Learning to Combine Mid-level Cues for Object Proposal Generation

Tom Lee Sanja Fidler Sven Dickinson
University of Toronto
{tshlee,fidler,sven}@cs.toronto.edu

Abstract

In recent years, region proposals have replaced sliding windows in support of object recognition, offering more discriminating shape and appearance information through improved localization. One powerful approach for generating region proposals is based on minimizing parametric energy functions with parametric maxflow. In this paper, we introduce Parametric Min-Loss (PML), a novel structured learning framework for parametric energy functions. While PML is generally applicable to different domains, we use it in the context of region proposals to learn to combine a set of mid-level grouping cues to yield a small set of object region proposals with high recall. Our learning framework accounts for multiple diverse outputs, and is complemented by diversification seeds based on image location and color. This approach casts perceptual grouping and cue combination in a novel structured learning framework which yields baseline improvements on VOC 2012 and COCO 2014.

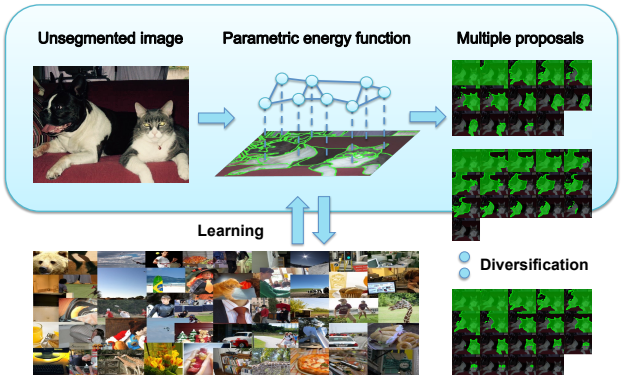

Figure 1. Our approach takes an input image, partitions it into superpixels, and groups superpixels into region proposals using a novel structured learning framework for parametric energy functions, called Parametric Min-Loss (PML). The parametric energy function combines mid-level cues with weights that are trained to generate multiple region proposals. Finally, we diversify the energy function to generate a diverse set of region proposals.

1. Introduction

For many years, the recognition community focused on the problem of object detection, in which a strong object prior was "tested" at all possible locations using the brute-force approach of sliding windows. Bottom-up segmentation, *e.g.* [26, 9, 31], was clearly unnecessary in the presence of a strong object prior, and while the complexity of this framework grew linearly with the number of detectors, parallel processing allowed a significant number of classes to be detected before a linear search became intractable. At that point, the concept of an "objectness" detector was introduced [2], resurrecting interest in bottom-up saliency and attention. By testing which of the window locations contained salient object information, the linear search of detectors could be restricted to a small subset of windows. While an objectness detector could arguably be considered a weak form of perceptual grouping (with the grouping provided by the sliding window), bottom-up segmentation still remained in the shadows.

Only when the number of categories grew to thousands or more did the community advance the need for more discriminative features such as shape and appearance which, in turn, require bottom-up region segmentation. The extraction of such "region proposals", *e.g.* [29, 5], meant that brute-force sliding window searches, numbering in the tens or hundreds of thousands, could be replaced by the extraction of hundreds or thousands of region proposals. Each proposal offers (boundary) shape, appearance, and scale information which, for a correct proposal, can be exploited to recognize an object (or part of an object) or select a small number of object detectors that can be applied to the region. As long as the region proposals exhibit good recall, there's a chance that the object(s) will be recognized.

The return to bottom-up region segmentation is an invitation to integrate the many mid-level cues that ultimately play a role in such perceptual grouping, including proximity, symmetry closure, similarity, and continuity, to name a few. But even with computational models of such cues, how should they be combined and what is their relative importance? As shown in Figure 1, we explore these issues within

the framework of graphical models, which encode contextual relationships such as grouping cues between adjacent image regions. Computationally, graphical models can be solved exactly in a tractable manner, *e.g*. by minimizing a pairwise submodular energy function of binary variables with a maxflow algorithm. They can be discriminatively trained to predict structured outputs, offering a consistent learning and inference framework for bottom-up segmentation [24, 27].

Until recently, discriminative graphical models for segmentation have been restricted to single-output predictions, and lacked a framework for learning to predict diverse multiple outputs, *e.g*. as introduced in Multiple Choice Learning (MCL) [13]. Multiple-output models are especially important for region proposals due to the principle of least commitment needed for bottom-up grouping. One such tool that has emerged in vision is parametric maxflow [17], which is used to minimize an energy function $E^{\lambda}(y)$ for multiple values of parameter λ, generating multiple solutions at a time. Parametric maxflow was applied to proposing object regions in CPMC [5] and in subsequent variants [15, 23].

Despite frequent use in region proposal generation, however, parametric energy functions are, as a rule, not trained to predict multiple outputs, but rather trained to predict a single output. To the best of our knowledge, this paper is the first to bridge the gap between learning and inference for parametric maxflow. Our formulation is inspired by MCL, which models multiple-output learning with a loss function that evaluates multiple outputs against a correct output. Our model, however, differs from MCL in 1) having a single parametric energy function, and in 2) automatically adapting the number of output solutions to the input image. Despite these significant differences, we find that MCL's block-coordinate descent strategy applies to parametric maxflow and yields a solution that decomposes into simple alternating steps.

In summary, we introduce Parametric Min-Loss (PML), a novel model and algorithm for structured multiple-output learning using parametric maxflow. We demonstrate its use for learning to combine a set of mid-level grouping cues to yield a set of region proposals with high recall. Besides having applications to perceptual grouping, the model bridges the disparity between learning and inference for parametric energy functions and can be applied to any domain that uses parametric maxflow. While learning accounts for diverse multiple outputs, we include a complementary diversification step that allows the proposals to adapt to different conditions. With a large-scale experimental validation, we cast mid-level cue combination in a structured learning environment, representing an exciting new direction for perceptual grouping.

2. Related work

Perceptual grouping has often been formulated as an energy minimization problem, *e.g*. [12, 30, 32, 7, 34, 16, 26], yielding a single region or (possibly) closed contour, or a partition into regions. In the more recent context of generating region proposals, a *parametric* energy minimization problem is often formulated (*e.g*. CPMC [5]) in which the energy is parameterized by λ and minimized for multiple values of λ using parametric maxflow, yielding multiple solutions. Such an approach is an extension of energy minimization from predicting a single output to predicting multiple outputs in support of the principle of least commitment, and has been refined by subsequent variants [15, 23]. However, the combination of cues is typically specified manually in the energy or not trained jointly in the energy.

Moreover, a gap has emerged between learning and inference for parametric maxflow because prediction has been extended to multiple outputs while learning has not. This disparity exists in general for multiple-output models, an example of which is the M-Best MAP approach for generating multiple hypotheses [10]. Recently, Multiple Choice Learning (MCL) [13] addressed this gap in a tractable way using an M-tuple of independent structured predictors that predicts M outputs. The model is efficient and minimizes the loss of only the most accurate prediction in the set of outputs. Subsequent improvements included an explicit criterion to encourage diversity among the predictors [14], however the model remains fundamentally different from parametric maxflow, which solves a single parametric energy function that accounts for multiple outputs, and whose number of outputs is adaptive and does not need to be prespecified. Our method for parametric maxflow, however, is similar to MCL in using a block-coordinate descent strategy in a large-margin formulation to close the train-test gap.

Approaches for region proposals typically consist of a generation stage for hypothesizing proposals, followed by a ranking stage that attempts to order them by "objectness". A diversity of approaches exist in which many generate proposals in the form of bounding boxes, *e.g*. Objectness [2] and Edge Boxes [35]. In such methods, a sliding window suffices as no explicit grouping is required, and they are suitable for box-based detectors even though proposals do not explicitly capture the underlying shape of the objects. Selective Search [29] efficiently generates region-based proposals based on greedily merging superpixels and was subsequently improved with trained affinity functions [33]. The approach is similar to ours in using region-based similarity cues, however the agglomerative grouping procedure is brute force.

In approaches for region-based proposals, such as GOP [18], RIGOR [15], MCG [4], the principle of least commitment is typically not built into learning. Only very recently was such a method proposed [19] that minimized the

loss of the most accurate region proposal, with efficient runtime at test time and achieving competitive results. In our work, we minimize the same loss function, however one of our key aims is to develop a graphical model that is unified across learning and inference. Another recent work [6] also uses learning to combine several cues for generating object proposals in 3D, but it does not use parametric energies. Earlier methods gave a significant role to learning in the ranking stage, *e.g.* [4, 8, 23]. CPMC [5] uses parametric maxflow to generate proposals and is most similar to ours in spirit, however we perform grouping at superpixel-level rather than pixel-level. This allows access to region-based mid-level cues during the generation stage. In contrast to the above methods, our approach emphasizes the generation stage over the ranking stage, and emphasizes the role of learning to group using mid-level cues. The closest methods to our approach are Superpixel Closure [22], which uses mid-level closure, but does not combine other cues, and Multicue [21], which combines mid-level cues in a parametric energy function, but only trains the energy to generate a single proposal.

3. Perceptual grouping cues

Our method begins by segmenting the input image x into a single layer of superpixels that forms the basis of feature extraction, labeling, and grouping. Superpixels reduce search complexity while providing access to local region and contour scope. At the same time, we are restricting regions to superpixel boundaries, so it is important to preserve boundary recall. The resulting strategy is to oversegment the image into superpixels which remain to be grouped.

Formally, we partition the image x into a set S of superpixels, from which we seek a subset $R \subset S$ that represents an object. Equivalently, we represent R as a binary labeling $\mathbf{y} \in \{0,1\}^{|S|}$, where $y_p = 1$ exactly when superpixel p is in R, for $p = 1, \ldots, |S|$, hence $R = \{p : y_p = 1\}$. The space of possible regions lies in $\mathcal{Y} = \{0,1\}^{|S|}$.

Given an image x, we seek a minimum energy region $\mathbf{y} \in \mathcal{Y}$ with respect to the energy $E^\lambda(x) : \mathcal{Y} \to \mathbb{R}$ which is defined for the image and a parameter λ. Specifically, we minimize the energy function:

$$E^\lambda(x, \mathbf{y}) = \lambda \sum_p \phi_0(x, y_p) + \mathbf{w}_1^\mathsf{T} \sum_p \phi_1(x, y_p) + \mathbf{w}_2^\mathsf{T} \sum_{p,q} \phi_2(x, \mathbf{y}_{p,q}), \quad (1)$$

whose terms here are grouped by weighted features ϕ_0, ϕ_1, ϕ_2. This energy can be minimized for multiple values of λ by parametric maxflow under further constraints (see [17]), however the goal of this section is to model mid-level grouping cues in the energy. To do so, we regroup the energy (1) into subenergies that model their respective cues:

$$E^\lambda(x, \mathbf{y}) = E_{\text{app}}(x, \mathbf{y}) + E_{\text{clo}}(x, \mathbf{y}) + E_{\text{sym}}(x, \mathbf{y}) + E^\lambda_{\text{scale}}(x, \mathbf{y}). \quad (2)$$

The following sections will define the subenergies above.

3.1. Proximity

The grouping cue of proximity is a basic image relation that is preserved through image projection. Since pairwise potentials encode grouping relations, proximity is reflected in placing a potential on every pair of *adjacent* superpixels, thereby defining the edge set $\mathcal{A}(S) \subset S^2$.

3.2. Appearance similarity

Appearance similarity is a non-accidental regularity of objects—the more similar a group of elements are to each other, the more likely they belong to the same object. We extract a color histogram $\mathbf{h}_p^{\text{col}}$ of d^{col} dimensions for every superpixel p, and define a similarity for every pair $(p, q) \in \mathcal{A}(S)$ using the histogram intersection kernel [29]:

$$\text{sim}_{p,q}(\mathbf{h}) = \sum_{i=1}^{d} \min(\mathbf{h}_p(i), \mathbf{h}_q(i)) \quad (3)$$

We similarly define similarity for a texture histogram $\mathbf{h}_p^{\text{text}}$ of d^{text} dimensions. Appearance similarity is encoded into our energy as a 2-dimensional feature consisting of color and texture:

$$\phi_{\text{app}}(x, \mathbf{y}_{p,q}) = \mathbb{1}_{[y_p \neq y_q]}(\text{sim}_{p,q}(\mathbf{h}^{\text{col}}), \text{sim}_{p,q}(\mathbf{h}^{\text{text}})) \quad (4)$$

We note that ϕ_{app} contributes a cost only when neighboring superpixels with strong similarity are labeled differently. The potentials are weighted by $\mathbf{w}_{\text{app}}$ which is trained and shared across all superpixel pairs, and overall contributes to the following energy:

$$E_{\text{app}}(x, \mathbf{y}) = \mathbf{w}_{\text{app}}^\mathsf{T} \sum_{p,q} \phi_{\text{app}}(x, \mathbf{y}_{p,q}). \quad (5)$$

3.3. Contour closure

Contour closure is a non-accidental regularity of objects, in which object coherence in 3D projects to a closed boundary in 2D. The more contour evidence there is along the boundary of a given region $\mathbf{y}$, the more likely it is to enclose an object. We use a cost function that sums contour gap $G(x, \mathbf{y}) = \sum_{b \in \partial(\mathbf{y})} g(x, b)$ along the region boundary $\partial(\mathbf{y})$, where $g(x, b)$ is the gap (lack of contour) evaluated at pixel b of image x.

To express the cost $G(x, \mathbf{y})$ in the form of unary and pairwise features [22], we first define a unary feature:

$$\phi_{\text{clo}}(x, y_p) = \sum_{b \in \partial(p)} \mathbb{1}_{[y_p = 1]} g(x, b) \quad (6)$$

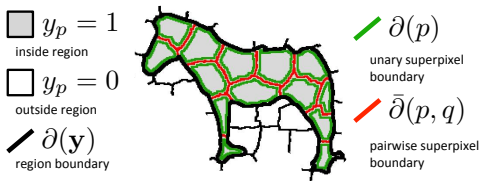

Figure 2. Given a region defined by $\mathbf{y} \in \{0,1\}^{|S|}$, the closure cue sums gap along its boundary $\partial(\mathbf{y})$. Summation is regrouped into unary superpixel boundaries $\partial(p)$ and pairwise superpixel boundaries $\bar{\partial}(p,q)$ for superpixels inside the region (see text for details).

that sums gap along selected superpixel boundaries. For a region consisting of a single superpixel, the unary feature sums the correct gap cost. However, as shown in Figure 2, for a region consisting of multiple superpixels, simply summing the unary features will double count the gaps along the internal boundaries shared by adjacent superpixels. We thus define pairwise features to cancel them out:

$$\phi_{\text{clo}}(x, \mathbf{y}_{p,q}) = \sum_{b \in \bar{\partial}(p,q)} \mathbb{1}_{[y_p = y_q = 1]} g(x, b) \tag{7}$$

The gap $G(x, \mathbf{y})$ of region $\mathbf{y}$ is thus the sum of the unary features, minus twice the pairwise features. In summary, the closure cue contributes the weighted energy:

$$E_{\text{clo}}(x, \mathbf{y}) = w_{\text{clo}}^{\mathsf{T}} \left(\sum_p \phi_{\text{clo}}(x, y_p) - 2 \sum_{p,q} \phi_{\text{clo}}(x, \mathbf{y}_{p,q}) \right) \tag{8}$$

3.4. Symmetry

Symmetry is a powerful regularity in objects. While symmetry captures interactions among all parts of an object, this must be balanced with the need for a low-order energy. Coarse superpixels help by expanding the spatial scope of each unit, however superpixel size must also be limited in order to preserve boundary recall. Overall, it is a computational challenge to capture grouping by symmetry.

We follow the approach of [21] of "outsourcing" symmetry to a region-based symmetry detector [20], and biasing our energy to detected symmetric parts. Formally, given a set T of region-scoped, scored symmetric parts, we define pairwise potentials that prefer to merge superpixels when they fall in the same symmetric part. For every pair $(p,q) \in \mathcal{A}(S)$ we define:

$$\phi_{\text{sym}}(x, \mathbf{y}_{p,q}) = \mathbb{1}_{[y_p \neq y_q]} \max_{s \in S(p,q)} \text{score}(s), \tag{9}$$

where the max considers symmetric parts $T(p,q) \subseteq T$ that overlap p and q by at least $\tau = 0.75$, and selects the best-scoring one. A value of zero is assigned when $T(p,q)$ is empty. Non-maximum suppression is applied over all superpixel pairs so that at most one symmetric part contributes to each pair. Overall, symmetry contributes the weighted energy:

$$E_{\text{sym}}(x, \mathbf{y}) = w_{\text{sym}}^{\mathsf{T}} \sum_{p,q} \phi_{\text{sym}}(x, \mathbf{y}_{p,q}). \tag{10}$$

3.5. Object scale

The grouping energies above accumulate higher costs for regions with more superpixels, and thus the energy is artificially biased toward smaller regions and needs to be normalized by the region's size. To do so, we subtract unary features ϕ_{area} scaled by a factor $|\lambda|$ from the energy, with the effect of accommodating larger regions as $|\lambda|$ increases. Practically, a non-zero λ is necessary to remove trivial solutions. We define $\phi_{\text{area}}(x, y_p) = \mathbb{1}_{[y_p=1]}\text{area}(p)$, which contributes the negative quantity:

$$E^{\lambda}_{\text{scale}}(x, \mathbf{y}) = \lambda \sum_p \phi_{\text{area}}(x, y_p). \tag{11}$$

A diverse set of solutions can be obtained with different values of λ. Note that the cost of selecting an individual superpixel is influenced by the magnitude of λ against other potentials: a very large $|\lambda|$ will more than offset the other potentials and cause all superpixels to be selected. Since potentials are empirically below 1, we can obtain all solutions by varying λ within $[-1, 0]$.

4. Parametric energy minimization

The domain $\mathcal{Y}$ over which the energy (1) is minimized is too large for exhaustive search, but when written as a sum of unary and pairwise potentials, the energy is seen to have the required structure for an efficient solution. When submodular pairwise potentials are guaranteed by requiring $\mathbf{w} \geq 0$, and λ is held at a fixed non-positive value, the problem

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}} E^{\lambda}(x, \mathbf{y}, \mathbf{w}) \tag{12}$$

can be solved exactly by a maxflow algorithm. Solving (12) for all values $\lambda \in [\lambda^{\min}, \lambda^{\max}]$ simultaneously is known as a parametric problem, and can be done via parametric maxflow. Furthermore, since the linear term ϕ_0 measures area, the monotonicity property is satisfied that guarantees a solution size of size linear in $|S|$ [17]. See Figure 3 for a visualization of the solution set in an input image.

We rewrite (1) in a linear form that is amenable to large-margin learning [28] by stacking the features and weights of individual cues together. Specifically, we define a weight vector $\mathbf{w} = (\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2)$ where $\mathbf{w}_0 = 1$, and a feature vector $\phi^{\lambda} = (-\lambda\Phi_0, -\Phi_1, -\Phi_2)$. We can then rewrite $E^{\lambda}(x, \mathbf{y}, \mathbf{w}) = -\mathbf{w}^{\mathsf{T}}\phi^{\lambda}(x, \mathbf{y})$ and thus rewrite (12) as:

$$\hat{\mathbf{y}}(x, \mathbf{w}) = \arg\max_{\mathbf{y}} \mathbf{w}^{\mathsf{T}}\phi^{\lambda}(x, \mathbf{y}). \tag{13}$$

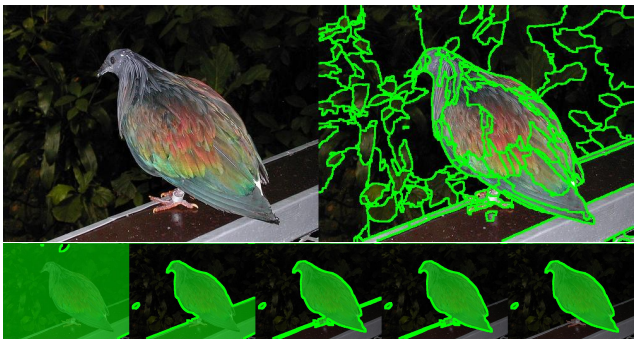

Figure 3. Given the input image and superpixel segmentation shown in the first row, our approach defines a parametric problem whose solution set is shown in the second row. Optimal labelings are listed in order of increasing $\lambda \in [-1, 0]$.

Finally, the structured prediction function (13) is generalized to a set of solutions over a range of λ:

$$\hat{Y}(x, \mathbf{w}) = \{\hat{\mathbf{y}}^{\lambda}(x, \mathbf{w}) : \lambda \in [-1, 0]\}. \tag{14}$$

5. Parametric Min-Loss learning

When a ground truth region g annotates an object in input image x, the quality of the set $\hat{Y}(x, \mathbf{w})$ of predicted regions can be evaluated against g. In the evaluation of region proposals, for example, Jaccard similarity is considered by the Average Best Overlap (ABO) metric [29]. In S-SVM learning [28], a task loss $\ell(\hat{\mathbf{y}}, \mathbf{y})$ measures the mismatch of a structured prediction $\hat{\mathbf{y}}$ against $\mathbf{y}$. To measure the mismatch of a set $\hat{Y}$ of structured predictions, however, we generalize the task loss to a set in the following way:

$$\mathcal{L}(\hat{Y}, \mathbf{y}) = \min_{\hat{\mathbf{y}} \in \hat{Y}} \ell(\hat{\mathbf{y}}, \mathbf{y}), \tag{15}$$

where (15) says that the quality of the entire set $\hat{Y}$ of predictions is the quality of the best prediction. As in standard S-SVM, the task loss ℓ is defined to be amenable to loss-augmented inference [28] and decomposes into a sum of unary losses. Each unary loss uses v_p, as defined below, to measure the mismatch of superpixel p against the ground truth region g as follows:

$$\ell(\hat{\mathbf{y}}, \mathbf{y}(g)) = \frac{1}{|g|} \sum_p |p| \begin{cases} v_p & \hat{y}_p = 0 \\ 1 - v_p & \hat{y}_p = 1, \end{cases} \tag{16}$$

where v_p is the fraction of p's pixels that lie in g.

The weights of (1) are ideally learned by minimizing $\mathcal{L}(\hat{Y}, y)$ in $\mathbf{w}$, but in order to circumvent difficulties arising from non-convexity and discontinuities, we develop a related loss function $H(\mathbf{w})$ that is easier to minimize. Our derivation of $H(\mathbf{w})$ follows a strategy based on the (structured) hinge loss: as the hinge loss is an upper bound of the task loss, we derive a min hinge loss that is an upper bound of the min task loss [13]. We first write the hinge loss for parametric maxflow as follows, with dependence on the training example $(x, \mathbf{y})$ omitted for brevity:

$$h(\mathbf{w}, \lambda) = \max_{\hat{\mathbf{y}}} \ell(\hat{\mathbf{y}}, \mathbf{y}) + \mathbf{w}^{\mathsf{T}} \phi^{\lambda}(x, \hat{\mathbf{y}}) - \mathbf{w}^{\mathsf{T}} \phi^{\lambda}(x, \mathbf{y}). \tag{17}$$

The min-hinge $H(\mathbf{w})$ then takes the minimum of $h(\mathbf{w}, \lambda)$ over a range of λ:

$$H(\mathbf{w}) = \min_{\lambda \in [-1,0]} h(\mathbf{w}, \lambda). \tag{18}$$

Unlike in standard S-SVM, the loss function $H(\mathbf{w})$ is not guaranteed to be convex, however it is shown that H is an upper bound for h [13].

Accounting for all ground truth regions, we obtain the regularized min-hinge minimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{N} \sum_{n=1}^{N} \min_{\lambda_n \in [-1,0]} h_n(\mathbf{w}, \lambda_n). \tag{19}$$

Although solving (19) is an NP-hard problem, we can derive an efficient solution by rewriting the problem as:

$$\min_{\mathbf{w}} \min_{\{\lambda_n\}} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{N} \sum_{n=1}^{N} h_n(\mathbf{w}, \lambda_n) \tag{20}$$

Algorithm 1 Parametric Min-Loss

Require: $\{\phi^{\lambda}(x, \cdot), \mathbf{y}, \ell\}_{n=1}^{N}$
Ensure: $\mathbf{w}^* \geq 0$

$\tau \leftarrow 0$
repeat
  $\tau \leftarrow \tau + 1$
  $\mathbf{w}^{(\tau)} \leftarrow \text{SSVM}(\{\phi^{\lambda^{(\tau)}}(x, \cdot), \mathbf{y}, \ell\}_{n=1}^{N})$
  for $n \leftarrow 1 \ldots N$ **do**
    $\{(\lambda_i, \mathbf{y}_i)\} \leftarrow \text{PMF}(-\ell(\mathbf{y}_n) - \mathbf{w}^{\mathsf{T}} \phi(\mathbf{x}_n), [-1, 0])$
    $h_i \leftarrow \ell(\mathbf{y}_n, \mathbf{y}_i) + \mathbf{w}^{\mathsf{T}}[\phi^{\lambda_i}(\mathbf{x}_n, \mathbf{y}_i) - \phi^{\lambda_i}(\mathbf{x}_n, \mathbf{y}_n)], \forall i$
    $\lambda_n^{(\tau)} \leftarrow \lambda_{\arg\min h_i}$
  end for
until converged or maxed out
return $\mathbf{w}^* \leftarrow \mathbf{w}^{(\tau)}$

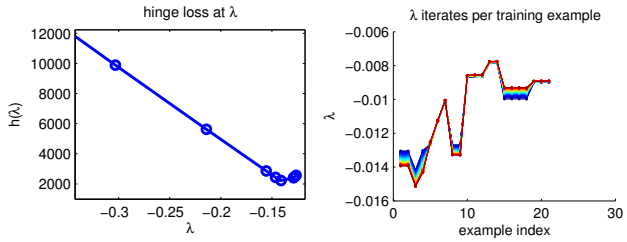

Figure 4. The convex function $h(\lambda)$ sampled at breakpoints (left), and evolution of $\lambda_n^{(\tau)}$'s over time τ (right). Warmer colors represent later iterations.

and decomposing into two simpler problems that we identify as λ-update and standard S-SVM. Our algorithm, summarized in Algorithm 1, alternates between holding $\mathbf{w}$ fixed and optimizing the λ's, and holding the λ's fixed and solving S-SVM.

Fixing $\mathbf{w}$, we obtain N independent problems that can be solved in parallel. Each problem amounts to solving parametric minimization of the hinge-loss:

$$\arg \min_{\lambda \in [-1,0]} h(\mathbf{w}, \lambda) \tag{21}$$

Since the function $h(\mathbf{w}, \lambda)$ is the maximum of $2^{|S|}$ linear functions, it is convex and piecewise-linear. It follows that $h(\mathbf{w}, \lambda)$ reaches its minimum value at one of the λ breakpoints, and so we need only to search for the breakpoint that evaluates to a minimum. The set of breakpoints $\{\lambda_i\}$ and their solutions $\{\mathbf{y}_i\}$ are found by solving the parametric maxflow problem:

$$\forall \lambda \in [-1, 0], \min_{\hat{\mathbf{y}}} -\ell(\hat{\mathbf{y}}, \mathbf{y}) - \mathbf{w}^{\mathsf{T}} \phi^{\lambda}(x, \hat{\mathbf{y}}) \tag{22}$$

To solve (21), we exhaustively evaluate $h(\mathbf{w}, \lambda)$ for each λ_i using $\mathbf{y}_i$. We note that λ has monotonic coefficients and thus there are at most $O(|S|)$ breakpoints [17] containing the solution. See Figure 4 for an illustration.

With $\{\lambda_n\}$ fixed, problem (19) reduces to a single, standard S-SVM problem:

$$\mathbf{w} = \arg \min_{\mathbf{w}} \frac{1}{2} ||\mathbf{w}||^2 + \frac{C}{N} \sum_{n=1}^{N} h_n(\mathbf{w}, \lambda_n) \tag{23}$$

We solve (23) with the constraint $\mathbf{w} \geq 0$ using the cutting-plane implementation of [25].

Although the learning algorithm alternates between minimizing $\mathbf{w}$ and λ, the learning goal for region proposals is to optimize the weights. Minimization in λ reflects the selection of the best region by a ground truth oracle, and prediction has no access to such an oracle. Moreover, in the absence of a specified object category, bottom-up grouping cues are the only means of predicting region proposals.

6. Diversification

Diversification is an important step toward achieving recall without a specified object category. In a given image, we sample seeds that make assumptions about object properties such as image location and color distribution. An energy function that is biased with a particular seed will then yield proposals that are customized toward a particular location or color distribution. Pooling together proposals associated with different seeds allows us to cover a wide range of conditions with greater precision.

Location. We use individual superpixels to seed image locations, with a total of $|S|$ seeds. As shown in Figure 5, each seed p defines unary features that discourage selecting $y_q = 1$ depending on q's "distance" from p. This is encoded for any superpixel q with a cost based on the maximum distance of q from p. For non-compact superpixels, this promotes compactness by encouraging smaller, nearby regions to be "annexed" first (regardless of their color).

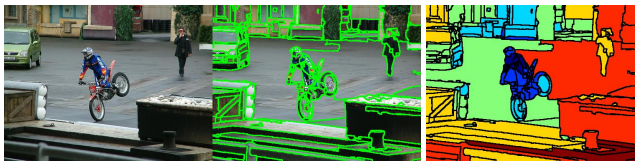

Figure 5. A location-based seed is sampled on the motorcyclist's back, which induces the unary potentials as shown. Warmer colors represent higher costs.

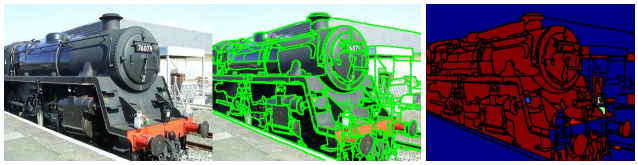

Figure 6. A color-based seed is sampled on the pair of foreground-background color distributions, which induces the unary potentials as shown.

Color similarity. Here, we seed image colors (without considering proximity) using a Gaussian mixture model applied to the color space. Specifically, we seed foreground-background pairs of color distributions. For any given seed, each superpixel q has a likelihood under the foreground distribution and a likelihood under the background distribution. The higher the likelihood ratio, the lower the cost of assigning $y_q = 1$. An example is provided in Figure 6.

In our application, diversification is complementary to PML. While learning accounts for diversity over scale in λ, here the energy is further diversified in location and color. Moreover, learning and diversification balance each other out, as the grouping cues combined in the energy have a tempering effect on diversification seeds, *e.g.* by helping a seed centered on a location to adapt to irregular shapes.

7. Postprocessing

All solutions pooled over diversification seeds enter a pipeline of postprocessing steps. First, we process each solution $\mathbf{y} \in \mathcal{Y}$ to ensure that contiguous regions are considered for recall. We find connected components efficiently in superpixel space, and include the top $M = 2$ connected components as region proposals.

We then remove artefactual regions in the form of empty labelings and labelings that are within a very high percentage of the image's total area. We filter out redundant regions in the form of duplicate labelings and clusters of labelings

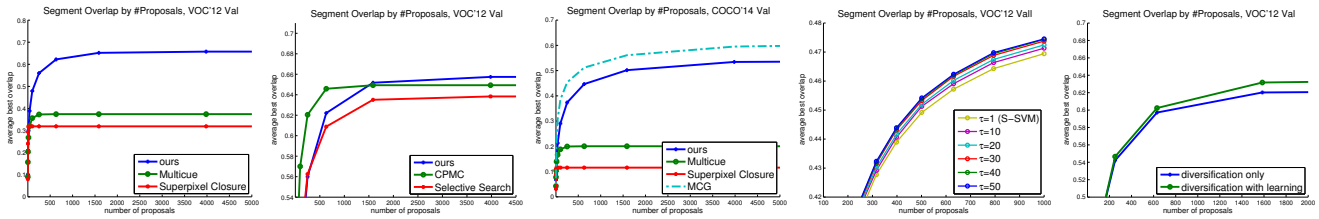

Figure 7. Comparison with perceptual grouping methods Multicue [21] and Superpixel Closure [22] (far left), and CPMC [5] and Selective Search [29] (left) on VOC'12. Comparison with MCG [4] and perceptual grouping methods on COCO'14 (middle). See text for comparisons with more recent methods like GOP [18] and RIGOR [15]. Parametric Min-Loss learning improves segment overlap as weights evolve (right). Improvement from smoothing the diversification seeds with trained mid-level cues (far right).

that are similar in overlap. Similarly to [5], we perform agglomerative clustering of labelings by intersection-over-union overlap, and consider clusters of labelings that exceed a very high overlap threshold. For each cluster, we keep the labeling with the best closure and discard the rest. Closure is efficiently computed using the gap cost $G(x, \mathbf{y})$.

Finally, we rank the proposals to allow a small number of proposals to be selected. We cast this as a problem of assigning a classification score to each region that indicates how object-like it is. Unlike the perceptual grouping problem above, this is a verification step in which higher-order relations are more easily captured over the full region scope. We turn to convolutional neural networks as they yield good categorization results. The final network layers are fully connected, and can be thought of as learned, mid-level features that encode category-independent information that is relevant for categorization. Specifically, to extract a feature vector for a given region proposal R, we place a cropping box tightly around R, and warp the cropped image to normalized dimensions. After normalizing pixel values, we evaluate OxfordNet and retain layer 20 as a 4096-dimensional feature vector. Like R-CNN [11], we then trained a SVM classifier on the feature to assign 'object' or 'non-object' to each R, and trained a logistic regressor to map the output margin to a score between 0 and 1.

We obtained features for positive and negative training examples by sampling from the training images of the VOC 2012 SEGMENTATION subset. For each image, we use the ground truth boxes as positive examples, and a matching number of random boxes as negative examples.

8. Results

For quantitative evaluation, the SEGMENTATION subset of VOC 2012 provides a set of images containing different objects annotated with at least one ground truth region per image. We apply Parametric Min-Loss on the TRAIN subset and evaluate our trained method on the VAL subset. We use $\mathcal{P}$ to denote the set of proposed regions to be evaluated, and $\mathcal{G}$ to denote the corresponding set of ground truth regions. For all pairs $(p \in \mathcal{P}, g \in \mathcal{G})$ contained by the same image, we consider the Jaccard similarity $\mathcal{J}(p, g) = \frac{|p \cap g|}{|p \cup g|}$ to score the quality of a potential match. The Average Best Overlap (ABO) metric [29] is defined over all images as follows:

$$\text{ABO}(k) = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \max_{p \in \mathcal{P}(k)} \mathcal{J}(p, g),$$

where $\mathcal{P}(k)$ represents the top k regions proposed for g's image. Plots sample ABO on increasing values of k to show the trade-off between recall and the number of proposals.

Our method requires a single superpixel layer as preprocessing. We found that non-compact superpixels, *e.g.* Felzenszwalb & Huttenlocher [9], yielded better results than compact superpixels, *e.g.* SLIC [1]. Results in this paper were generated using UCM [3], thresholded at $k = 0.1$.

Overall results. We first compare our method with two recent perceptual grouping methods most similar to ours, in Figure 7 (far left). Superpixel Closure [22] used parametric maxflow to group superpixels into regions of minimum closure cost, while Multicue [21] used S-SVM to train a parametric energy function that combines appearance, closure, and symmetry cues. We improve on these methods via a holistic learning framework and effective diversification. While both methods were quantitatively evaluated only on the Weizmann Horse Database, we evaluate on VOC 2012 VAL segmentations and COCO 2014 VAL segmentations.

Our results are comparable with leading region proposal methods such as Selective Search [29] and CPMC [5] on VOC 2012 VAL, as shown in Figure 7 (left). At 1585 proposals, GOP [18], RIGOR [15], MCG [4], and CMPC achieve 75.1, 74.4, 69.8, and 64.9 ABO, respectively, while ours achieves 65.2 ABO, so we are outperformed by the most recent methods. Ours takes a similar approach to Selective Search in using regional features to group superpixels, however we train a combination of cues and find minimum energy regions, allowing "better focused" proposals. Our level of recall, however, is more comparable to that of CPMC's for higher numbers of proposals. While we achieve higher recall with learning and effective diversification, our simple ranking procedure with a SVM classifier does less for precision than the sophisticated overlap regressors of CPMC.

In Figure 8, we show some example region proposals.

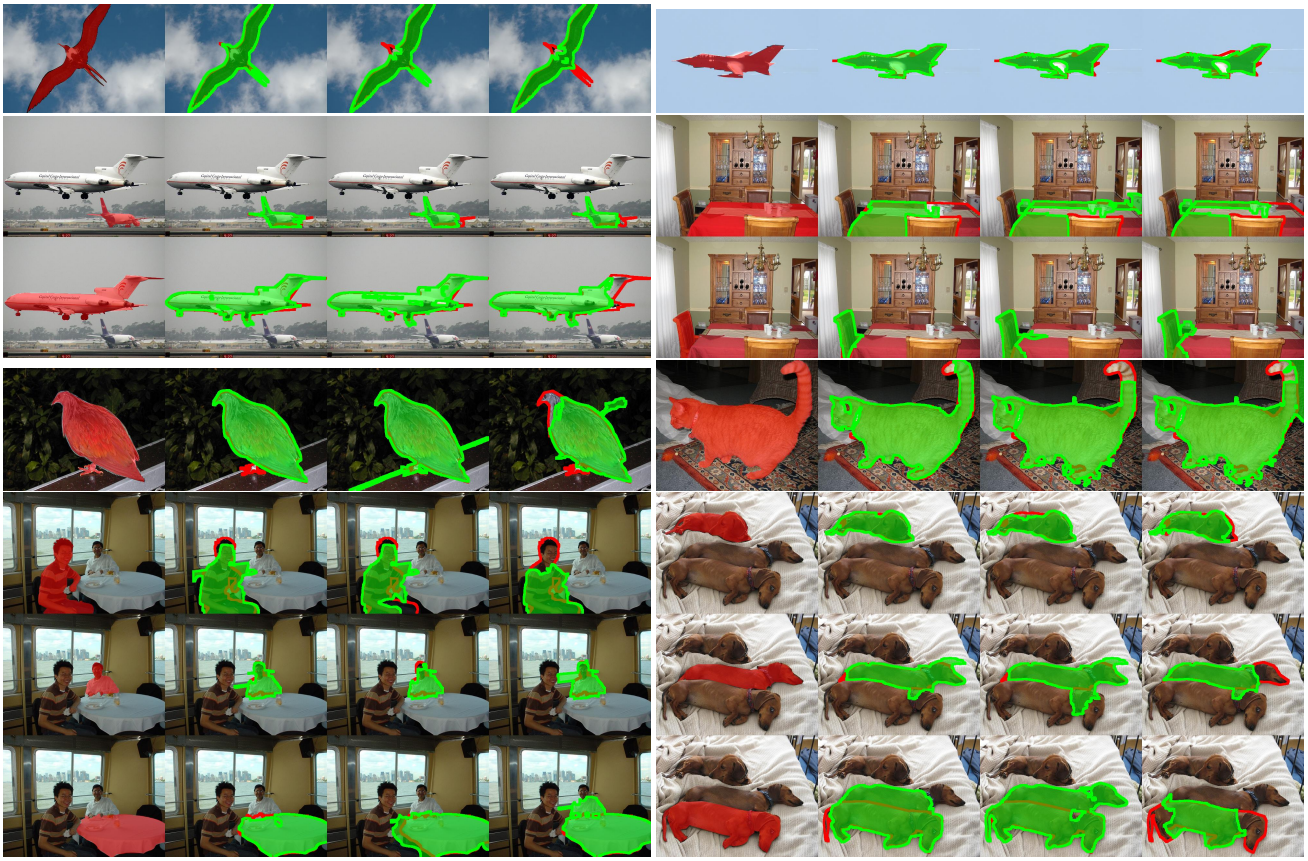

Figure 8. Example region proposals found for images from VOC 2012. Red masks denote ground truth, green masks denote the corresponding top proposals (from left to right).

For the images of the airplane, bird, and cat, our approach does well in separating figure from ground. For images of more complex scenes such as the dinner table, over- and undersegmentation occurred due to low contrast or in objects of highly heterogeneous appearance.

Learning. The second part of our results focuses on learning. We note that S-SVM is a natural baseline for Parametric Min-Loss due to the structure of the iterative algorithm. Specifically, we track the energy functions corresponding to weights as they evolve over iterations (indexed by τ), where the first iteration corresponds to S-SVM with initial λ values. We initialize λ's to -0.01, as done in Multicue [21]. As shown in Figure 7 (right), successive energy functions yield better recall, iteratively improving on the S-SVM baseline. Additionally, since recall is measured by segment overlap, the result also shows that Parametric Min-Loss and its surrogate are effective approximating training objectives. Finally, in Figure 7 (far right), we demonstrate the effectiveness of structured learning within our own method. In particular, we test for an increase in recall achieved by combining mid-level grouping cues with diversification seeds. As we expected, recall is significantly boosted with mid-level cues.

9. Conclusion

We introduced Parametric Min-Loss (PML), a novel structured learning framework for parametric energy functions, and demonstrated it in the context of region proposal generation. Our perceptual grouping method learns how to combine multiple cues to generate a set of figure-ground region proposals. By applying the MCL optimization strategy to parametric maxflow, we bridge the gap between learning and inference for parametric energy functions. Moreover, our framework supports efficient superpixel-based diversification that yields a diverse set of region proposals that competes favorably with recent state of the art on VOC 2012. In future work, we plan to use our general framework to learn how we can integrate other classical grouping cues to improve region proposal generation.

Acknowledgements

We thank Richard Zemel, Allan Jepson, Marcus Brubaker, and Raquel Urtasun for valuable discussions on the learning framework.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels. *Ecole Polytechnique Fédéral de Lausssanne (EPFL), Tech. Rep*, 2:3, 2010. 7

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, Jan 2012. 1, 2

[3] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898 – 916, 2011. 7

[4] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. *CVPR*, 2014. 2, 3, 7

[5] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 34(7):1312–1328, 2012. 1, 2, 3, 7

[6] X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals for accurate object class detection. In *NIPS*, 2015. 3

[7] L. Cohen and T. Deschamps. Multiple contour finding and perceptual grouping as a set of energy minimizing paths. *EMMCVPR*, Jan 2001. 2

[8] I. Endres and D. Hoiem. Category independent object proposals. *ECCV*, Jan 2010. 3

[9] P. Felzenswalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004. 1, 7

[10] M. Fromer and A. Globerson. An lp view of the m-best map problem. *NIPS*, Jan 2009. 2

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, Jan 2014. 7

[12] G. Guy and G. Medioni. Inferring global perceptual contours from local features. *CVPR*, Jan 1993. 2

[13] A. Guzman-Rivera, D. Batra, and P. Kohli. Multiple choice learning: Learning to produce multiple structured outputs. *NIPS*, pages 1799–1807, 2012. 2, 5

[14] A. Guzman-Rivera, P. Kohli, D. Batra, and R. Rutenbar. Efficiently enforcing diversity in multi-output structured prediction. *AISTATS*, Jan 2014. 2

[15] A. Humayun, F. Li, and J. Rehg. Rigor: Reusing inference in graph cuts for generating object regions. *CVPR*, Jan 2014. 2, 7

[16] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988. 2

[17] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. *ICCV*, 8, 2007. 2, 3, 4, 6

[18] P. Krähenbühl and V. Koltun. Geodesic object proposals. *ECCV*, Jan 2014. 2, 7

[19] P. Krähenbühl and V. Koltun. Learning to propose objects. *CVPR*, 2015. 2

[20] T. Lee, S. Fidler, and S. Dickinson. Detecting curved symmetric parts using a deformable disc model. *ICCV*, 2013. 4

[21] T. Lee, S. Fidler, and S. Dickinson. Multi-cue mid-level grouping. *ACCV*, 2014. 3, 4, 7, 8

[22] A. Levinshtein, C. Sminchisescu, and S. Dickinson. Optimal contour closure by superpixel grouping. *ECCV*, pages 480–493, 2010. 3, 7

[23] P. Rantalankila, J. Kannala, and E. Rahtu. Generating object segmentation proposals using global and local search. *CVPR*, Jan 2014. 2, 3

[24] X. Ren, C. Fowlkes, and J. Malik. Cue integration for figure/ground labeling. *NIPS*, 2005. 2

[25] A. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: Joint 3d layout and object reasoning from single images. *ICCV*, Jan 2013. 6

[26] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000. 1, 2

[27] M. Szummer, P. Kohli, and D. Hoiem. Learning crfs using graph cuts. *ECCV*, Jan 2008. 2

[28] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, Jan 2005. 4, 5

[29] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 1, 2, 3, 5, 7

[30] S. Ullman and A. Sha'ashua. Structural saliency: The detection of globally salient structures using a locally connected network. *ICCV*, 1988. 2

[31] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *PAMI*, 13(6):583–598, 1991. 1

[32] L. Williams and D. Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. *Neural Computation*, Jan 1997. 2

[33] V. Yanulevskaya, J. Uijlings, and N. Sebe. Learning to group objects. *CVPR*, Jan 2014. 2

[34] S. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *PAMI*, 18(9):884–900, 1996. 2

[35] C. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. *ECCV*, Jan 2014. 2