Scene Labeling Using Beam Search Under Mutex Constraints

Anirban Roy
Oregon State University
Corvallis, OR 97330
royan@eecs.oregonstate.edu

Sinisa Todorovic
Oregon State University
Corvallis, OR 97330
sinisa@eecs.oregonstate.edu

Abstract

This paper addresses the problem of assigning object class labels to image pixels. Following recent holistic formulations, we cast scene labeling as inference of a conditional random field (CRF) grounded onto superpixels. The CRF inference is specified as quadratic program (QP) with mutual exclusion (mutex) constraints on class label assignments. The QP is solved using a beam search (BS), which is well-suited for scene labeling, because it explicitly accounts for spatial extents of objects; conforms to inconsistency constraints from domain knowledge; and has low computational costs. BS gradually builds a search tree whose nodes correspond to candidate scene labelings. Successor nodes are repeatedly generated from a select set of their parent nodes until convergence. We prove that our BS efficiently maximizes the QP objective of CRF inference. Effectiveness of our BS for scene labeling is evaluated on the benchmark MSRC, Stanford Backgroud, PASCAL VOC 2009 and 2010 datasets.

1. Introduction

This paper addresses the problem of scene labeling, where the goal is to label each image pixel with a class label from a set of classes. The classes of interest include objects and scene surfaces (e.g., grass, sky). Real-world images present significant challenges for scene labeling, since objects may appear at different scales, under occlusion, and in a wide range of spatial configurations in the scene.

Prior work has demonstrated that holistic reasoning about occurrences of all classes, their co-occurrences, and spatial layouts offers a viable framework for scene labeling (e.g., [31, 30, 32, 8, 9, 28, 13, 18, 34, 15, 29]). These approaches typically model the scene by a conditional random field (CRF) grounded onto superpixels (or image patches). In this way, they adopt common recognition strategies: a) Smoothness – neighboring image regions may be occupied by the same object, and b) Context – neighboring image regions may be occupied by frequently co-occurring objects.

Motivated by the success of these approaches, we represent the scene as a fully connected CRF grounded onto superpixels, and formulate scene labeling as assignment of class labels to superpixels in CRF inference. Following a well-established line of research [26, 12, 17, 34, 11], we cast CRF inference as quadratic program (QP). In comparison with existing liner programming counterparts, QP involves computing comparatively less variables, provides a separable constraint set in optimization, and allows for a differentiable large-margin parameter estimation.

Our key contribution is a beam search algorithm (BS) for QP-based CRF inference. BS is well-suited for scene labeling, because it:

1. Accounts explicitly for spatial extents of objects;
2. Solves QP entirely in the discrete domain conforming to useful domain constraints;
3. Does not require common convexification and relaxation of QP, and ultimate discretization of the continuos solution;
4. Has low computational costs, allowing for a large number of CRF nodes, and a full-node connectivity, as often needed for modeling long-range dependencies between objects in the scene.

BS starts from an initial labeling of superpixels corresponding to the initial state, as illustrated in Fig. 1. Then, it gradually builds a search tree, where tree nodes correspond to search states, i.e., candidate scene labelings. The tree depth is incremented as new successor states are generated from a subset of states at the current depth. The search continues until convergence, when no "better" successors can be generated. BS is defined by the following three functions: Successor – for stochastic exploration of the search space by randomly generating successor states, given parent states; Heuristic – for selecting a set of B "best" current states for exploration, where B is the input beam-search width parameter; and Score – for selecting the "best" state as the solution.

The Successor function explicitly accounts for spatial extents of objects by *jointly* flipping the class labels of a *connected component* of superpixels when generating new

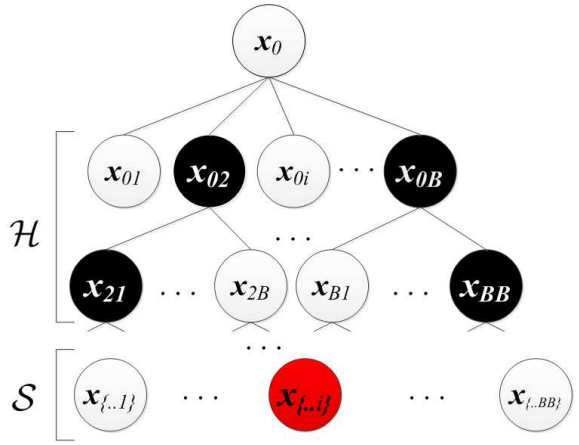

Figure 1. BS: States in the search tree correspond to candidate scene labelings. The tree is gradually expanded by generating successor states from a subset of B best states (black) estimated by the Heuristic function. The Score function selects the optimal leaf state (red) as the solution of CRF inference.

states. The Heuristic function is efficiently computed as a difference between the CRF's conditional log-likelihoods of the parent and successor states. It is efficient because it takes into account only (a few) changes in superpixel labeling between the states, instead of all superpixels. Finally, the Score function efficiently evaluates the CRF conditional log-likelihood of leaf states, and selects the state with the largest Score as an optimal solution. Note that, by construction, the leaves are guaranteed to have the largest Score values among all states in the search tree, and thus it suffices to look for the solution only among the leaf states.

We present effectiveness of our BS for scene labeling on the MSRC [28], Stanford Backgroud [8] , PASCAL VOC 2009 and 2010 [3] datasets.

In the sequel, Sec. 2 points out our contributions relative to prior work; Sec. 3 formulates CRF; Sec. 4 specifies CRF inference as QP; Sec. 5 formulates BS; Sec. 6 defines the CRF potentials, and describes how to learn the potentials and mutex constraints from training data; and Sec. 7 presents our results.

2. Prior Work and Our Contributions

This section reviews related work and points out our contributions in terms of: i) Enforcing hard constraints in CRF inference; ii) Directly solving QP in the discrete domain; and iii) Low complexity of our BS.

Accounting for domain constraints between objects in the scene is important, because they can help resolve competing hypothesis in inference. In this paper, we focus on the mutual-exclusion (mutex) constraints that prohibit certain label assignments (e.g., the sky cannot occur below grass). CRF represents domain constraints with *features* which are weighted to form potential functions. For enforcing constraints in CRF inference, the feature weights should be sufficiently large so as to penalize scene labelings that violate the constraints. However, since the weights cannot be infinite, in some cases CRF inference may overrule the constraints, yielding non-sensical results.

We address this problem by keeping separate domain constraints from the other potential functions of CRF aimed at encoding smoothness and context. We cast CRF inference as a QP with quadratic constraints. Specifically, we use the smoothness and contextual potential functions of the CRF to express the QP objective, and separately use mutual-exclusion constraints of the domain to express the quadratic constraints of the QP.

CRF inference as QP typically requires convexification of the QP objective to allow using standard convex optimization algorithms [26, 12]. Convexification can be avoided, e.g., by using message-passing [16], or gradual progression between linear programming and QP [17]. However, these approaches are not suitable for our fully connected CRF, since their complexity depends on the number of CRF edges. Existing semidefinite programming approximations of QP are also inappropriate, because the matrix of our QP's quadratic objective cannot be assumed as being ("close to") positive semidefinite.

Importantly, most QP solvers relax the optimization constraints to the continuous domain. This would be unsuitable, because domain constraints may be violated under continuous relaxation. In contrast, our BS does not use continuous relaxation, but directly solves QP in the discrete domain, strictly enforcing domain constraints.

Our approach is related to Swendsen-Wang (SW) cut [1], which iterates Metropolis-Hastings reversible jumps. Each jump randomly cuts graph edges and flips the labels of a connected group of nodes for a faster exploration of the search space than other MCMC algorithms. However, SW evaluates CRF in each visited state, which is expensive for large graphs as ours. Also, in practice, SW iterations are often interrupted before convergence, due to long running times. In contrast, our BS is efficiently guided by a heuristic function to select "good" candidate solutions, and guaranteed to converge fast to a local maximum.

Search-based structure prediction methods are gaining momentum in computer vision [10, 5, 14, 24], but they have never been used for scene labeling. Their key limitation is the requirement to approximate the loss function, and thus guide the search. Inspired by HC-Search [25, 2], we instead use a rank-based search strategy that makes search decisions by comparing relative values of the search states assigned by the Heuristic function. As in [25, 2], we use the Heuristic function to guide the beam search, and the Score function to identify the solution. The key difference is that we derive the Heuristic and Score from the original (CRF-based) optimization objective, whereas these two functions

are learned separately in a distributed learning architecture of [25, 2].

3. The CRF Model

Images are partitioned into superpixels, which are used to ground our CRF of the scene. In particular, superpixels are organized in a graph, $G = (V, E)$. V is a set of nodes, $i = 1, \ldots, n$, $|V| = n$, corresponding to superpixels. E is a set of edges $(i, j) \in E$ that capture dependencies between pairs of superpixels i and j. In this paper, we consider a fully connected graph, where edges connect all node pairs $E = V \times V$, $|E| = n^2$.

The CRF associates an indicator random variable X_i with every node $i \in V$. Each X_i takes values from a set of object class labels $L = \{1, 2, ..., k\}$, where $|L| = k$. When $X_i = i' \in L$ then CRF assigns class label i' to superpixel i. The set of all random variables is denoted as $\mathbf{X} = \{X_i : i \in V\}$. The conditional log-likelihood of the CRF is specified as

$$\log P(\mathbf{X}|G) = \sum_{i \in V} \phi_i(X_i = i') + \sum_{(i,j) \in E} \phi_{ij}(X_i = i', X_j = j') - \log Z, \tag{1}$$

where $\{i', j'\} \in L$, and Z is the partition function. The unary potential $\phi_i(X_i = i')$ is defined as a log-likelihood of X_i having label $i' \in L$. The pairwise potential $\phi_{ij}(X_i = i', X_j = j')$ represents a joint log-likelihood of X_i and X_j having labels i' and j', respectively. In the following, we will use shorthand notation $\phi_{ii'} = \phi_i(X_i = i')$, and $\phi_{ii'jj'} = \phi_{ij}(X_i{=}i', X_j{=}j')$. Sec. 6 specifies the unary and pairwise potentials.

We formulate our scene labeling problem as finding the MAP assignment $\hat{\mathbf{X}} = \arg\max_{\mathbf{X}} P(\mathbf{X}|G)$. In the following section, we explain how to conduct this inference.

4. CRF Inference as QP

This section formulates the MAP assignment problem as QP. We begin by deriving the quadratic objective of QP, and then extend that formulation to include domain constraints.

It is convenient to express $\log P(\mathbf{X}|G)$, given by (1), in terms of binary random variables $x_{ii'} \in \{0, 1\}$ over superpixel-label pairs. When $X_i = i'$ we have $x_{ii'} = 1$, and when $X_i \neq i'$ we have $x_{ii'} = 0$. A column vector of all $(n \cdot k)$ binary random variables is denoted as $\boldsymbol{x} = [\ldots x_{ii'} \ldots]^\top$. Thus, the MAP assignment problem can be posed as

$$\begin{aligned} &\max. \sum_{i \in V; i' \in L} \phi_{ii'} x_{ii'} + \sum_{(i,j) \in E; i', j' \in L} \phi_{ii'jj'} x_{ii'} x_{jj'} \\ &\text{s.t.} \quad \text{for all } i \in V, \sum_{i' \in L} x_{ii'} = 1, \quad \boldsymbol{x} \in \{0, 1\}^{n \cdot k} \end{aligned} \tag{2}$$

The quadratic objective of (2) can be compactly expressed as $\boldsymbol{x}^\top \Phi \boldsymbol{x}$, where Φ is an $(n{\cdot}k){\times}(n{\cdot}k)$ affinity matrix whose elements are the unary and pairwise potentials. The off-diagonal elements of Φ are defined as $\Phi_{(ii'),(jj')} = \phi_{ii'jj'}$, and the main diagonal elements of Φ are defined as $\Phi_{(ii'),(ii')} = \phi_{ii'}$.

As mentioned in Sections 1 and 2, our next goal is to incorporate domain constraints in QP, which are expected to improve the quality of solutions by eliminating illegal configurations from consideration. In this paper, we focus on the mutual-exclusion (mutex) constraints that prohibit certain non-sensical label assignments. For example, suppose that a QP solver considers a hypothesis that a superpixels i and j get assigned candidate class labels i' = "grass" and j' = "sky", where i is located at the top of the image, and j at the bottom. As common-sense knowledge rules out that grass can occur above the sky in natural scenes, if i gets assigned label i' = "grass", i.e., $x_{ii'} = 1$, then j must not be assigned label j' = "sky", i.e., the QP solver must set $x_{jj'} = 0$. This type of reasoning can be formalized as the equality constraint: $x_{ii'} \cdot x_{jj'} = 0$. Intuitively, this equality constraint strictly enforces that only one of the two labels are allowed for the two superpixels.

Following the approach of [21], all mutex constraints can be compactly represented as

$$\boldsymbol{x}^\top M \boldsymbol{x} = 0, \tag{3}$$

where M is an $(n{\cdot}k){\times}(n{\cdot}k)$ binary constraint matrix. When its elements are set to one, $M_{(ii'),(jj')} = 1$, then the corresponding label assignments are prohibited $x_{ii'} \cdot 1 \cdot x_{jj'} = 0$. Conversely, when $M_{(ii'),(jj')} = 0$ then superpixels i and j may be assigned any arbitrary class labels from L, because the quadratic equality constraint still remains satisfied, $x_{ii'} \cdot 0 \cdot x_{jj'} = 0$. Note that M is typically sparse. Sec. 6 specifies M for each image.

Further, it is convenient to merge the set of linear constraints of the problem in (2) — namely that for all $i \in V$, $\sum_{i' \in L} x_{ii'} = 1$ — with the quadratic equality constraints in (3). For every superpixel i, we set all the corresponding elements of matrix M to one, $M_{(ii'),(ij')} = 1$, if $i' \neq j'$. This prohibits illegal assignments of multiple distinct labels to a single superpixel, since for all i we will have $x_{ii'} \cdot 1 \cdot x_{ij'} = 0$, if $i' \neq j'$.

By using the affinity matrix Φ, and the constraint matrix M, from (2), we finally derive the following QP:

$$\begin{aligned} &\max. \ \boldsymbol{x}^\top \Phi \boldsymbol{x} \\ &\text{s.t.} \quad \boldsymbol{x}^\top M \boldsymbol{x} = 0, \quad \boldsymbol{x} \in \{0, 1\}^{n \cdot k}. \end{aligned} \tag{4}$$

Note that (4) does not relax the original problem in (2). While the constraints in (4) and (2) are not equivalent, the objective and constraints of (4) make the problem of (4) equivalent to that of (2). The constraints in (2) enforce that

every superpixel is assigned exactly one label. The constraints in (4) only enforce that every superpixel is not assigned multiple labels. But the objective of (4) will not be maximum if a superixel is unlabeled.

In the following section, we specify our new algorithm for solving the QP problem in (4).

5. Beam Search

Given an image and its superpixels, the search starts from their initial labeling $\boldsymbol{x}_0$, and gradually builds the search tree with new states $\boldsymbol{x}$. At every tree depth, BS considers at most B best states for further exploration, based on Heuristic values of these states. Exploration consists of stochastically sampling successor states from the selected parent states, which increments the current tree depth. The sequential tree expansion stops when no successor state gives a positive Heuristic value. Below, we formally define the elements of our search framework.

State-space: The state-space is defined as $\Omega = \{\boldsymbol{x} : \boldsymbol{x} \in \{0,1\}^{nk},\ \boldsymbol{x}^\top M\boldsymbol{x} = 0\}$. The states correspond to candidate scene labelings respecting mutex constraints.

Successor function, $\Gamma : \boldsymbol{x} \rightarrow \boldsymbol{x}'$, generates new states $\boldsymbol{x}'$ from $\boldsymbol{x}$. Γ modifies a given state $\boldsymbol{x}$ by jointly changing the labels of a group of superpixels in $\boldsymbol{x}$, resulting in $\boldsymbol{x}'$ which strictly satisfies mutex constraints. Thus, Γ defines: i) How to select a set of superpixels to be re-labeled; and ii) How to determine their new labels, as explained below. Fig. 2 illustrates an example of generating a new state.

For choosing a set of superpixels, we first probabilistically cut edges in E whose pairwise potentials are below a random threshold. Specifically, edges $(i,j) \in E$ are characterized by pairwise potentials $\phi_{ii'jj'}$, where i and j are assigned labels i' and j' in state $\boldsymbol{x}$. A threshold is randomly selected in the range between the minimum and maximum values of $\phi_{ii'jj'}$ to cut all edges with pairwise potentials less than the threshold. This partitions G into a set of disconnected subgraphs. We then randomly select one of the subgraphs, and then, within the subgraph, again randomly select a connected component (CC) of superpixels that are neighbors and have the same label. To respect spatial extents of objects, we jointly re-label all superpixels in the selected CC to the label of one of the neighboring connected components in the selected subgraph. This encourages spatial smoothness, and removes holes within objects in the resulting scene labeling.

The successor state $\boldsymbol{x}'$ is accepted if mutex constraints are satisfied, $\boldsymbol{x}'^\top M\boldsymbol{x}' = 0$. We efficiently compute this quadratic as follows. Re-labeling of nodes in the selected CC does not change the entire $\boldsymbol{x}$, but only a part of this vector. Let us denote this difference as $\boldsymbol{\delta} = \boldsymbol{x}' - \boldsymbol{x}$, which is non-zero for only a few indices of nodes that belong to the

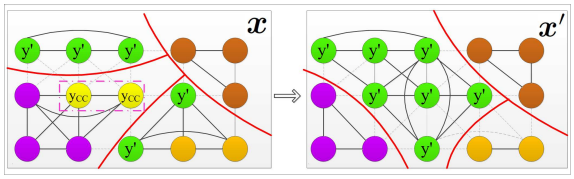

Figure 2. Consider a state $\boldsymbol{x}$ with the shown labeling. After randomly cutting the edges we have a set of subgraphs which are partitioned by red curves. We randomly select a CC with labels y_{CC} in a random subgraph. Updated labels are randomly chosen as the label of neighboring CC (y' in this case) which leads to a new state $\boldsymbol{x}'$. Note that CRF edges are due to less pairwise potential, so subgraph boundaries does not imply object class boundary.

CC. Then, the mutex constraints for $\boldsymbol{x}'$ can be expressed as

$$\boldsymbol{x}'^\top M\boldsymbol{x}' = (\boldsymbol{x}+\boldsymbol{\delta})^\top M(\boldsymbol{x}+\boldsymbol{\delta}) = 2\boldsymbol{\delta}^\top M\boldsymbol{x} + \boldsymbol{\delta}^\top M\boldsymbol{\delta} = 0, \tag{5}$$

because it is already guaranteed that the parent state satisfies mutex constraints, $\boldsymbol{x}^\top M\boldsymbol{x} = 0$. Complexity of verifying (5) is low, because the CC would typically consist of only a few nodes, and M is sparse. This step guarantees that the final solution found by BS strictly satisfies all mutex constraints. The stochastic generation of successor states helps avoid local optima.

Heuristic function, $\mathcal{H}(\boldsymbol{x}', \boldsymbol{x})$, evaluates new states $\boldsymbol{x}'$ given their parent $\boldsymbol{x}$, and guides the expansion of the search tree by selecting at most B best successors of $\boldsymbol{x}$. Ideally, new states should be evaluated using the QP objective, $\boldsymbol{x}'^\top \Phi\boldsymbol{x}'$, stated in (4). This would ensure that BS is guided toward an optimal solution of the QP. However, computing the quadratic objective for large CRFs as ours at every candidate state would be prohibitively expensive. To address complexity issues, we again use the difference vector $\boldsymbol{\delta}$ between $\boldsymbol{x}'$ and $\boldsymbol{x}$ to express the QP objective as

$$\boldsymbol{x}'^\top \Phi\boldsymbol{x}' = \boldsymbol{x}^\top \Phi\boldsymbol{x} + 2\boldsymbol{\delta}^\top \Phi\boldsymbol{x} + \boldsymbol{\delta}^\top \Phi\boldsymbol{\delta}. \tag{6}$$

For all new states $\boldsymbol{x}'$, we note that (6) has the same first term, $\boldsymbol{x}^\top \Phi\boldsymbol{x}$. Fortunately, evaluating the other two terms in (6) is not computationally expensive, because they account for only a few nodes in the CC. This motivates our definition of Heuristic as

$$\mathcal{H}(\boldsymbol{x}', \boldsymbol{x}) = 2\boldsymbol{\delta}^\top \Phi\boldsymbol{x} + \boldsymbol{\delta}^\top \Phi\boldsymbol{\delta}. \tag{7}$$

A more global heuristic function might better evaluate candidate states, but at the price of increasing computational complexity relative to ours.

The Strategy for selecting B best successors is to keep generating $\boldsymbol{x}' = \Gamma(\boldsymbol{x})$ until we obtain B new states that satisfy mutex constraints *and* yield a positive Heuristic, $\mathcal{H}(\boldsymbol{x}', \boldsymbol{x}) > 0$. As we prove below, the latter requirement ensures that successors must monotonically increase the QP objective. BS stops when no successor can satisfy both of the two requirements after a sufficiently long running time.

Score function, $\mathcal{S}(\boldsymbol{x}_l)$, is efficiently computed in at most B^2 leaf states by summing already available Heuristic values $\mathcal{H}(\boldsymbol{x}', \boldsymbol{x})$ along the path, $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_t, \ldots \boldsymbol{x}_l\}$, that connects the leaf $\boldsymbol{x}_l$ with the root $\boldsymbol{x}_0$ (i.e., initial state):

$$\mathcal{S}(\boldsymbol{x}_l) = \boldsymbol{x}_0^\top \Phi \boldsymbol{x}_0 + \sum_{t=0}^{l} H(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t). \tag{8}$$

Again, note that the first term in (8) has to be computed only once for all the leaves. The final scene labeling solution is chosen among the leaves as $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}_l} \mathcal{S}(\boldsymbol{x}_l)$.

It is straightforward to show that maximizing score $\mathcal{S}$ in (8) amounts to optimizing the QP objective specified in (4). Thus, BS monotonically increases the QP objective subject to mutex constraints, given by (4), where the solution is found when scoring all the leaves $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}_l} \mathcal{S}(\boldsymbol{x}_l)$.

From (8), our complexity is $O(n{\cdot}k){+}B{\cdot}l{\cdot}O((n{\cdot}k){+}n^2)$. The first term comes from the initial labeling of n superpixels with k labels. The second term comes from subsequent generating and evaluating B states at l search levels. Selecting a CC for generating a new state requires $O(n^2)$ computations for n^2 edges in the CRF, and complexity of verifying that a candidate state satisfies the mutex constraints and has positive heuristic is $O(n \times k)$. Note that the complexity of BS grows linearly with the label set. It is worth noting that BS can be easily parallelized. This parallel implementation provides significant speedup resulting average convergence time less than a second per image on an Intel i7 machine with 8 GB memory.

6. CRF Potentials and Mutex Matrix

This section first describes the superpixels segmentation method, then specifies the unary and pairwise potentials of our CRF. After that, it describes how to compute the potentials and finally specifies how to estimate the mutex matrix M.

Superpixel segmentation. For a fair comparison with the state of the art, we use the same method for extracting superpixels as that used in related work [28, 9, 13, 34, 15] – namely, the low-level segmentation algorithm of [4].

Unary Potential is defined as a sum of texture, color and location potentials, $\phi_{ii'} = \phi^{\text{tex}}_{ii'} + \phi^{\text{col}}_{ii'} + \phi^{\text{loc}}_{ii'}$. $\phi^{\text{tex}}_{ii'}$ is specified as confidence of a boosted classifier, where each weak classifier is a decision stamp based on a multi-class logistic regression of texture features. For texture features of every superpixel i, we use the response of 17-dimensional filter bank of Gaussian and Laplacians-of-Gaussian filters, as in [28]. $\phi^{\text{col}}_{ii'}$ is computed as the negative Log-Mixture of Gaussian of the 16×3 color histogram of superpixel i for class i'. $\phi^{\text{loc}}_{ii'}$ is defined as the negative log-prior (i.e., frequency) of the class i' appearing at the normalized location of i. We use the piecewise training approach [28] to learn each of the potentials separately.

Pairwise Potential is defined as a sum of color-smoothness and distance potentials, $\phi_{ii'jj'} = \phi^{\text{col}}_{ii'jj'} + \phi^{\text{dis}}_{ii'jj'}$, as in [34]. The color pairwise potential between two superpixels i and j is computed as $\phi^{\text{col}}_{ii'jj'} = g(\boldsymbol{I}_i{-}\boldsymbol{I}_j)$, if $i' = j'$, else 0, where g is a negative log-Gaussian with identity covariance matrix, and $\boldsymbol{I}_i, \boldsymbol{I}_j$ are the color histograms of superpixels i and j. The distance potential is defined as $\phi^{\text{dis}}_{ii'jj'} = g(\boldsymbol{s}_i{-}\boldsymbol{s}_j)$, if $i' = j'$, else 0, where $\boldsymbol{s}_i, \boldsymbol{s}_j$ are the locations of superpixels i, j.

Mutex estimation. For specifying the mutex matrix M, we make the assumption that the training dataset is sufficiently large. We use the frequency of co-occurrence of object classes in particular spatial layouts, estimated directly from training data. Specifically, we define an augmented label set {(object, object, configuration)}. For configuration labels we use four qualitative spatial relations: "left", "right", "above", "below". For every pair of superpixels (i, j) in training data we identify their configuration label, i.e., estimate one of the four spatial relations, relative to i. Then, we count the number of times the true class labels i'^* and j'^* of every pair of superpixels (i, j) occur in the four configurations. When a new image is encountered, every pair of its superpixels (i, j) is first assigned one of the four configuration labels, and then all corresponding elements of matrix M are set to either one, $M_{(ii'),(jj')} = 1$, if the pair of object classes (i', j') has never been observed in the spatial configuration of superpixels (i, j) in training; or set to zero, $M_{(ii'),(jj')} = 0$, otherwise.

7. Results

Datasets. We evaluate BS on four benchmark datasets: the MSRC dataset [28], the Stanford Background dataset (SBD) [8] and the PASCAL VOC 2009 and 2010 [3] datasets. The MSRC dataset consists of 591 images of 21 object classes. We duplicate the evaluation setup of [28] to have the standard split of training and test images. The SBD dataset has 715 images having seven background classes and one generic foreground class. We follow the five fold cross validation experiment setup of [8]. The PASCAL VOC 2009 and 2010 datasets consist of images of 20 object classes. Here we train on training images and test on validation images as done in [20] and [15]. Accuracy is measured as the standard VOC measure [3].

For training, we compute the ground-truth label of a superpixel as the majority ground-truth class labels of its pixels. For testing, we compute the label assignment accuracy at the pixel level. As convergence time of BS and final solution depend on the initial state, we initialize the search to a structured prediction of logistic regression. We use top 50 ranked logistic regression predictions as multiple initial states, and then run BS for each, and finally select the best solution. Note that these multiple searches can be paral-

lelized for efficiency.

In each step of BS when the search tree-depth is incremented, we choose at most B best candidates from $2B$ successor states of a parent state. Overall, the acceptance rate of new states has a large standard deviation over the search steps, since while generating new states, we discard those that do not satisfy mutex constraints *or* do not have positive heuristic score.

Evaluation of Input Parameters. We evaluate the following input parameters: beam width B, and number of initial states. Fig. 5 shows that the accuracy increases initially as B becomes larger, but saturates after $B = 10$. The same effect can be noticed for varying the number of initial sates. As we increase B, BS keeps a larger number of promising candidate sates, but after a certain limit (10 in our case), the beam gets populated with spurious sates. This does not affect our accuracy, but increases our running time. In our experiments, we use $B = 10$, and set the number of initializations of BS to 50. From figures 4, 5, our running time has a linear-like profile with respect to both B and initial points, when BS is parallelized over the beam.

Baselines. We compare BS with the following four baselines, B1–B4. Comparison with the baselines is done on the MSRC dataset.

B1. Swendsen-Wang cut (SW-cut): SW-cut addresses the intractable CRF inference with the Metropolis-Hastings (MH) sampling algorithm. MH draws samples $\boldsymbol{x}$ from the CRF's posterior, $P(\boldsymbol{x}|G)$, to generate new states. The jumps between the states are reversible, and governed by a proposal distribution $Q(\boldsymbol{x} \rightarrow \boldsymbol{x}')$. This also cuts CRF edges for choosing a connected component, CC, and updates its label to a random label. The proposal is accepted if the acceptance rate, α, drawn from the uniform distribution, $U(0,1)$, satisfies $\alpha < \min\{1, \frac{Q(\boldsymbol{x}' \rightarrow \boldsymbol{x})}{Q(\boldsymbol{x} \rightarrow \boldsymbol{x}')} \frac{P(\boldsymbol{x}'|G)}{P(\boldsymbol{x}|G)}\}$. Here, CRF posterior, $P(\boldsymbol{x}|G)$ is computed as in (1), and proposal distribution is proportional to the number of edges which are cut during CC selection as in [23]. For fair comparison, we keep the CC selection method the same as ours. The accuracy of 81.5% is 10% less than ours with higher running time of 30-32sec (Tab. 1). This added accuracy comes from exploring B states in every step of the search instead of only one state as in SW-cut.

B2. QP without mutex constrains (QPWOM): In this baseline, we exclude the hard mutex constraints while conducting the inference. We only keep the constraint that does not allow a superpixel to have multiple labels. This justifies the importance of having mutex constraints to guide the search (Fig. 4). Fig. 3 shows an example where QPWOM results in an infeasible labeling: superpixels labeled with 'sky' are below the superpixels with label 'boat'. In BS such labelings are restricted due to hard mutex constraints.

B3. QP with relaxed mutex constrains: Here we use a standard QP solver (IBM CPLEX Optimizer) aiming to solve the optimization : $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} \boldsymbol{x}^T(\Phi - \gamma M)\boldsymbol{x}$ ($\gamma \geq \max_i \sum_j \Phi_{ij}$) while relaxing the integer constraints as $\boldsymbol{x} \in [0,1]^{nk}$. This makes the mutex as soft constraints. The accuracy is 85.4% with running time 22 sec, which is approx. 6% less than ours with higher running time. Comparison to B3 shows solving the optimization in the discrete domain is more efficient than the relaxed counterpart.

Figure 3. Comparison with QPWOM with BS on an image from MSRC dataset. Infeasible labeling is done by QPWOM due to the missing mutex constarints.

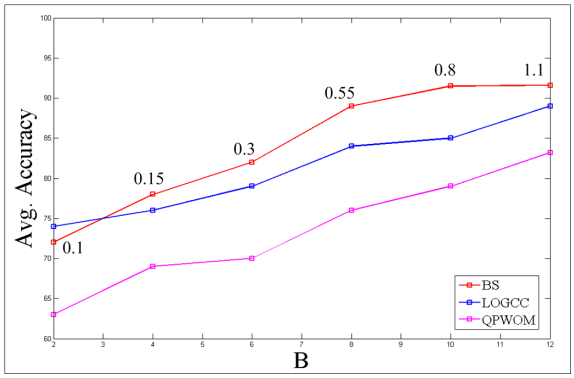

Figure 4. Comparison with baselines on the MSRC dataset. Beam Search (BS) is our proposed approach, LOGCC is B4 where nodes of a CC is updated with logistic learning and QPWOM is our approach without mutex constraints. We also show the running time (sec) of BS for each B.

B4. CC updates with logistic regression (LOGCC): Here, instead of choosing the updated label of a CC from the neighboring superpixels, we update the label of the CC using a multiclass logistic regression learner. Thus each node of the CC is assigned the label having highest class likelihood measured by logistic regression classifier. We notice that the performance is not improved with the additional learning (Fig. 4).

State of the art comparison: Tab. 1 shows the comparison with the state-of-art methods on the MSRC dataset, where our accuracy 4.5% better the previous best approach [34] which uses QP relaxation as inference for fully connected CRF model over pixels. Comparisons on the SBD and PASCAL 09, 10 datasets are shown in the Tab. 2. On the SBD, our approach is slightly worse than the two state-of-the approaches ([19, 27]) which, unlike ours, use multi-scale segments and higher order potentials. On PASCAL 2009 dataset our approach achieves higher accuracy than the previous state-of-art approach ([20]) by 2.9%. On PASCAL 2010 dataset, two modified versions of the approaches presented in [7] and [20] achieve better performances than us. These methods use object segmentation or foreground

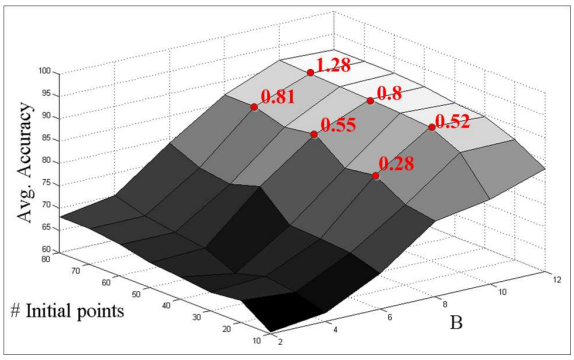

Figure 5. Evaluation of input parameters (B and number of initial starting points) on the MSRC dataset. We vary B in X axis and number of initial starting points in Y axis. Running times (sec) are shown for specific B and number of initial points

.

Method	MSRC	Test time
[6]	70.0	N/A
[8]	76.4	N/A
[23]	82.9	30-32s
[15]	86.0	0.2s
[33]	86.5	N/A
[34]	87.0	N/A
Ours	91.5	0.8 s

Table 1. State-of-the-art comparison of pixel classification accuracy(%) and computation times(seconds) per image on the MSRC dataset.

Method	SBD
[8]	76.4
[22]	76.9
[29]	74.1
[19]	81.9
[27]	82.9
Ours	81.1

Method	P '09
[20]	37.2
[7]	34.1
[23]	35.7
Ours	40.1

Method	P '10
[33]	31.2
[15]	30.2
[7]	40.1
[20]	39.7
Ours	34.2

Table 2. State-of-the-art comparison of segmentation accuracy(%) on the SBD (left), PASCAL VOC 2009 (middle) and 2010 (right) datasets.

Figure 7. Failure case of BS on an image from the Pascal 09 dataset. Ground Truth = GT.

segmentation as additional cues, whereas we do not use such cues.

Fig. 6 presents qualitative results of our approach on four datasets and Fig. 7 shows a failure case of BS for an image from the Pascal 09 dataset where the object class 'gas cylinder' is confused with the 'bottle' class and back portion of the person body is not detected due to the presence of shadow in the image.

8. Conclusion

We have presented a new approach to scene labeling. Scene labeling is posed as the MAP assignment of a fully connected CRF, grounded onto superpixels. The MAP assignment is formulated as quadratic program, and solved using our new Beam Search (BS) algorithm. BS uses the following three functions to build a search tree, where search states correspond to candidate scene labelings. The Successor function generates successor states from a subset of parents. The Heuristic function evaluates and selects top B states for exploration. The Score function finds the leaf that provably maximizes the QP objective of our CRF inference. BS is well-suited for scene labeling, because it: solves the QP in the discrete domain strictly conforming to useful domain constraints, and has low computational costs, allowing for a large number of CRF nodes and full-node connectivity.

Our experimental evaluation demonstrates that BS outperforms the state of art on some benchmark datasets (e.g., MSRC) and achieves competitive performance on the other datasets (e.g., Stanford Background). Also, when we account for inconsistency constraints from domain knowledge, performance is improved by 9% on the MSRC dataset relative to a variant of our approach that ignores the constraints. Interestingly, initializing BS with predictions of class labels by logistic regression does not notably improve performance over the case when BS is initialized with a random selection of class labels. BS is computationally efficient, and can also be easily parallelized.

Acknowledgment

This work was supported in part by grant NSF RI 1302700.

References

[1] A. Barbu and S.-C. Zhu. Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities. *PAMI*, 27(8):1239–1253, 2005. 2

[2] J. R. Doppa, A. Fern, and P. Tadepalli. HC-search: A learning framework for search-based structured prediction. *JAIR*, 2014. 2, 3

[3] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. 2, 5

[4] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004. 5

[5] P. F. Felzenszwalb and D. Mcallester. The generalized A* architecture. *JAIR*, 29, 2007. 2

[6] C. Galleguillos, B. McFee, S. Belongie, and G. Lanckriet. Multi-class object localization by combining local contextual interactions. In *CVPR*, 2010. 7

[7] J. M. Gonfaus, X. Boix, J. Van De Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzalez. Harmony potentials for joint classification and segmentation. In *CVPR*, 2010. 6, 7

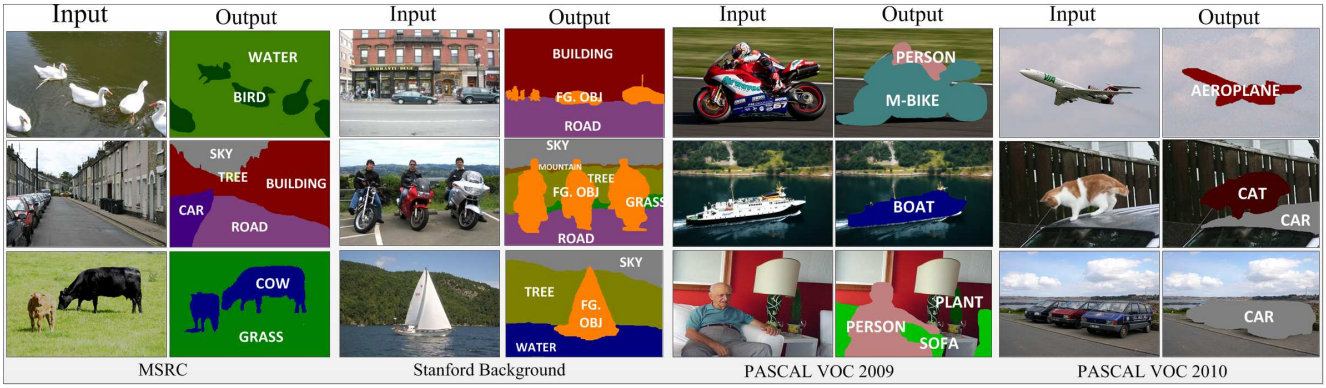

Figure 6. Qualitative results on the MSRC, Stanford background and PASCAL 09, 10 datasets.

[8] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009. 1, 2, 5, 7

[9] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *IJCV*, 80(3):300–316, 2008. 1, 5

[10] P. Gupta, D. Doermann, and D. DeMenthon. Beam search for feature selection in automatic SVM defect classification. In *ICPR*, 2002. 2

[11] J. Jancsary, S. Nowozin, and C. Rother. Learning convex QP relaxations for structured prediction. In *ICML*, 2013. 1

[12] J. Kappes and C. Schnörr. MAP-inference for highly-connected graphs with DC-programming. In *Pattern Recognition*, pages 1–10. Springer, 2008. 1, 2

[13] P. Kohli, L. Ladicky, and P. H. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 82(3):302–324, 2009. 1, 5

[14] I. Kokkinos. Rapid deformable object detection using dual-tree branch-and-bound. In *NIPS*, 2011. 2

[15] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. In *NIPS*, 2012. 1, 5, 7

[16] A. Kumar and S. Zilberstein. Message-passing algorithms for quadratic programming formulations of MAP estimation. In *UAI*, 2011. 2

[17] A. Kumar, S. Zilberstein, and M. Toussaint. Message-passing algorithms for MAP estimation using DC programming. In *AISTAT*, 2012. 1, 2

[18] M. P. Kumar and D. Koller. Efficiently selecting regions for scene understanding. In *CVPR*, 2010. 1

[19] V. S. Lempitsky, A. Vedaldi, and A. Zisserman. Pylon model for semantic segmentation. In *NIPS*, 2011. 6, 7

[20] F. Li, J. Carreira, and C. Sminchisescu. Object recognition as ranking holistic figure-ground hypotheses. In *CVPR*, 2010. 5, 6, 7

[21] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, 2012. 3

[22] D. Munoz, J. A. Bagnell, and M. Hebert. Stacked hierarchical labeling. In *ECCV*, 2010. 7

[23] N. Payet and S. Todorovic. Hough forest random field for object recognition and segmentation. *PAMI*, 2012. 6, 7

[24] N. Payet and S. Todorovic. SLEDGE: sequential labeling of image edges for boundary detection. *IJCV*, 104(1):15–37, 2013. 2

[25] J. Rao Doppa, A. Fern, and P. Tadepalli. Structured prediction via output space search. *JMLR*, 15, 2014. 2, 3

[26] P. Ravikumar and J. Lafferty. Quadratic programming relaxations for metric labeling and markov random field map estimation. In *ICML*, 2006. 1, 2

[27] X. Ren, L. Bo, and D. Fox. RGB-(D) scene labeling: Features and algorithms. In *CVPR*, 2012. 6, 7

[28] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1):2–23, 2009. 1, 2, 5

[29] G. Singh and J. Košecká. Nonparametric scene parsing with adaptive feature relevance and semantic context. In *CVPR*, 2013. 1, 7

[30] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *NIPS*, 2004. 1

[31] J. Verbeek and W. Triggs. Scene segmentation with CRFs learned from partially labeled images. In *NIPS*, 2007. 1

[32] L. Yang, P. Meer, and D. J. Foran. Multiple class segmentation using a unified framework over mean-shift patches. In *CVPR*, 2007. 1

[33] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012. 7

[34] Y. Zhang and T. Chen. Efficient inference for fully-connected CRFs with stationarity. In *CVPR*, 2012. 1, 5, 6, 7