Global Connectivity Potentials for Random Field Models

Sebastian Nowozin and Christoph H. Lampert
Max Planck Institute for Biological Cybernetics
Spemannstr. 38, 72076 Tübingen, Germany
{sebastian.nowozin,christoph.lampert}@tuebingen.mpg.de

Abstract

Markov random field (MRF, CRF) models are popular in computer vision. However, in order to be computationally tractable they are limited to incorporate only local interactions and cannot model global properties, such as connectedness, which is a potentially useful high-level prior for object segmentation. In this work, we overcome this limitation by deriving a potential function that enforces the output labeling to be connected and that can naturally be used in the framework of recent MAP-MRF LP relaxations. Using techniques from polyhedral combinatorics, we show that a provably tight approximation to the MAP solution of the resulting MRF can still be found efficiently by solving a sequence of max-flow problems. The efficiency of the inference procedure also allows us to learn the parameters of a MRF with global connectivity potentials by means of a cutting plane algorithm. We experimentally evaluate our algorithm on both synthetic data and on the challenging segmentation task of the PASCAL VOC 2008 data set. We show that in both cases the addition of a connectedness prior significantly reduces the segmentation error.

1. Introduction

We consider a discrete conditional random field [20, 28] representing a distribution $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w})$ over a discrete label set $\boldsymbol{y} \in \mathcal{Y}$, given a sample $\boldsymbol{x} \in \mathcal{X}$ and a parameter vector $\boldsymbol{w} \in \mathbb{R}^d$. The distribution is a Gibbs distribution over the possible labels,

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) = \frac{1}{Z} \exp\left(-E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w})\right),$$

where $E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w})$ is an *energy function* and $Z = \sum_{\boldsymbol{y} \in \mathcal{Y}} \exp\left(-E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w})\right)$ is a normalization constant known as *partition function*. The energy function is representable in terms of the graph structure of the random field as a sum over *potential functions* $\psi_c : \mathcal{Y}_c \times \mathcal{X} \times \mathbb{R}^d \to \mathbb{R}_+ \cup \{\infty\}$ over the cliques $c \in \mathcal{C}$ of the graph, *i.e.*

$$E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w}) = \sum_{c \in \mathcal{C}} \psi_c(\boldsymbol{y}_c; \boldsymbol{x}, \boldsymbol{w}).$$

A convenient simplification is to define the potential functions as inner products between the parameter vector $\boldsymbol{w}$ and a *feature function* ϕ which is independent of $\boldsymbol{w}$, that is $\psi_c(\boldsymbol{y}_c; \boldsymbol{x}, \boldsymbol{w}) := \boldsymbol{w}^\top \phi_c(\boldsymbol{y}_c, \boldsymbol{x})$. This makes the overall model *log-linear*, as the potential function and hence the energy are linear functions in $\boldsymbol{w}$. If we treat all cliques of size k in the same way – termed *clique template* in [28], we can define individual feature functions $\phi_c^{(k)}(y_c, \boldsymbol{x})$ and use one weight vector $\boldsymbol{w}_k$ for all cliques of the same size. Then, the energy can be written as follows.

$$E(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w}) = \sum_{i \in V} \boldsymbol{w}_1^\top \phi_i^{(1)}(y_i, \boldsymbol{x}) + \sum_{(i,j) \in V \times V} \boldsymbol{w}_2^\top \phi_{i,j}^{(2)}(y_i, y_j, \boldsymbol{x}) + \cdots + \boldsymbol{w}_{|V|}^\top \phi_V^{(|V|)}(\boldsymbol{y}, \boldsymbol{x}). \quad (1)$$

Many computer vision applications assume a grid structure for the graph such that the cliques $c \in \mathcal{C}$ are only single nodes and pairs of nodes, hence only $\boldsymbol{w}_1$, $\boldsymbol{w}_2$ and $\phi^{(1)}$ and $\phi^{(2)}$ are used. The function $\phi_i^{(1)}(y_i, \boldsymbol{x})$ is the node feature function, extracting a feature vector at node i for a given labeling y_i. Likewise, the edge feature function $\phi_{i,j}^{(2)}(y_i, y_j, \boldsymbol{x})$ extracts a feature vector for the edge (i, j) with respective node labeling y_i and y_j. Restricting the energy to only pairwise potentials limits the modeling power to local properties but allows efficient algorithms such as graph cuts [4] to minimize (1), the so called MAP problem.

Maximum Aposteriori Inference. For a given sample $\boldsymbol{x}$ and weight vector $\boldsymbol{w}$, it is of great practical importance to find the maximum aposteriori (MAP) labeling $\boldsymbol{y}$, that is, to solve for

$$\boldsymbol{y}^* = \operatorname{argmax}_{\boldsymbol{y} \in \mathcal{Y}} p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) = \operatorname{argmin}_{\boldsymbol{y} \in \mathcal{Y}} E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w}).$$

Linear Programming Relaxation. Recently, linear programming relaxations have been rediscovered [32, 34, 37] for approximately solving for the MAP solution $\boldsymbol{y}^*$ when the underlying graph $G = (V, E)$ consists of single and edge potentials. The MAP problem can then be formulated exactly as integer linear program (ILP). By relaxing the integer requirement one can obtain a corresponding linear program (LP). In order to avoid confusion, in the following integer linear program, only $\boldsymbol{\mu}$ are variables, all remaining expressions are constants. The variable $\mu_i(y_i) \in \{0, 1\}$ indicates whether node i is in state $y_i \in \mathcal{Y}_i$. The variable

$\mu_{i,j}(y_i, y_j) \in \{0, 1\}$ indicates whether node i is in state $y_i \in \mathcal{Y}_i$ *and* node j is in state $y_j \in \mathcal{Y}_j$.

$$\begin{aligned}
\min_{\boldsymbol{\mu}} \quad & \sum_{i \in V} \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) \left(\boldsymbol{w}_1^\top \phi_i^{(1)}(y_i, \boldsymbol{x}) \right) \qquad (2) \\
& + \sum_{\substack{(i,j) \\ \in E}} \sum_{\substack{(y_i, y_j) \\ \in \mathcal{Y}_i \times \mathcal{Y}_j}} \mu_{i,j}(y_i, y_j) \left(\boldsymbol{w}_2^\top \phi_{i,j}^{(2)}(y_i, y_j, \boldsymbol{x}) \right) \\
\text{sb.t.} \quad & \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) = 1, \qquad i \in V, \\
& \sum_{y_j \in \mathcal{Y}_j} \mu_{i,j}(y_i, y_j) = \mu_i(y_i), \ (i,j) \in E, y_i \in \mathcal{Y}_i, \\
& \mu_i(y_i) \in \{0, 1\}, \quad i \in V, y_i \in \mathcal{Y}_i, \\
& \mu_{i,j}(y_i, y_j) \in \{0, 1\}, \ (i,j) \in E, (y_i, y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j.
\end{aligned}$$

The first set of equality constraints enforce that each node is assigned exactly one label. The second set of equality constraints enforce proper consistency between node and edge states. Given a solution vector $\boldsymbol{\mu}$ to the ILP (2) the labeling $\boldsymbol{y}^*$ is obtained by simply setting $y_i \leftarrow \text{argmax}_{y_i \in \mathcal{Y}_i} \mu_i(y_i)$.

The integer program (2) is exact but NP-hard. The corresponding *LP relaxation* is obtained by relaxing the last two set of constraints to the range $[0; 1]$. The LP relaxation has been analyzed extensively [32, 33, 34]. Although linear programming is among the best developed numerical disciplines [2], the primal LP (2) is practically restricted to medium sized graphs with a few ten thousand nodes and tens of node labels, because on the order of $O(|V|^2(\max_{i \in V} |\mathcal{Y}_i|)^2)$ variables are used. Recent improvements have been made in several directions, i) improving the relaxation tightness [16, 19, 26, 27, 35], ii) examining tightness of relaxations [18, 13], iii) deriving fast specialized solvers for (2) by means of the dual [8, 17, 19, 27], and iv) making precise the relationship between (2) and traditional message passing algorithms [14, 33].

Related work. Recently, higher-order than pairwise potentials have been considered. They are known as "higher-order cliques" [11, 23, 32], or "high-arity interactions" [35]. With exception of the last paper, the potentials considered in these works are of a restricted form or limited to small clique sizes of three or four nodes.

Kohli et al. [11] extend the generalized Potts model for pairwise potentials [4] to higher order interactions. For a clique $C \subseteq V$ of size two or larger, he considers potential functions which are zero if all nodes in the clique are assigned the same label and a constant if otherwise. For $|C| = 2$ the potential function reduces to a pairwise Potts $\mathcal{P}^2$ potential In [12], Kohli et al. use potential functions of this form to ensure label-consistency over large image regions. The image regions are created by multiple unsupervised segmentations of the image and each region becomes a clique with associated high-order potential function, such that homogeneous and large regions receive a large potential if their respective pixel labels are not assigned to the same label. In [11] specialized α-expansion graphcut moves are developed to solve these high-order potentials.

For the general problem of global potential functions, Werner [35] is most close to our work; he discusses global interactions and uses as example a hard potential on the number of nodes labeled with a certain class label. A simple greedy algorithms is used to solve a relaxation. We continue his line of work and derive global constraints to be used in (2) directly from the combinatorial polytope associated with the global interaction.

Segmentation under connectivity constraints has recently been considered by Vicente, Kolmogorov and Rother [31]. They define a "problem C0" which is a binary segmentation task where the subset of nodes labeled as foreground is restricted to form a single connected component. Because Vicente et al. consider this problem too complex to solve, they propose a simplified problem C1, in which only a given pair of nodes must be connected. They prove NP-hardness for these problems. For this restricted problem C1 they propose DijkstraGC, a heuristic based on the graphcut algorithm [4] able to produce good connected segmentations from an unconnected segmentation and user-supplied pairwise connectivity constraints. The DijkstraGC method is not directly applicable in our setting because it does not solve problem C0. Our contribution can be seen to provide a tractable way to solve problem C0.

Zeng et al. [38] incorporate global topology-preserving constraints into the graph cut framework. Given a global user initialization, their algorithm finds a local optimum which respects the initial topology. Impressively, the algorithm is as fast as the popular min-cut algorithm of [4]. Their algorithm considers a global NP-hard potential but only obtains a local minimum; our method instead also uses a NP-hard global potential, but solves a relaxation for the global optimum. Das et al. [5] propose a simple global shape prior which favors compact shapes and can be realized within the normal graph cut energy framework. For this approach to work, the object center needs to be marked by a user; moreover their approach is not rotation invariant.

The potential functions we consider are defined on *all* nodes in the graph, denoted $\psi_V(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w})$. We consider a "connectedness potential", which enforces connectedness of the output labeling with respect to a graph. We derive our algorithm in a principled way using results from polyhedral combinatorics. Although in this work we only consider one global potential function, the overall approach by which we incorporate the function is general and applicable to other higher-order potential functions.

In the following section we formalize connectedness by analyzing the set of all connected MRF labelings. In Section 3 we derive tractable global potential functions. Sec-

tion 4 evaluates the proposed MRF/CRF with connectedness potentials on both a synthetic data set and on the challenging PASCAL VOC 2008 segmentation data set; we conclude in Section 5.

2. Connected Subgraph Polytope

The LP relaxation (2) has variables $\mu_i(y_i) \in \{0, 1\}$ encoding if a node i has label y_i. In this section we derive a polyhedral set which can be intersected with the feasible set of LP (2) such that for all remaining feasible solutions all nodes labeled with the same label form a connected subgraph. This set is the *connected subgraph polytope*, the convex hull of all possible labeling which are connected. We first define this set and then analyze its properties.

Definition 1 (Connected Subgraph Polytope) *Given a simple, connected, undirected graph $G = (V, E)$, consider indicator variables $y_i \in \{0, 1\}$, $i \in V$. Let $C = \{\boldsymbol{y} : G' = (V', E')$ connected, with $V' = \{i : y_i = 1\}, E' = (V' \times V') \cap E\}$ denote the finite set of connected subgraphs of G. Then we call the convex hull $Z = conv(C)$ the* connected subgraph polytope.

The convex hull of a finite set of points is the tightest possible convex relaxation of the set. Furthermore, for the case of minimizing a linear function over the convex hull, it is known from classic linear programming theory [2, 25] that at least one optimal solution exists at a vertex of the polytope. By construction, this solution is then also in C and the relaxation is exact. Unfortunately, optimizing over this polytope is NP-hard, as the following theorem shows. The theorem is identical to Theorem 1 in [31]; we state it here for the reference to the earlier work of Karp [10].

Theorem 1 (Karp, 2002) *It is NP-hard to optimize a linear function over $Z = conv(C)$.*

The problem is known as Maximum-Weight Connected Subgraph Problem and it has been shown to be NP-hard in [9, 10].

Therefore, if we plan to intersect $\text{conv}(C)$ with the feasible set of (2), we are planning to optimize a linear function over this polytope. Unfortunately, from Theorem 1 it follows that optimizing a linear function over $\text{conv}(C)$ is NP-hard, and it is unlikely that $\text{conv}(C)$ has a "simple" description (one which is polynomially separable), see [25, chapter 18]. To overcome this difficulty we will derive a tight relaxation to $\text{conv}(C)$ which is still polynomially solvable.

To do this, we focus on the properties of C and its convex hull Z. We first show that Z has full dimension, *i.e.* does not live in a proper subspace. Second, we show that $y_i \geq 0$ and $y_i \leq 1$ are facet-defining inequalities for all graphs. Figure 1 shows what this means: $d_1^\top y \leq 1$ and $d_2^\top y \leq 1$ are both *valid*, but only $d_3^\top y \leq 1$ is facet-defining [36].

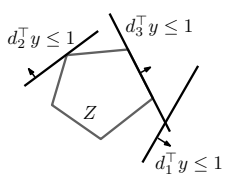

Figure 1. Valid inequalities.

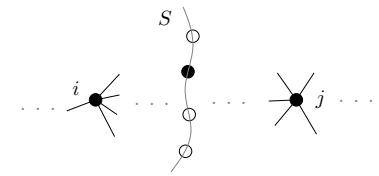

Figure 2. Vertex i and j and one vertex separator set $S \in \bar{\mathcal{S}}(i, j)$.

Lemma 1 *$dim(Z) = |V|$.*

Lemma 2 *For all $i \in V$, the inequalities $y_i \geq 0$ and $y_i \leq 1$ are facet-defining for Z.*

The proofs can be found in the supplementary materials.

Definition 2 (Vertex-Separator Set) *Given a simple, connected, undirected graph $G = (V, E)$, for any pair of vertices $i, j \in V$, $i \neq j$, $(i, j) \notin E$, the set $S \subseteq V \setminus \{i, j\}$ is said to be a* vertex-separator set *with respect to $\{i, j\}$ if the removal of S from G disconnects i and j.*

If the removal of S from G disconnects i and j, then there exist no path between i and j in $G' = (V \setminus S, E \setminus (S \times S))$. As an additional definition, a set $\bar{S}$ is said to be an *essential vertex-separator set* if it is a vertex-separator set with respect to $\{i, j\}$ and any strict subset $T \subset \bar{S}$ is not. Let $\mathcal{S}(i, j) = \{S \subset V : S$ is a vertex-separator set with respect to $\{i, j\}\}$ denote the collection of all vertex-separator sets, and $\bar{\mathcal{S}}(i, j) \subset \mathcal{S}(i, j)$ be the subset of essential vertex-separator sets.

Theorem 2 *C, the set of all connected subgraphs, can be described exactly by the following constraint set.*

$$y_i + y_j - \sum_{k \in S} y_k \leq 1, \forall (i, j) \notin E : \forall S \in \mathcal{S}(i, j), \quad (3)$$

$$y_i \in \{0, 1\}, \qquad i = 1, \ldots, |V|. \quad (4)$$

The proof can be found in the supplementary materials.

Theorem 2 has a simple intuitive interpretation, shown in Figure 2. If two vertices i and j are selected ($y_i = y_j = 1$, shown in black), then any set of vertices which separate them (set S) must contain at least one selected vertex. Otherwise i and j cannot be connected because any path from i to j must pass through at least one vertex in S.

Having characterized the set of all connected subgraphs exactly by means of (3) and (4) it is natural to look at the linear relaxation, replacing (4) by $y_i \in [0; 1], \forall i$. Such a relaxation yields a polytope $P \supseteq Z = \text{conv}(C) \supset C$, which can be a tight, hence good, or loose, hence bad, approximation to $\text{conv}(C)$. The quality of the approximation improves if *facets* of the polytope P are true facets of $\text{conv}(C)$. The following theorem states that in our relaxation a large subset of the constraints (3) – exactly those associated to *essential* vertex-separator sets – are indeed facets of $\text{conv}(C)$.

Theorem 3 *The following linear inequalities are* facet-defining *for* $Z = conv(C)$.

$$y_i + y_j - \sum_{k \in S} y_k \leq 1, \quad \forall (i,j) \notin E : \forall S \in \bar{\mathcal{S}}(i,j). \quad (5)$$

The proof can be found in the supplementary materials.

Let us summarize our progress so far. We have described the set of connected subgraphs and the associated connected subgraph polytope. Furthermore we have shown that a relaxation of the connected subgraph polytope is locally exact in that the set of linear inequalities (5) are true facets of conv(C). However, in general the number of linear inequalities (5) used in our relaxation is exponential in $|V|$.

We now show that optimization over the set defined by (5) is still tractable because finding violated inequalities – the so called *separation problem* – can be solved efficiently using max-flow algorithms.

Theorem 4 (Polynomial-time separation) *For a given point* $\boldsymbol{y} \in [0;1]^{|V|}$ *to find the most violated inequality (5) or prove that no violated inequality exists requires only time polynomial in* $|V|$.

Proof. We give a constructive separation algorithm based on solving a linear max-flow problem on an auxiliary directed graph. For a given point $\boldsymbol{y} \in [0;1]^{|V|}$, consider all $(i,j) \in V \times V$ with $i \neq j$, $(i,j) \notin E$ and $y_i > 0$, $y_j > 0$. For any such (i,j) consider the statement

$$y_i + y_j - \sum_{k \in S} y_k - 1 \leq 0, \qquad \forall S \in \bar{\mathcal{S}}(i,j).$$

Note that in the above statement, the individual variables y are not necessarily binary. We can rewrite the set of inequalities above in equivalent variational form,

$$\max_{S \in \bar{\mathcal{S}}(i,j)} \left(y_i + y_j - \sum_{k \in S} y_k - 1 \right) \leq 0. \quad (6)$$

If we prove that (6) is satisfied, we know that no violated inequalities exists for (i,j). If, however, a violation exist, then the essential vertex-separator set producing the highest violation is given as

$$S^*(i,j) = \text{argmin}_{S \in \bar{\mathcal{S}}(i,j)} \sum_{k \in S} y_k. \quad (7)$$

In order to find this separator set, we transform G into a directed graph G' with edge capacities. In the directed graph each original edge is split into two directed edges with infinite capacity. Additionally each vertex k in the original graph is duplicated and an edge of finite capacity equal to y_k is introduced between the two copies.

Formally, we construct $G' = (V', E')$, $E' \subseteq V' \times V' \times \mathbb{R}$ as follows. Let $V' = V \cup \{k' : k \in V \setminus \{i,j\}\}$. Further let $E' = \{(i,k,\infty) : k \in V, (i,k) \in E\} \cup \{(k',j,\infty) : k \in V, (j,k) \in E\} \cup \{(s',t,\infty), (t',s,\infty) : (s,t) \in E \setminus (\{i,j\} \times \{i,j\})\} \cup \{(k,k',y_k) : k \in V \setminus \{i,j\}\}$. The construction is illustrated for an example graph in Figures 3 and 4. Finding an (i,j)-cut of finite capacity in G' is equivalent to finding an essential (i,j) vertex separator set in G. This can be seen by recognizing that the only edges that can be cut – hence saturated in a maximum flow problem – are the edges (k,k') with finite capacity, which correspond to vertices in the original graph. Solving the max-flow problem in the auxiliary directed graph solves (7). After finding $S^*(i,j)$, we simply check whether (6) is satisfied.

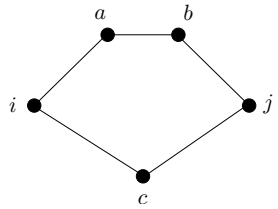

Figure 3. Example graph G. There are three vertex-separator sets in $\mathcal{S}(i,j) = \{\{a,c\}, \{b,c\}, \{a,b,c\}\}$, of which only $\{a,c\}$ and $\{b,c\}$ are *essential*.

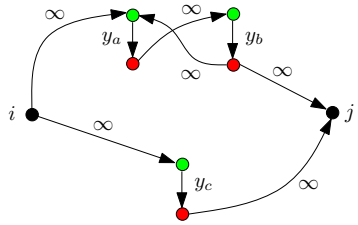

Figure 4. Directed auxiliary graph G' for finding the minimum essential vertex-separator set in G among all sets in $\bar{\mathcal{S}}(i,j)$.

Solving a linear maximum network flow problem is very efficient [4]. The best known algorithms have a computational complexity of $O(|V|^3)$ and $O(|V||E|\log(|V|))$. We need to solve one max-flow problem per (i,j) pair with $y_i > 0$, $y_j > 0$, so the overall separation problem of checking feasibility with respect to (5) can be solved in time $O(|V|^5)$. □

In practice we do not have to check all (i,j) node pairs. Instead, we decompose the graph into connected components such that for all vertices in a connected component there exist an *all*-1-*path* to each other vertex in the component. Only one representative node is chosen at random from each component and the separation is carried out only for the representative vertices. This procedure is exact.

3. Global Potentials

We now transform the connected subgraph polytope into a potential function of a random field.

From the Connected Subgraph Polytope to ψ_V^{conn}. Let $\mu^j(\boldsymbol{y}) = [\mu_1(y_j), \ldots, \mu_{|V|}(y_j)]^\top \in \mathbb{R}^{|V|}$ be the set of vari-

Algorithm 1 MAP-MRF LP Cutting Plane Method

$(\boldsymbol{y}, B) = \text{LPCUTTINGPLANE}(\boldsymbol{x}, \boldsymbol{w})$
Input:
Sample $\boldsymbol{x} \in \mathcal{X}$, weight vector $\boldsymbol{w} \in \mathbb{R}^d$
Output:
Approximate MAP-MRF labeling $\boldsymbol{y}^* \in \mathcal{Y}$
Lower bound on MAP energy $B \in \mathbb{R}$
Algorithm:
$C \leftarrow \mathbb{R}^{\dim(\mathcal{Y})}$, $B \leftarrow -\infty$ {Initially: no cutting planes}
loop
$\boldsymbol{y}^* \leftarrow \text{argmin}_{\boldsymbol{y} \in \mathcal{Y}, \boldsymbol{y} \in C} E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w})$
$\boldsymbol{c} \leftarrow$ most violating constraint (5) with $\boldsymbol{c}^\top \mu^j(\boldsymbol{y}^*) > 1$
if no $\boldsymbol{c}^\top \mu^j(\boldsymbol{y}) > 1$ can be found **then**
break
end if
$C \leftarrow C \cap \{\boldsymbol{y} : \boldsymbol{c}^\top \mu^j(\boldsymbol{y}) \leq 1\}$
end loop
$B \leftarrow E(\boldsymbol{y}^*; \boldsymbol{x}, \boldsymbol{w})$

ables in the LP relaxation (2) indicating assignment to class j over all vertices. One way to enforce connectivity in the LP solution for the vertices assigned to the j'th class is to define the following *hard connectivity potential* function.

$$\psi_V^{\text{hard(j)}}(\boldsymbol{y}) = \begin{cases} 0 & \mu^j(\boldsymbol{y}) \in Z \\ \infty & \text{otherwise} \end{cases} \tag{8}$$

This potential function can be incorporated by adding the respective constraints (5) to the LP relaxation (2). Alternatively we can define a *soft connectivity potential* by defining a feature function measuring the violation of connectivity. We define $\psi_V^{\text{soft(j)}}(\boldsymbol{y}; \boldsymbol{w}) = w_{\text{soft(j)}} \phi^{\text{conn(j)}}(\boldsymbol{y})$ where $\phi^{\text{conn(j)}} \geq 0$ measures the violation of connectivity:

$$\phi^{\text{conn(j)}}(\boldsymbol{y}) = \begin{cases} 0 & \mu^j \in Z \\ \max_{\boldsymbol{d} \in D} \{\boldsymbol{d}^\top \mu^j(\boldsymbol{y}) - 1\} & \text{otherwise} \end{cases},$$

where D is the set of coefficient vectors of the inequalities (5). We can calculate $\max_{\boldsymbol{d} \in D} \{\boldsymbol{d}^\top \mu^j(\boldsymbol{y}) - 1\}$ efficiently by means of Theorem 4. This potential function can be realized by introducing constraints into the LP relaxation as for $\psi^{\text{hard(j)}}$ but also adding one global non-negative slack variable lower bounded by $\phi^{\text{conn(j)}}$ for all $\boldsymbol{y} \in \mathcal{Y}$ and having an objective coefficient of $w_{\text{soft(j)}}$.

LP MAP-MRF with ψ_V. Algorithm 1 iteratively solves the MAP-MRF LP relaxation (2). After each iteration (8) is checked and if the labeling is connected, the algorithm terminates. In the case of an unconnected segmentation, a violated constraint is found and added to the master LP (2).

4. Experiments and Results

We now validate our connectedness potential on two tasks, i) a MRF denoising problem, and ii) object segmentation by learned CRFs.

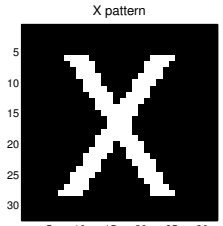

Figure 5. Pattern "X" to be recognized.

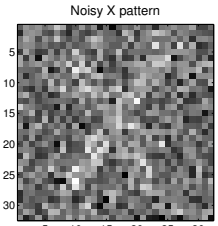

Figure 6. Noisy node potential, $\sigma = 0.9$.

4.1. Motivating Experiment: Denoising

We consider a standard denoising problem [15]. The 32x32 pixel pattern shown in Figure 5 is corrupted with additive Gaussian noise, as shown in Figure 6. The pattern should be recovered by means of solving a binary MRF. We use a 4-neighborhood graph defined on the pixels, and the node potentials are derived from ground truth labeling as

$$\psi_i(\text{"FG"}) = \begin{cases} -1 + \mathcal{N}(0, \sigma) & \text{if } i \text{ is true foreground} \\ 0 & \text{otherwise} \end{cases}$$

$$\psi_i(\text{"BG"}) = \begin{cases} -1 + \mathcal{N}(0, \sigma) & \text{if } i \text{ is true background} \\ 0 & \text{otherwise} \end{cases}$$

The edge potentials are regular [15] and chosen as Potts $\psi_{i,j}(y_i, y_j) = |\mathcal{N}(0, k/\sqrt{d})| I(y_i \neq y_j)$, where $d = 4$ is the average degree of our vertices. The parameters are varied over $\sigma \in \{0, 0.1, \ldots, 1.0\}$, $k \in \{0, 0.5, \ldots, 4\}$ and each run is repeated 30 times. For each of the 30 runs, the potentials are sampled once and we derive three solutions, i) "MRF", the solution to standard binary MRF, ii) "MRFcomp", the largest connected component of the MRF, iii) "CMRF", a binary MRF with additional hard-connectivity potential (8) on the foreground plane.

The results are shown in Figures 7 to 9. They show the connected MRF averaged absolute error over the parameter plane and the relative errors to the standard MRF and component heuristic. The advantage of the connectedness constraint over a standard MRF can be seen by looking at the relative errors in Figure 8. For almost all parameter regimes the error of the MRF is higher (positive values in the plot). Also, from Figure 9 it can be seen that the connectedness constraint outperforms the largest-connected-component heuristic except when very weak edge potentials are used (upper left corner). Typical examples are shown in Figure 10 and 11.

4.2. Experiment: Learning Object Segmentation

Connectivity is a strong global prior for object segmentation. In this experiment we use the connectivity assumption to segment out objects from the background in the PASCAL VOC 2008 data set [6]. The data set is known to be particularly challenging as the images contain objects of 20 different classes with a lot of variability in lighting, viewpoint,

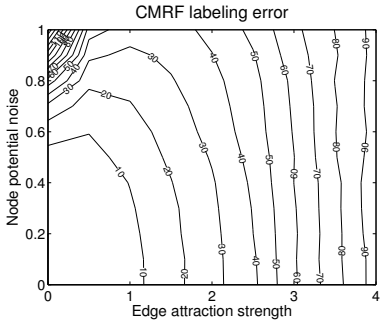

Figure 7. Connected MRF labeling error.

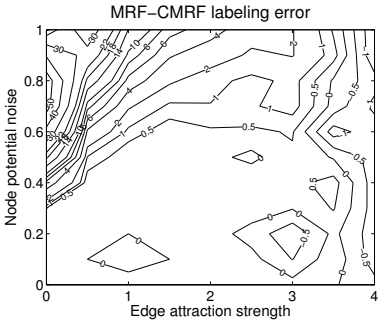

Figure 8. Error difference MRF-CMRF.

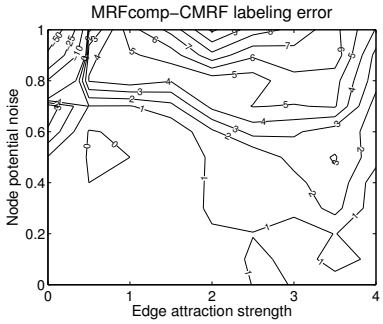

Figure 9. Error diff. MRFcomp-CMRF.

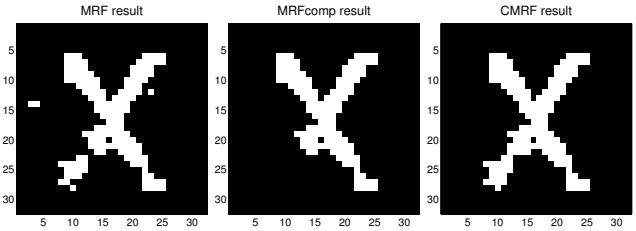

Figure 10. MRF/MRFcomp/CMRF results, with energies $E = -985.61$, $E = -974.16$, $E = -984.21$, and errors 36, 46, 28, respectively. The connectivity constraint solution CMRF is a substantial improvement over the solutions of MRF and MRFcomp.

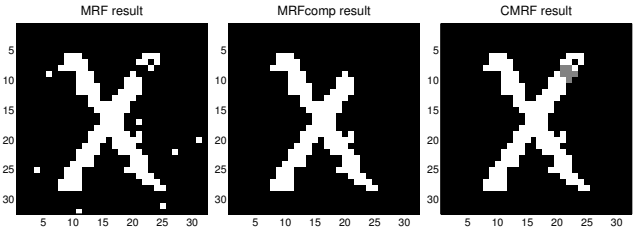

Figure 11. MRF/MRFcomp/CMRF results, with energies $E = -980.13$, $E = -974.03$, $E = -976.83$, and errors 34, 34, 24, respectively. Note although the CMRF solution becomes fractional, it is a substantial improvement over the MRF result.

size and positioning of the objects. We first look at a simple statistic of the training and validation set for the *detection* task: How many objects of each individual class are present on an image? It turns out that in 70% of all images, there is no more than a single object of each class present on the image. (The supplementary materials have a plot.)

Experimental Setup. In our setting, we let $\boldsymbol{x} = (V, E)$ be the graph resulting from a superpixel segmentation [24] of an image, where each $i \in V$ is a superpixel. The superpixel segmentation is obtained using the method[1] of Mori [22], where we use ≈ 100 superpixels. Segmentations are shown on the left side of Figures 12 to 14. Using superpixels has two advantages, i) the information in each superpixel is more discriminative because all image information in the region can be used to describe it, and ii) the complexity of the inference is drastically reduced with only a negligible approximation error. Each superpixel becomes a vertex in the graph. An edge joins two vertices if the superpixels are adjacent in the image. Therefore connectivity in the graph implies connectivity of the segmentation. For each image, we extract 50,000 SURF features [1] at random positions and assign each feature to the superpixel which contains the center pixel of the feature. For each vertex, a bag-of-words histogram $x_i \in \mathbb{R}^H$ is created by nearest-neighbor quantizing the features associated to the superpixel in a codebook of 500 words ($H = 500$), created by k-means clustering on a random sample of 500,000 features from the training set.

We treat each of the twenty classes separately as a binary problem. That is, for each image showing an object of the class, a class-vs-background labeling is sought. Hence each vertex i in the graph has a label vector $y_i \in \{0, 1\} \times \{0, 1\}$. We report the average intersection-union metric, defined as $\frac{TP}{TP+FP+FN}$ ratio, where TP, FP, FN are true positives, false positives and false negatives, respectively, per pixel labeling for the object class [6]. Because the VOC2008 segmentation `trainval` set includes only 1023 images for which ground truth is available, with some classes having as few as 44 positive images (only 19 for `train` alone), we use a three-fold cross validation estimate on the `trainval` set. For all CRF variants we will describe later, we use the following feature functions.

- Node features, $\phi_i^{(1)}(y_i, \boldsymbol{x}) = \text{vec}(x_i y_i^\top)$.

  Thus the output of $\phi_i^{(1)}(y_i, \boldsymbol{x})$ is a $(H, 2)$-matrix of two weighted replications of the node histogram x_i. The matrix is stacked columnwise.

- Edge features $\phi_{i,j}^{(2)}(y_i, y_j, \boldsymbol{x}) = \text{vec}_\Delta(y_i y_j^\top)$.

  This is the upper-triangular part including diagonal of the outer product $y_i y_j^\top$. By making this feature available, the CRF can learn the weights for the inter-class and intra-class Potts potentials separately.

We test three CRFs, i) a CRF with these feature functions, ii) the same CRF with $\psi_V^{\text{hard(class)}}$, and iii) the same CRF with $\psi_V^{\text{soft(class)}}$.

Learning the parameters $\boldsymbol{w}$. For learning the parameters of the model, we use the structured output support vector machine framework [30], recently also used in computer

[1] `http://cs.sfu.ca/~mori/research/superpixels/`

vision [3, 21, 29]. It minimizes the following regularized risk function.

$$\min_{\boldsymbol{w}} \quad \|\boldsymbol{w}\|^2 + \frac{C}{\ell}\sum_{n=1}^{\ell}\max_{\boldsymbol{y}\in\mathcal{Y}}\left(\Delta(\boldsymbol{y}_n,\boldsymbol{y})\right. \tag{9}$$
$$\left. +E(\boldsymbol{y}_n;\boldsymbol{x}_n,\boldsymbol{w}) - E(\boldsymbol{y};\boldsymbol{x}_n,\boldsymbol{w})\right),$$

where $(\boldsymbol{x}_n,\boldsymbol{y}_n)_{n=1,\dots,\ell}$ are the given training samples and $\Delta : \mathcal{Y}\times\mathcal{Y}\to\mathbb{R}_+$ is a compatibility function which has a high value if two segmentations are different and a low value if they are very similar. More precisely, we define $\Delta(\boldsymbol{y}^1,\boldsymbol{y}^2) = \sum_{i\in V}\frac{r_i}{\sum_{j\in V} r_j}\left(y_i^1 + y_i^2 - 2y_i^1y_i^2\right)$, where r_i is the size in pixels of the region i in the superpixel segmentation. Note that this definition is, i) symmetric, $\Delta(\boldsymbol{y}^1,\boldsymbol{y}^2) = \Delta(\boldsymbol{y}^2,\boldsymbol{y}^1)$, ii) zero-based, $\Delta(\boldsymbol{y},\boldsymbol{y}) = 0$, and non-negative, iii) corresponds to the Hamming loss if all elements are binary, and iv) decomposes linearly over the individual elements if one of $\boldsymbol{y}^1$, $\boldsymbol{y}^2$ is constant. Because of the last point it is easy to incorporate into the MRF inference procedure by means of a bias on the node potentials [7, 29]. We train with $C \in \{100, 10, \dots, .00001\}$ and report the highest achieved performance of each model.

The objective (9) is convex, but non-differentiable. Still, it can be solved efficiently by iteratively solving a quadratic program, see [30]. Given a parameter vector $\boldsymbol{w}$, for each sample $(\boldsymbol{x}_n,\boldsymbol{y}_n)$ one needs to solve $\max_{\boldsymbol{y}\in\mathcal{Y}}\ (\Delta(\boldsymbol{y}_n,\boldsymbol{y}) + E(\boldsymbol{y}_n;\boldsymbol{x}_n,\boldsymbol{w}) - E(\boldsymbol{y};\boldsymbol{x}_n,\boldsymbol{w}))$. As the last term is constant and $\Delta(\boldsymbol{y}_n,\boldsymbol{y})$ can be incorporated into $E(\boldsymbol{y};\boldsymbol{x}_n,\boldsymbol{w})$, Algorithm 1 can be used to find the maximizer $\boldsymbol{y}_n^*$. It defines a new constraint and by iterating between generating constraints and solving the QP we can obtain successively better parameter vectors $\boldsymbol{w}$.

Finley and Joachims [7] have shown that if the inference in the learning problem is hard, then *approximately solving* this hard problem can lead to classification functions which do not generalize well. Instead, it is preferable to *solve exactly* a relaxation to the original inference problem. This is precisely what we are doing, because the intersection of (5) with the MAP-MRF LP local polytope defines an exactly solvable relaxation.

Results. Table 1 shows for each class the averaged intersection-union scores of the three different methods.

For most classes the connected CRF models outperform the baseline CRF. This is especially true for classes such as aeroplane and cat, whose images usually contain only one large object. In contrast, classes such as bottle and sheep most often have more than one objects on an image. This is a violation of our connectedness assumption and in this case the CRF model outperforms the connected ones. We also see that in some cases the extra flexibility of the soft connectedness over the hard connectedness prior pays off: for the boat, bus, cow and motorbike classes, the ability to weight the connectivity strength versus the other potentials is useful in improving over both the baseline CRF and the hard connected CRF. The typical behaviour of the hard-connectedness CRF on test images is shown in Figures 12 to 14 for the aeroplane class. In the first two segmentations, connectedness helps by completing a discontinuous segmentation and by removing clutter. Figure 14 shows a hopeless case: if the CRF segmentation is wrong, connectedness cannot help.

5. Conclusions

We have shown how the limitation of only considering local interactions in discrete random field models can be overcome in a principled way. We considered a hard global potential encoding whether a labeling is connected or not. We derived an efficient relaxation that can naturally be used with MAP-MRF LP relaxations. Experimentally, we demonstrated that a connectedness potential reduces the segmentation error on both a synthetic denoising and real object segmentation task.

Clearly, other meaningful global potential functions could be devised by the method introduced in this paper. The principled use of polyhedral combinatorics opens a way to better model high-level vision tasks with random field models. Another direction of future work is to see if the addition of complicated primal constraints like (5) can be accommodated into recent efficient dual LP MAP solvers [8, 17, 19, 27].

All software and experiments used in this paper are available as open-source software at `http://www.kyb.mpg.de/bs/people/nowozin/cmrf/`.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool. Speeded-up robust features (SURF). *CVIU*, 110(3):346–359, 2008.

[2] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. 1997.

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

[4] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004.

[5] P. Das, O. Veksler, V. Zavadsky, and Y. Boykov. Semiautomatic segmentation with compact shape prior. *Image and Vision Computing*, 2008.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/.

[7] T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. In *ICML*, 2008.

[8] A. Globerson and T. Jaakkola. Fixing max-product: Convergent message passing algorithms for map lp-relaxations. In *NIPS*, 2007.

[9] T. Ideker, O. Ozier, B. Schwikowski, and A. F. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. In *ISMB*, 2002.

[10] R. M. Karp. Maximum-weight connected subgraph problem, 2002. `http://www.cytoscape.org/ISMB2002/`.

[11] P. Kohli, M. P. Kumar, and P. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[12] P. Kohli, L. Ladický, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

Method	aerop.	bicyc.	bird	boat	bottle	bus	car	cat	chair	cow	dtable	dog	horse	mbike	person	plant	sheep	sofa	train	tv
CRF	0.355	0.087	0.189	0.261	0.138	0.383	0.194	0.278	0.084	0.225	0.279	0.245	**0.232**	0.239	0.188	0.088	**0.298**	0.214	0.419	0.158
hard	**0.380**	0.091	0.202	0.275	0.115	0.391	0.185	**0.311**	**0.121**	0.236	0.269	0.244	0.209	0.268	0.194	0.075	0.249	0.200	0.393	0.152
soft	0.341	0.090	0.176	**0.288**	0.130	**0.406**	0.165	0.283	0.101	**0.270**	0.294	0.220	0.194	**0.273**	0.184	0.074	0.277	0.209	0.419	0.151

Table 1. Results of the VOC2008 segmentation experiment. Marked bold are the cases where a method outperforms the others.

Figure 12. Image/CRF/CRF+conn. Case where connectedness helps: the local evidence is scattered, enforcing connectedness (right) helps.

Figure 13. Image/CRF/CRF+conn. Connectedness can remove clutter: local evidence (edges on the runway) is overridden.

Figure 14. Image/CRF/CRF+conn. Failure case: the CRF segmentation is bad (middle) connectedness does not help (right).

[13] P. Kohli, A. Shekhovtsov, C. Rother, V. Kolmogorov, and P. H. S. Torr. On partial optimality in multi-label MRFs. In *ICML*, volume 307, pages 480–487, 2008.

[14] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10):1568–1583, 2006.

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

[16] N. Komodakis and N. Paragios. Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles. In *ECCV*, 2008.

[17] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*. IEEE, 2007.

[18] M. P. Kumar, V. Kolmogorov, and P. Torr. An analysis of convex relaxations for MAP estimation. In *NIPS*, 2008.

[19] M. P. Kumar and P. Torr. Efficiently solving convex relaxations for MAP estimation. In *ICML*, 2008.

[20] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[21] Y. Li and D. Huttenlocher. Learning for stereo vision using the structured support vector machine. In *CVPR*, 2008.

[22] G. Mori. Guiding model search using segmentation. In *ICCV*, 2005.

[23] S. Ramalingam, P. Kohli, K. Alahari, and P. Torr. Exact inference in multi-label CRFs with higher order cliques. In *CVPR*, 2008.

[24] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003.

[25] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, New York, 1998.

[26] D. Sontag and T. Jaakkola. New outer bounds on the marginal polytope. In *NIPS*, 2007.

[27] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening LP relaxations for MAP using message passing. In *UAI*, 2008.

[28] C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. In *Introduction to Statistical Relational Learning*, chapter 4. 2007.

[29] M. Szummer, P. Kohli, and D. Hoiem. Learning CRFs using graph cuts. In *ECCV*, 2008.

[30] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, Sept. 2005.

[31] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008.

[32] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches. *IEEE Trans. Information Theory*, 51(11):3697–3717, Nov. 2005.

[33] Y. Weiss, C. Yanover, and T. Meltzer. Map estimation, linear programming and belief propagation with convex free energies. In *UAI*, 2007.

[34] T. Werner. A linear programming approach to max-sum problem: A review. Research report, Center for Machine Perception, Czech Technical University, Dec. 2005.

[35] T. Werner. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF). In *CVPR*, 2008.

[36] L. A. Wolsey. *Integer Programming*. John Wiley & Sons, New York, 1998.

[37] C. Yanover, T. Meltzer, and Y. Weiss. Linear programming relaxations and belief propagation - an empirical study. *JMLR*, 7:1887–1907, 2006.

[38] Y. Zeng, D. Samaras, W. Chen, and Q. Peng. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in n-d images. *CVIU*, (112):81–90, 2008.