous section, share the property that they can be computed from the contingency table[1]. Given the number of positive and negative examples, N_p and N_n, the loss function corresponding to these accuracy measures is just a function of FP and FN. Using piecewise linear approximation, we can write

$$\Delta(FP, FN) \simeq \tilde{\Delta}(FP, FN) = \sum_{j=1}^{M} \mathbb{1}_{[(FP,FN)\in\mathfrak{R}_j]} \{\alpha_j FP + \beta_j FN + \gamma_j\} \quad (13)$$

where, M is the number of subregions (pieces), α_j, β_j and γ_j represent the j^{th} plane coefficients and $\mathfrak{R}_j$s are the subregions that partition the space spanned by FP and FN.



a b c

Fig. 1. Intersection over union loss surface in FP and FN space. a) Exact surface, b) a piecewise linear approximation with 40 subregions, c) a piecewise linear approximation with 15 subregions.

As an example, Figure 1 illustrates the intersection over union loss function,

$$\Delta_{\frac{\cap}{\cup}}(FP, FN) = \frac{FN + FP}{N_p + FP}, \quad (14)$$

along with its piecewise linear approximations using 15 and 40 pieces.

Given the subregion $\mathfrak{R}_j$, the original non-linear loss function is a linear function of FP and FN. The next step is to substitute the approximated loss function, $\tilde{\Delta}$ into Eq. 10 and solve for the most violated constraint.

[1] Is just a function of TP, FP, TN and FN.

4.2 Forming the Quadratic Program

Choosing the right form of Ψ function is crucial to achieve high performance. In segmentation, for example, employing only unary terms in the Ψ function that model the relationship between an observed pixel and its label result in a lack of smoothness in the labeling. Hence, methods usually incorporate pairwise terms in the Ψ function to smooth the output labeling. We define our Ψ with unary and pairwise terms as

$$\Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}) = \sum_i (2y_i - 1)\phi_u(\boldsymbol{x}_i) + \sum_i \sum_{j \in \mathcal{N}_i} (y_i + y_j - 2y_i y_j)\phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{15}$$

Here $\mathcal{N}_i$ is the set of neighbors of sample i and we have assumed $y \in \{0, 1\}$. We rewrite Eq. 10 with approximated loss function, $\tilde{\Delta}$ as

$$\tilde{\boldsymbol{y}}* = \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \tilde{\Delta}(\boldsymbol{y}, \boldsymbol{y}') + \boldsymbol{w}^T \Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}') \tag{16}$$

$$= \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \tilde{\Delta}(\boldsymbol{y}, \boldsymbol{y}') + \boldsymbol{w}_u^T \sum_i (2y_i' - 1)\phi_u(\boldsymbol{x}_i) \tag{17}$$

$$+ \boldsymbol{w}_p^T \sum_i \sum_{j \in \mathcal{N}_i} (y_i' + y_j' - 2y_i' y_j')\phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

where $\boldsymbol{w} = [\boldsymbol{w}_u; \boldsymbol{w}_p]$ (concatenation of the two).

Note that $FP = \sum_i (1 - y_i)y_i'$ and $FN = \sum_i y_i(1 - y_i')$, where y_i is the true label and y_i' is the predicted label for the i^{th} example. If we assume that the loss values fall in subregion $\mathfrak{R}_k$, we can write Eq. 17 as

$$\tilde{\boldsymbol{y}}* = \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \Bigg(\alpha_k \sum_i (1 - y_i)y_i' + \beta_k \sum_i y_i(1 - y_i') + \gamma_k + \tag{18}$$

$$\boldsymbol{w}_u^T \sum_i (2y_i' - 1)\phi_u(\boldsymbol{x}_i) + \boldsymbol{w}_p^T \sum_i \sum_{j \in \mathcal{N}_i} (y_i' + y_j' - 2y_i' y_j')\phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j) \Bigg).$$

Note that Eq. 18 only includes the predicted label y' in linear and quadratic forms. Hence, we can write a quadratic program based on Eq. 18 subject to the loss values being in subregion $\mathfrak{R}_k$,

Maximize:

$$\alpha_k \sum_i (1 - y_i)y_i' + \beta_k \sum_i y_i(1 - y_i') + \gamma_k + \tag{19}$$

$$\sum_i (2y_i' - 1)[\boldsymbol{w}_u^T \phi_u(\boldsymbol{x}_i)] + \sum_i \sum_{j \in \mathcal{N}_i} (y_i' + y_j' - 2y_i' y_j')[\boldsymbol{w}_p^T \phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j)]$$

Subject to:

$$\left\{ \sum_i (1 - y_i)y_i', \sum_i y_i(1 - y_i') \right\} \in \mathfrak{R}_k, \quad i = 1, \ldots, N \tag{20}$$

In order to have linear constraints in Eq. 20, the boundary of all subregions should be definable as a linear function of $\boldsymbol{y}'$. One way is to separate the subregions by straight lines. If for example, we partition the space spanned by FP and FN into triangles (Fig. 1b,c) then Eq. 20 will be substituted by three linear constraints corresponding to the three sides of the triangle.

4.3 Converting Quadratic Program to Linear Program

The quadratic program in Eq. 19 is potentially non-convex, since there is no constraint on the coefficients of the objective function. So, instead of looking for a local optima of this non-convex function, we relax the problem (MAP-MRF LP relaxation [11]) by introducing some variables that substitute the quadratic terms in the objective function and form a linear program, which is convex. In detail, we introduce four new variables corresponding to four different possible configurations of a pair of labels as follows.

$$\mathfrak{y}_{ij}^{00} \equiv (1-y'_i)(1-y'_j),\ \mathfrak{y}_{ij}^{01} \equiv (1-y'_i)y'_j,\ \mathfrak{y}_{ij}^{10} \equiv y'_i(1-y'_j),\ \mathfrak{y}_{ij}^{11} \equiv y'_i y'_j. \quad (21)$$

We also add a set of constraints to relate the introduced variables to y' variables. The final linear program is

Maximize:

$$\alpha_k \sum_i (1-y_i)y'_i + \beta_k \sum_i y_i(1-y'_i) + \gamma_k + \quad (22)$$
$$\sum_i (2y'_i - 1)[\boldsymbol{w}_u^T \phi_u(\boldsymbol{x}_i)] + \sum_i \sum_{j \in \mathcal{N}_i} (\mathfrak{y}_{ij}^{01} + \mathfrak{y}_{ij}^{10})[\boldsymbol{w}_p^T \phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j)]$$

Subject to:

$$\left\{ \sum_i (1-y_i)y'_i, \sum_i y_i(1-y'_i) \right\} \in \mathfrak{R}_k, \quad i = 1, \ldots, N, j \in \mathcal{N}_i \quad (23)$$
$$\mathfrak{y}_{ij}^{10} + \mathfrak{y}_{ij}^{11} = y'_i \quad (24)$$
$$\mathfrak{y}_{ij}^{01} + \mathfrak{y}_{ij}^{11} = y'_j \quad (25)$$
$$\mathfrak{y}_{ij}^{00} + \mathfrak{y}_{ij}^{01} + \mathfrak{y}_{ij}^{10} + \mathfrak{y}_{ij}^{11} = 1 \quad (26)$$

Solving this LP for thousands of binary variables (labels), is not computationally tractable. So instead we relax the label values to real numbers between zero and one and solve for optimal labeling. Later, we map the optimal labels to binary values by rounding the results. We solve Eq. 22 for each subregion separately, and return the labeling of the one with the maximum objective as the most violated constraint.

5. Experiments

As a concrete example, we experiment on object segmentation using our proposed approach. Given an input image, the goal is to produce a 0/1 mask, in

which a pixel gets label 1 if it is part of a given object category and label 0 otherwise.

Dataset. We run our experiments on the VOC2009 Segmentation [7] dataset. There are 749 images in the training set, 750 images in the validation set and 750 images in the test set. We train the parameters on the training set and evaluate performance on the validation set so that we can directly compare to baseline methods without relying on the VOC server. We compare the results using the intersection over union accuracy measure on 6 out of 20 object categories that can be localized the best employing our top-down features. Note that we perform the experiments on these objects independently. For example, when we segment object class car, any other object is taken as background. This is different from the VOC segmentation challenge in which the segmentation result should contain all object classes simultaneously. To combine our independent segmentations, we would need to have a score for each foreground pixel. Then, we could assign a pixel the label with maximum score. One way of scoring labels is the approach of Kohli [12] that can exactly compute the min marginals for graph cuts, however it is outside the scope of this paper.

Features. We define an MRF segmentation model with unary and pairwise features, for which the approximate inference is performed using FastPD [13]. Instead of working on the pixel level we first group the pixels into superpixels, which are fewer and therefore makes the learning process faster. Also they are larger so can be represented by more meaningful features. We use the superpixel extractor of Felzenszwalb et al. [14] that has three parameters. We set these parameters as $k = 200$, $MinArea = 1330$ and $\sigma = 0.01$. This setting of parameters result in an average of 50 superpixels per image of size 300×500 pixels.

To represent each superpixel, we use a set of bottom-up and top-down features, which form $\phi_u(\boldsymbol{x}_i)$ for superpixel i in Eq. 22. To create the bottom-up features, we compute Color SIFT features [15] on a dense grid with 6 pixel spacing in horizontal and vertical directions. We then turn this into a bag-of-words representation using a codebook of 1000 visual words.

For top-down features, we take a similar approach to the implicit shape model [16]. We first learn two appearance models for each of the 6 object categories using the detector of Felzenszwalb et al. [17]. The result includes two root filters and 6×2 part filters, where each root filter and 6 corresponding part filters model the object appearance in one pose. We run this detector on the training set and collect all bounding boxes that have positive scores. We then crop the ground-truth images on the bounding box locations and compute the average shape for the roots and parts, Fig 2.

We explain the rest of the process for one part, but the same process is applied to all parts and both roots. We find the potential part locations and their confidences by running the detector on the image in different scales. We call the result at each scale a confidence map, Fig. 3-b. Each potential part location casts its vote for the shape of that part proportional to its confidence. We implement this by convolving the confidence maps (different scales) with the average shape for that particular part. We call the convolution result in each

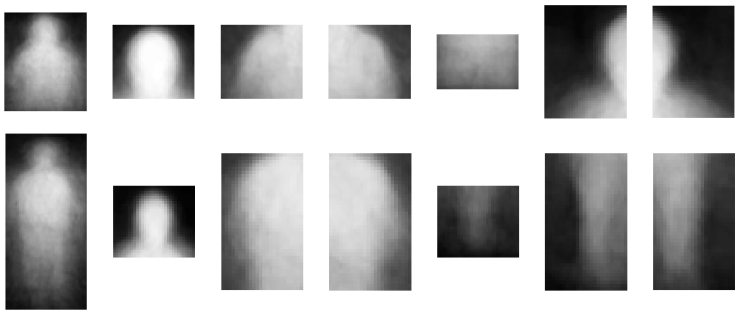


Fig. 2. Visualization of the average root and part shapes for person category. Each row corresponds to shape models obtained from root and part appearance models of one object pose.

scale a potential mask, Fig. 3-c. To merge the potential masks, we rescale them to the original image size and get the maximum of the masks, Fig. 3-d. We accumulate the mask values inside each superpixel to form the top-down feature corresponding to the part. Fig. 3 depicts the entire process for one part.

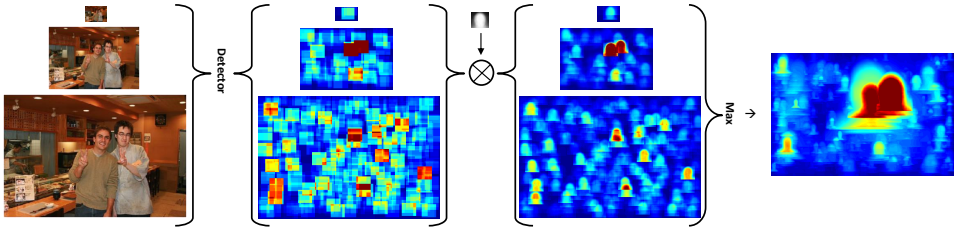


Fig. 3. The process of computing top-down features. Instead of showing the center of the detected parts we depict the bounding box for visualization purposes in the second stage.

To employ the pairwise interaction between neighboring superpixels i and j, we define a set of pairwise features that represent $\phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in Eq. 22. We first convert the image from RGB to La^*b^* color space. We define L_i, a_i and b_i to be the average L, a and b values inside superpixel i, respectively and assign the length of the common boundary between superpixel i and j to $\mathcal{P}_{ij}$. We then compute the pairwise features as

$$\boldsymbol{\phi}_p(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathcal{P}_{ij}. \exp\left[-\tau_1(L_i - L_j)^2, -\tau_2(a_i - a_j)^2, -\tau_2(b_i - b_j)^2\right]. \quad (27)$$

In our experiments the values of τ_1 and τ_2 are set to 2×10^{-2} and 5×10^{-3}, respectively.

Results We compare the proposed method to two other methods based on their intersection over union segmentation accuracy. We use the same set of features for all methods. All three methods share the same general framework

as explained by Tsochantaridis et al. [9]. The difference is in the form of their loss function Δ and their compatibility function Ψ. The first approach $BL1$ uses a decomposable Hamming loss function and a complex Ψ function including pairwise terms. The second method $BL2$ has been presented by Joachims [10] that can optimize a non-decomposable loss function, intersection over union in our experiment, but only includes unary terms in the Ψ function. And finally, the third method is the proposed approach that approximates the intersection over union loss function and can handle Ψ functions with unary and pairwise terms. We also show some segmentation results in Fig. 5 for all 6 object categories.

We use the same regularizer coefficient $C = 1$ for all three methods and set the number of subregions, M, for our piecewise linear approximation to 40. First, we triangulate the loss surface in FP, FN space finely. Then, we simplify the mesh into 40 triangles using a software called "Polygon Cruncher", which tries to approximate the original mesh as close as possible. To solve the LP problem of Eq. 22, we employ an off-the-shelf LP solver, Mosek [18].

In the training set, the number of superpixels that belong to the object are far fewer than the number of background superpixels, e.g., 1 foreground superpixel for every 25 background superpixels in person category. It means that reporting all superpixels as background gives Hamming score of $\frac{24}{25}$ or 96%. However, the same result obtains zero score based on intersection over union, because the intersection is simply empty. Therefore, we use *adjusted Hamming loss* defined as

$$\Delta_{AH} = \kappa FP + FN. \tag{28}$$

By changing κ we can adjust the relative contribution of foreground and background labels. In our experiment we set κ for each object to the ratio of foreground and background superpixels in the training set. Without this adjustment $BL1$ would always return every superpixel as background.

The results reported in Table 1 show significant improvement in segmentation accuracy by the proposed method. Moreover, the results of $BL1$ and $BL2$ are comparable in a sense that in half of the categories $BL1$ performs better than $BL2$ and performs worse in the other half.

Table 1. Intersection over union accuracies for 6 object categories.

	BL1	BL2	Proposed Method
	Δ = Adjusted Hamming	$\Delta = \frac{\cap}{\cup}$	$\Delta = \frac{\cap}{\cup}$
	Unary + Pairwise	Unary	Unary + Pairwise
person	20.73	26.7	32.53
bus	25.49	22.65	31.69
aeroplane	21.23	12.65	32.11
car	23.37	22.86	27.83
horse	0.0	5.2	13.85
tv/monitor	2.24	6.63	12.69

We compare the effect of optimizing adjusted Hamming loss versus intersection over union in Fig. 5. Adjusted Hamming loss tends to return fewer false positives, but with the cost of missing many true positives. In fact, it often marks all pixels as background, while intersection over union actually produces segmentations.

Fig. 4. Segmentation for person category. Optimizing adjusted Hamming loss (BL1) against our proposed method. a) input image, b) segmentation considering adjusted Hamming loss (BL1), c) our proposed method employing intersection over union. Intersection over union provides more true positives by possibly creating some false positives. Adjusted Hamming loss decreases false positive by sacrificing some true positives.

6 Conclusion

In this paper we develop a general algorithm for addressing learning problems with complex models and complex loss functions, those which are a function of false positive and false negative counts. We replace the original non-decomposable loss function with a piecewise linear approximation, and solve it using a linear programming relaxation of the original quadratic program. In future work it would be interesting to analyze the quality of these approximations.

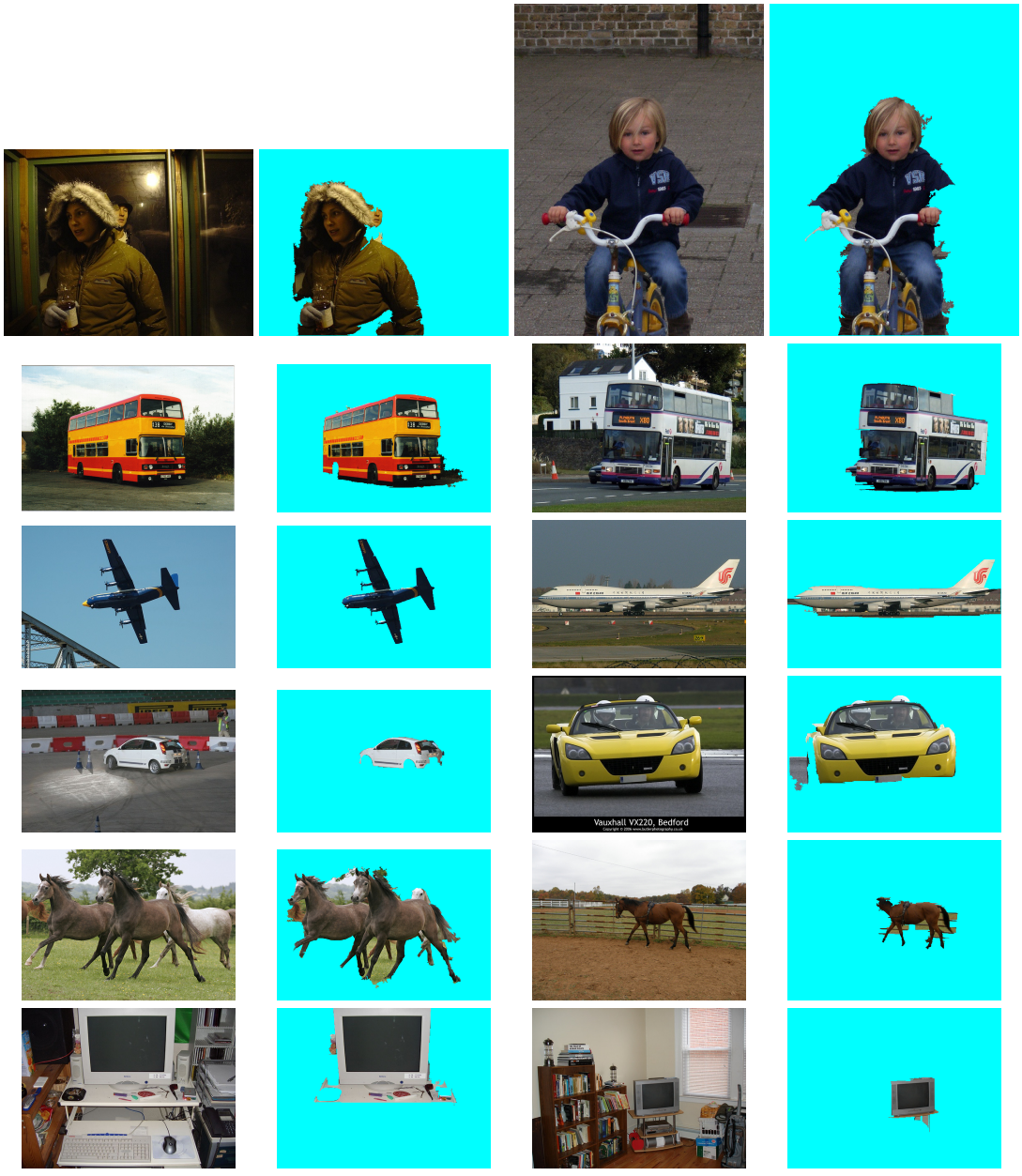


Fig. 5. Some segmentation results. Each row corresponds to one object category.

However, in this work we have provided experimental evidence of their effectiveness. In particular we apply this method to learning an image segmentation model that contains both unary terms for labeling pixels and pairwise terms on the labels of neighbouring pixels. We show that learning the parameters to this model under an objective directly tied to the performance measure significantly improves performance relative to baseline algorithms on the PASCAL VOC Segmentation Challenge.

References

1. Hoiem, D., Efros, A.A., Hebert, M.: Closing the loop in scene interpretation. In: Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn. (2008)
2. Blaschko, M.B., Lampert, C.H.: Learning to localize objects with structured output regression. In: ECCV. (2008)
3. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: ICCV. (2009)
4. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. Int. Journal of Computer Vision **43** (2001) 7–27
5. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. PAMI **23** (2001) 1222–1239
6. Szummer, M., Kohli, P., Hoiem, D.: Learning crfs using graph cuts. In: Proc. 10th Europ. Conf. Comput. Vision. (2008)
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html (2009)
8. Taskar, B., Chatalbashev, V., Koller, D., Guestrin, C.: Learning structured prediction models: a large margin approach. In: ICML '05. (2005) 896–903
9. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML. (2004)
10. Joachims, T.: A support vector method for multivariate performance measures. In: ICML '05, New York, NY, USA, ACM (2005) 377–384
11. Werner, T.: A linear programming approach to max-sum problem: A review. IEEE Trans. PAMI **29** (2007) 1165–1179
12. Kohli, P., Torr, P.H.S.: Measuring uncertainty in graph cut solutions. Comput. Vis. Image Underst. **112** (2008) 30–38
13. Komodakis, N., Tziritas, G.: Approximate labeling via graph-cuts based on linear programming. IEEE Trans. PAMI **29** (2007)
14. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Int. Journal of Computer Vision **59** (2004)
15. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluating color descriptors for object and scene recognition. IEEE Trans. PAMI (2010)
16. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: In ECCV workshop on statistical learning in computer vision. (2004) 17–32
17. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Trans. PAMI (2009)
18. Mosek: The mosek optimization software. http://www.mosek.com (2010)