SIAM J. IMAGING SCIENCES
Vol. 3, No. 4, pp. 1048–1074

Global Interactions in Random Field Models: A Potential Function Ensuring Connectedness*

Sebastian Nowozin† and Christoph H. Lampert‡

Abstract. Markov random field (MRF) models, including conditional random field models, are popular in computer vision. However, in order to be computationally tractable, they are limited to incorporating 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 forces the output labeling to be connected and that can naturally be used in the framework of recent maximum a posteriori (MAP)-MRF linear program (LP) relaxations. Using techniques from polyhedral combinatorics, we show that a provably strong 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 an 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 image segmentation task of the PASCAL Visual Object Classes 2008 data set. We show that in both cases the addition of a connectedness prior significantly reduces the segmentation error.

Key words. Markov random fields, potential functions, large cliques, high-arity interactions

AMS subject classifications. 90C57, 90C35, 90C27

DOI. 10.1137/090752614

1. Introduction.

We consider a discrete conditional random field (CRF) [32, 44] representing a probability distribution $p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})$ over a finite 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,

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

where $E(\boldsymbol{y};\boldsymbol{x},\boldsymbol{w})$ is an *energy function* and $Z(\boldsymbol{x},\boldsymbol{w}) = \sum_{\boldsymbol{y}\in\mathcal{Y}} \exp\left(-E(\boldsymbol{y};\boldsymbol{x},\boldsymbol{w})\right)$ is a normalization constant known as a *partition function* [15, 33, 5, 24]. The energy function is representable in terms of the graph structure of the random field as a sum over *potential*

*Received by the editors March 13, 2009; accepted for publication (in revised form) December 7, 2009; published electronically December 21, 2010. A preliminary version of this paper appeared in [37]. This work was funded in part by the EU CLASS project, IST 027978. This work was also supported in part by the IST Programme of the European Community under the PASCAL Network of Excellence, IST-2002-506778. This publication reflects only the authors' views.

http://www.siam.org/journals/siims/3-4/75261.html

†Max Planck Institute for Biological Cybernetics, 72076 Tübingen, Germany. Current address: Microsoft Research, Roger Needham Building, 7 J J Thomson Ave, Cambridge CB3 0FB, United Kingdom (Sebastian.Nowozin@microsoft.com).

‡Max Planck Institute for Biological Cybernetics, 72076 Tübingen, Germany. Current address: Institute of Science and Technology (IST) Austria, Am Campus 1, A-3400 Klosterneuburg, Austria (chl@ist.ac.at).

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.,

$$E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w}) = \sum_{c \in \mathcal{C}} \psi_c(\boldsymbol{y}_c; \boldsymbol{x}, \boldsymbol{w}). \tag{1.1}$$

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 [44]—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 (1.1) can be written as follows:

$$E(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w}) = \sum_{i \in V} \boldsymbol{w}_1^\top \phi_i^{(1)}(y_i, \boldsymbol{x}) + \sum_{(i,j) \in V \times V} \boldsymbol{w}_2^\top \phi_{i,j}^{(2)}(y_i, y_j, \boldsymbol{x}) + \cdots + \boldsymbol{w}_{|V|}^\top \phi_V^{(|V|)}(\boldsymbol{y}, \boldsymbol{x}). \tag{1.2}$$

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 [7] to minimize (1.2), the so-called *maximum a posteriori* (MAP) problem.

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

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

In general, solving for the MAP state is NP-hard, even for the case of binary states. For this reason, two classes of *approximate* inference approaches are popular: first, to give up on global optimality and to solve for the MAP state only approximately by iteratively improving a candidate solution, and second, to *relax* the problem but to solve this modified problem exactly. In this work we take the latter route and extend the so-called *linear programming relaxation* for the MAP-MRF (Markov random field) problem.

Linear programming relaxation. Recently, linear programming relaxations have been rediscovered [48, 50, 53] 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 an 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 ILP, 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 $y_j \in \mathcal{Y}_j$:

$$\min_{\boldsymbol{\mu}} \quad \sum_{i \in V} \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) \left(\boldsymbol{w}_1^\top \phi_i^{(1)}(y_i, \boldsymbol{x}) \right) + \sum_{\substack{(i,j) \\ \in E}} \sum_{\substack{(y_i, y_j) \\ \in \mathcal{Y}_i \times \mathcal{Y}_j}} \mu_{i,j}(y_i, y_j) \left(\boldsymbol{w}_2^\top \phi_{i,j}^{(2)}(y_i, y_j, \boldsymbol{x}) \right) \tag{1.3}$$

$$\begin{aligned} \text{subject to (s.t.)} \quad & \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) = 1 \quad \forall\, i \in V, \\ & \sum_{y_j \in \mathcal{Y}_j} \mu_{i,j}(y_i, y_j) = \mu_i(y_i) \quad \forall\, (i,j) \in E,\ \forall\, y_i \in \mathcal{Y}_i, \\ & \mu_i(y_i) \in \{0, 1\} \quad \forall\, i \in V, \quad \forall\, y_i \in \mathcal{Y}_i, \\ & \mu_{i,j}(y_i, y_j) \in \{0, 1\} \quad \forall\, (i,j) \in E, \quad \forall\, (y_i, y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j. \end{aligned}$$

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 (1.3) the labeling $\boldsymbol{y}^*$ is obtained by setting $y_i \leftarrow \operatorname{argmax}_{y_i \in \mathcal{Y}_i} \mu_i(y_i)$ for all $i \in V$.

The integer program (1.3) is exact but NP-hard. The corresponding *LP relaxation* is obtained by relaxing the last two sets of constraints to the range $[0; 1]$. The LP relaxation has been analyzed extensively [48, 49, 50]. Although linear programming is among the best developed numerical disciplines [4], the primal LP (1.3) is practically restricted to graphs with less than a hundred thousand nodes and with tens of node labels, because on the order of $O(|E|(\max_{i \in V} |\mathcal{Y}_i|)^2)$ variables are used. Recent improvements have been made in several directions: (i) improving the relaxation tightness [27, 31, 42, 43, 51], (ii) examining tightness of relaxations [30, 23], (iii) deriving fast specialized solvers for (1.3) by means of the dual [16, 31, 43, 29], and (iv) making precise the relationship between (1.3) and traditional message passing algorithms [25, 49].

1.1. Related work. Recently, higher-order than pairwise potentials have been considered. They are known as "higher-order cliques" [21, 38, 48] or "high-arity interactions" [51]. With the 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. In this work we consider a high-order potential not limited to a small number of nodes but restricted to a special functional form.

Kohli, Kumar, and Torr [21] extend the generalized Potts model for pairwise potentials [8] to higher-order interactions. For a clique $C \subseteq V$ of size two or larger, they consider the potential function

$$\psi_C(\boldsymbol{\mu}_C) = \begin{cases} \gamma_k & \text{if all vertices } i \in C \text{ are assigned label } k, \\ \gamma_{\max} & \text{otherwise.} \end{cases} \tag{1.4}$$

The constants γ_k, $\gamma_{\max}$ must satisfy $\gamma_{\max} > \gamma_k$ for all labels k. For $|C| = 2$ the potential function reduces to a pairwise Potts $\mathcal{P}^2$ potential [8]. In case $\gamma_k = 0$ for all classes k, the potential is a *metric potential* [21].

In [22], Kohli, Ladický, and Torr 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. For each image region a potential function of the form (1.4) is used, and γ_k, $\gamma_{\max}$ are set to constants derived from the image such that homogeneous and large regions receive a large potential if their respective pixel labels are not assigned to the same label. In [21] specialized α-expansion moves [8] are developed to solve these high-order potentials. Recently, sparse higher-order potentials in which only a small subset of the feasible configurations have a value distinct from a default value have been considered by Rother et al. [40] and Komodakis and Paragios [28]. The latter paper also introduces a general and efficient method for higher-order potentials, but their algorithm cannot be applied to the potential functions considered here because it requires all configurations to be feasible.

For the general problem of global potential functions, Werner [51] is closest to the spirit of our work; he discusses global interactions and uses as an example a hard potential on the number of nodes labeled with a certain class label. A greedy algorithm is used to solve a relaxation of the potential. We continue his line of work and derive global constraints to be used in (1.3) directly from the combinatorial polytope associated with the global interaction.

Segmentation under connectivity constraints has recently been considered by Vicente, Kolmogorov, and Rother [47]. 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 the authors 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 both problems. For this restricted problem C1 they propose DijkstraGC, a heuristic based on the graph cut algorithm [7], which is 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. [54] incorporate global topology-preserving constraints into the graph cut framework. Given a global user initialization, their algorithm finds a local optimum that respects the initial topology. Impressively, the algorithm is as fast as the popular min-cut algorithm of [7]. Their algorithm considers a global NP-hard potential, but obtains only a local minimum; our method instead also uses an NP-hard global potential, but solves a relaxation for the global optimum. Das et al. [11] propose a simple global shape prior which favors compact shapes and can be realized within the normal graph cut energy framework. For their approach to work, the object center needs to be marked by a user; additionally, 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 consider only one global potential function, the overall approach by which we incorporate the function is general and applicable to other higher-order potential functions with suitable polyhedral structure.

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, and in section 4

and section 5 we evaluate the proposed MRF/CRF with connectedness potentials on both a synthetic data set and on the challenging PASCAL Visual Object Classes (VOC) 2008 segmentation data set. We conclude in section 6.

This work is an extended version of our earlier work [37].

2. Connected subgraph polytope.

The LP relaxation (1.3) 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 (1.3) 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 labelings that are connected. We first define this set and then analyze its properties.

Definition 2.1 (connected subgraph polytope). *Given a simple, connected, undirected graph $G = (V, E)$, consider indicator variables $y_i \in \{0,1\}$, $i \in V$. Then 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 $Z = \operatorname{conv}(C)$ 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 [4, 41] 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 [47]; we state it here for the reference to the earlier work of Karp [20].

Theorem 2.2 (see Karp [20]). *It is NP-hard to optimize a linear function over $Z = \operatorname{conv}(C)$.*

The proof can be found in [18, 20], where the problem appears under the name "Maximum-Weight Connected Subgraph Problem."

Therefore, if we plan to intersect $\operatorname{conv}(C)$ with the feasible set of (1.3), we are planning to optimize a linear function over this polytope. Unfortunately, from Theorem 2.2 it follows that optimizing a linear function over $\operatorname{conv}(C)$ is NP-hard, and it is unlikely that $\operatorname{conv}(C)$ has a "simple" description (one which is polynomially separable); see [41, Chapter 18]. To overcome this difficulty we will derive a strong relaxation to $\operatorname{conv}(C)$ which is still polynomially solvable.

To do this, we focus on the properties of C and the polyhedral structure of its convex hull Z. We first show that Z has full dimension, i.e., does not live in a proper subspace of $\mathbb{R}^{|V|}$. 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 [52]. Therefore, the polytope is fully contained in the $|V|$-dimensional hypercube and touches all sides of the hypercube.

Lemma 2.3. $\dim(Z) = |V|$.

Lemma 2.4. *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 Appendix A. For a better characterization of the connected subgraph polytope we need to define *vertex-separator sets*, as follows.

Definition 2.5 (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.*

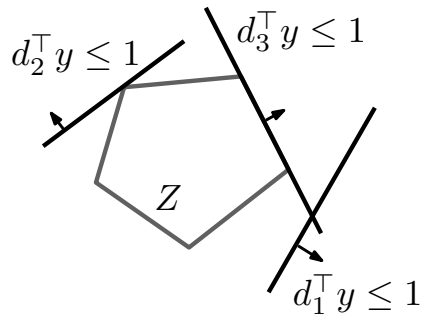

Figure 1. *Three valid inequalities, only one of which is facet-defining.*

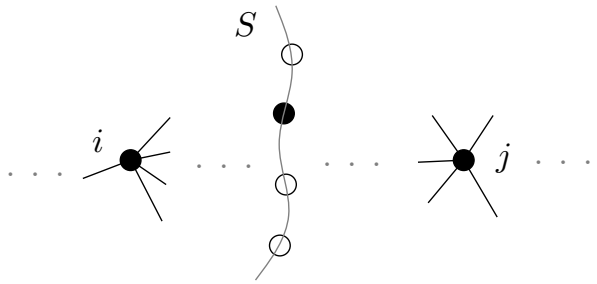

Figure 2. *Vertex i and j and one vertex separator set $S \in \bar{\mathcal{S}}(i,j)$.*

If the removal of S from G disconnects i and j, then there exists 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 $\mathcal{S}(i,j) = \{S \subset V : S \text{ is a vertex-separator set with respect to } \{i,j\}\}$ denote the collection of all vertex-separator sets, and let $\bar{\mathcal{S}}(i,j) \subset \mathcal{S}(i,j)$ be the subset of essential vertex-separator sets.

Theorem 2.6. *C, the set of all connected subgraphs, can be described exactly by the following constraint set:*

$$y_i + y_j - \sum_{k \in S} y_k \le 1 \quad \forall (i,j) \notin E,\ \forall S \in \mathcal{S}(i,j), \tag{2.1}$$

$$y_i \in \{0,1\}, \quad i = 1, \dots, |V|. \tag{2.2}$$

The proof can be found in Appendix A.

Theorem 2.6 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 S of vertices separating them 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 (2.1) and (2.2), it is natural to look at the linear relaxation, replacing (2.2) by $y_i \in [0;1]$ for all i. Such a relaxation yields a polytope $P \supseteq Z = \text{conv}(C) \supset C$, which can be a strong, hence good, or loose, hence bad, approximation to conv(C). The quality of the approximation improves if *facets* of the polytope P are true facets of conv(C). The following theorem states that in

our relaxation a large subset of the constraints (2.1)—exactly those associated to *essential* vertex-separator sets—are indeed facets of $\mathrm{conv}(C)$.

Theorem 2.7. *The following linear inequalities are* facet-defining *for* $Z = \mathrm{conv}(C)$:

$$y_i + y_j - \sum_{k \in S} y_k \leq 1, \quad \forall (i,j) \notin E,\ \forall S \in \bar{\mathcal{S}}(i,j). \tag{2.3}$$

The proof can be found in Appendix A.

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 (2.3) are true facets of $\mathrm{conv}(C)$. However, in general the number of linear inequalities (2.3) used in our relaxation is exponential in $|V|$.

We now show that optimization over the set defined by (2.3) is still tractable because finding violated inequalities—the so called *separation problem*—can be solved efficiently using max-flow algorithms.

Theorem 2.8 (polynomial-time separation). *For a given point* $\boldsymbol{y} \in [0;1]^{|V|}$, *finding the most violated inequality* (2.3) *or proving 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,

$$\max_{S \in \bar{\mathcal{S}}(i,j)} \left(y_i + y_j - \sum_{k \in S} y_k - 1 \right) \leq 0. \tag{2.4}$$

If we prove that (2.4) is satisfied, we know that no violated inequalities exist for (i,j). If, however, a violation exists, then the essential vertex-separator set producing the highest violation is given as

$$S^*(i,j) = \operatorname*{argmin}_{S \in \bar{\mathcal{S}}(i,j)} \sum_{k \in S} y_k. \tag{2.5}$$

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 $E' = \{(i,k,\infty) : k \in V, (i,k) \in E\} \cup \{(k',j,\infty) : k \in V, (j,k) \in$

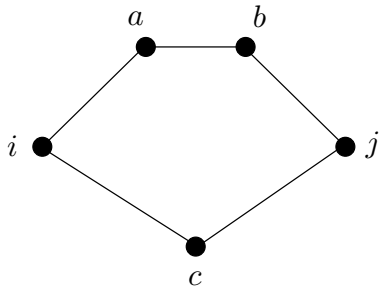

Figure 3. *Example graph G. There are three vertex-separator sets in $\mathcal{S}(i,j) = \{\{a,c\},\{b,c\},\{a,b,c\}\}$, of which only $\{a,c\}$ and $\{b,c\}$ are* essential.

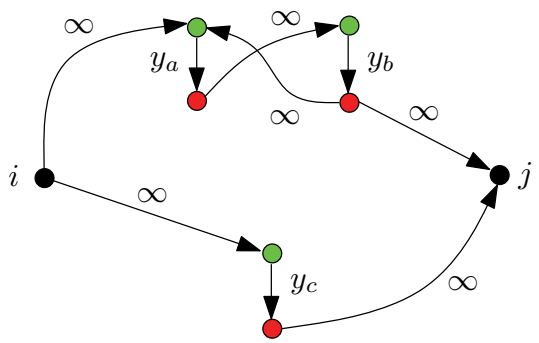

Figure 4. *Directed auxiliary graph G' for finding the minimum essential vertex-separator set in G among all sets in $\tilde{\mathcal{S}}(i,j)$.*

$E\} \cup \{(s',t,\infty),(t',s,\infty) : (s,t) \in E \setminus (\{i,j\} \times \{i,j\})\} \cup \{(k,k',y_k) : k \in V \setminus \{i,j\}\}$. 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 max-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 (2.5). After finding $S^*(i,j)$, we simply check whether (2.4) is satisfied.

Solving a linear maximum network flow problem is very efficient [7]. The best algorithms known 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 that the overall separation problem of checking feasibility with respect to (2.3) 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 exists an *all*-1-*path* to every other vertex in the component. These connected components can be found in linear time using a disjoint set union-rank data structure [10]. 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.

Note that the structure of the separation problem (2.4) remains fixed and the solution thus depends only on the coefficients y_k. Therefore, one could also exploit warm-starting dynamic network flow algorithms to quickly resolve (2.4) for new y_k coefficients.

Integrality of the solution. Both the polytope defined by the MRF LP relaxation and our relaxation of the connected subgraph polytope are not exact: a relaxation is a superset of the true feasible set. This property allows tractable optimization of otherwise NP-hard problems. If the optimal solution over the relaxed feasible set is integral, that is, if the solution is $0, 1$-valued, then the relaxation is locally exact and the solution is globally optimal also over the true feasible set.

On the other hand, if the solution has fractional elements $0 < v < 1$, then the solution is outside the true feasible set and the achieved objective of the relaxation provides a lower bound on the true optimal objective. In this case, a popular method for dealing with fractional solutions is to use rounding to construct a feasible solution from the fraction solution.

Our construction to enforce high-order potentials by intersecting a polytope with the MRF LP relaxation is exact if restricted to the set of integral solutions. But in order to obtain a tractable optimization problem, we do not enforce integrality and instead solve the relaxed LP. Then our approach provides only the solution to the relaxation, which may have fractional elements.

Because we started with two relaxations it seems natural that when intersecting their feasible sets we also obtain a relaxation. In general, however, even if we would have started with the exact marginal polytope with only integral vertices, and another integral polytope, then their intersection could have fractional vertices and therefore provide only a relaxation [41]. In Appendix B, we elaborate further on this point by means of a simple example.

3. From the connected subgraph polytope to ψ_V^{conn}. We now transform the connected subgraph polytope into a potential function of a random field. We let $\mu^j(\boldsymbol{y}) = [\mu_1(y_j), \ldots, \mu_{|V|}(y_j)]^\top \in \mathbb{R}^{|V|}$ be the set of variables in the LP relaxation (1.3) indicating assignment to class j over all vertices. One way to enforce connectivity in the LP solution for the vertices assigned to the jth class is to define the following *hard-connectivity potential* function:

$$\psi_V^{\mathrm{hard}(j)}(\boldsymbol{y}) = \begin{cases} 0, & \mu^j(\boldsymbol{y}) \in Z, \\ \infty & \text{otherwise.} \end{cases} \tag{3.1}$$

This potential function can be incorporated by adding the respective constraints (2.3) to the LP relaxation (1.3). Geometrically, this intersects the connected subgraph polytope relaxation with a subspace of the feasible set of the LP (1.3). By applying this potential function to different labels y_j, this naturally applies also for multilabel MRFs.

Alternatively, we can define a *soft connectivity potential* by defining a feature function measuring the violation of connectivity. We define $\psi_V^{\mathrm{soft}(j)}(\boldsymbol{y}; \boldsymbol{w}) = w_{\mathrm{soft}(j)} \phi^{\mathrm{conn}(j)}(\boldsymbol{y})$ where $\phi^{\mathrm{conn}(j)} \geq 0$ measures the violation of connectivity:

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

where D is the set of coefficient vectors of the inequalities (2.3). We can calculate the violation $\max_{\boldsymbol{d} \in D} \{\boldsymbol{d}^\top \mu^j(\boldsymbol{y}) - 1\}$ efficiently by means of Theorem 2.8. This potential function can be realized by introducing constraints into the LP relaxation as for $\psi^{\mathrm{hard}(j)}$ but also adding one global nonnegative slack variable lower bounded by $\phi^{\mathrm{conn}(j)}$ for all $\boldsymbol{y} \in \mathcal{Y}$ and having an objective coefficient of $w_{\mathrm{soft}(j)}$.

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
$\mu \leftarrow \operatorname{argmin}_{\boldsymbol{y}\in\mathcal{Y}, \boldsymbol{y}\in C} E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w})$
$\boldsymbol{c} \leftarrow$ most violating constraint (2.3) with $\boldsymbol{c}^\top \mu^j > 1$
if no $\boldsymbol{c}^\top \mu^j > 1$ can be found **then**
break
end if
$C \leftarrow C \cap \{\boldsymbol{y} : \boldsymbol{c}^\top \mu^j \le 1\}$
end loop
$B \leftarrow E(\boldsymbol{y}(\mu); \boldsymbol{x}, \boldsymbol{w})$

3.1. LP MAP-MRF with ψ_V. Algorithm 1 iteratively solves the MAP-MRF LP relaxation (1.3). After each iteration (3.1) 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 (1.3).

The algorithm is finitely convergent because the number of facet-defining inequalities is finite. The equivalence of optimization and separation guarantees polynomial-time solvability of the MAP-MRF LP relaxation with the connected subgraph polytope relaxation [41, section 14.2].

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

4. Experiment: Denoising. We consider a standard denoising problem [26]. The 32×32 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 [26] 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

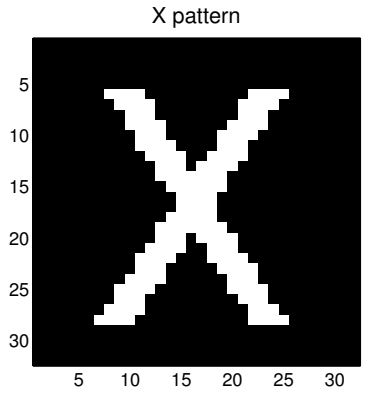

Figure 5. *Pattern "X" to be recognized.*

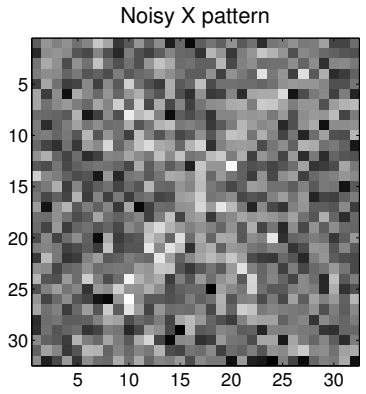

Figure 6. *Noisy node potential, $\sigma = 0.9$.*

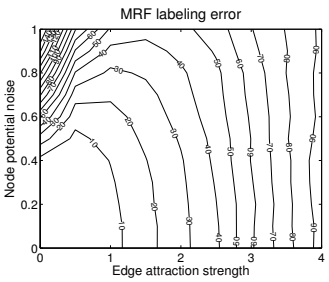

Figure 7. *MRF labeling error.*

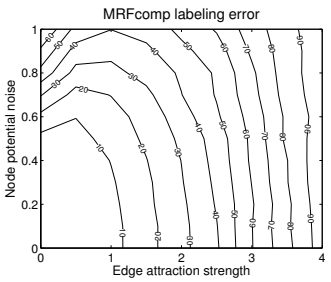

Figure 8. *MRFcomp labeling error.*

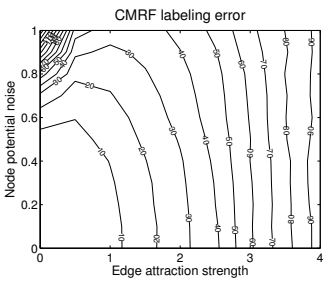

Figure 9. *Connected MRF labeling error.*

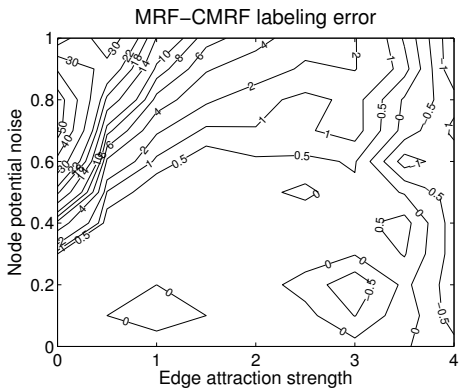

Figure 10. *Error diff. MRF-CMRF.*

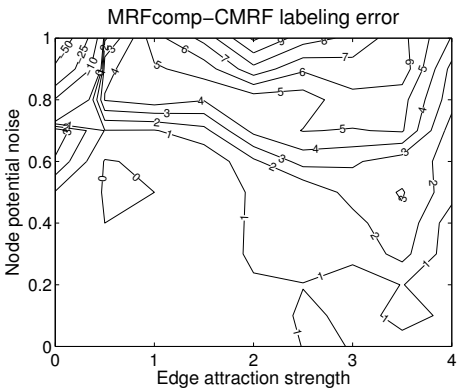

Figure 11. *Error diff. MRFcomp-CMRF.*

to standard binary MRF, (ii) "MRFcomp," the largest connected component of the MRF, and (iii) "CMRF," a binary MRF with additional hard-connectivity potential (3.1) on the foreground plane.

The results are shown in Figures 7 to 11. They show the connected MRF averaged absolute error over the parameter plane and the errors relative to the standard MRF and component

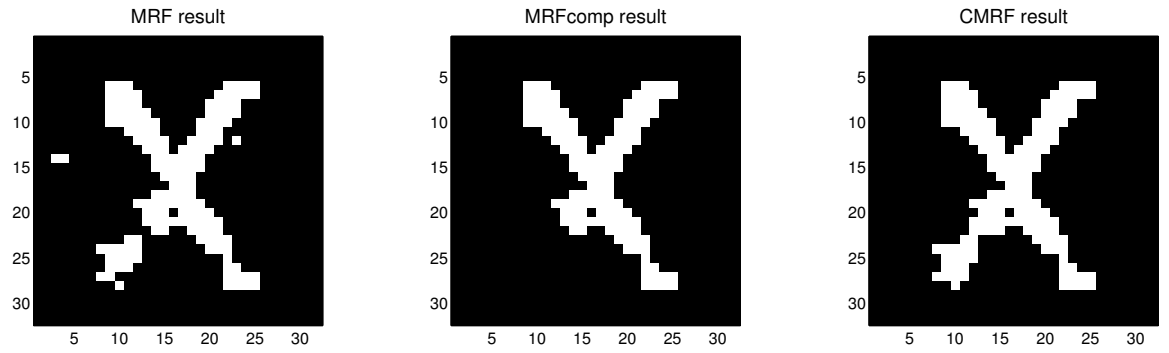

Figure 12. *MRF/MRFcomp/CMRF results, with energies $E = -985.61$, $E = -974.16$, $E = -984.21$, and errors 36, 46, 28, respectively. The connectivity constraint solution CMRF is a substantial improvement over the solutions of MRF and MRFcomp.*

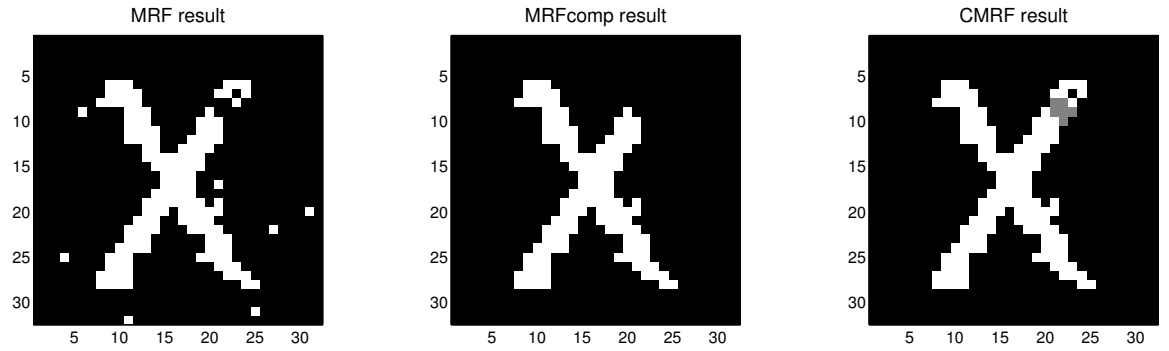

Figure 13. *MRF/MRFcomp/CMRF results, with energies $E = -980.13$, $E = -974.03$, $E = -976.83$, and errors 34, 34, 24, respectively. Note that although the CMRF solution becomes fractional, it is a substantial improvement over the MRF and MRFcomp results.*

heuristic. The advantage of the connectedness constraint over a standard MRF can be seen by looking at the relative errors in Figure 10. For almost all parameter regimes the error of the MRF is higher (positive values in the plot). Also, from Figure 11 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 Figures 12 and 13.

4.1. Integrality. Because we use relaxations for both the marginal polytope (the LP relaxation) and the connected subgraph polytope (the relaxation described by (2.3)), it is not a priori clear that the solution obtained will be integral. Only if it is, we do have a solution to the true, unrelaxed problem. If it is fractional, the solution is still optimal in the relaxation, but outside the true feasible set.

In Figure 14 we show the integrality, i.e., the fraction of variables which are integral. We see that our approach is very effective: for medium noise and edge interactions, the solution is always integral, whereas even when there is more noise and edge interaction, very few variables—less than 0.5% for most configurations—become fractional.

The problems defined by the marginal polytope and the connected subgraph polytope are

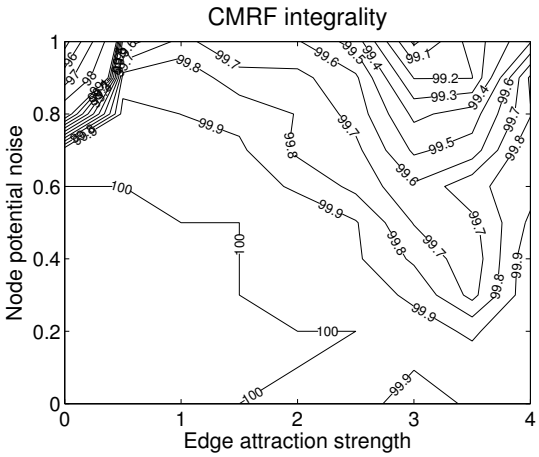

Figure 14. *Mean solution integrality of the MRF with hard-connectivity potential over* 30 *runs for varying problem parameters.*

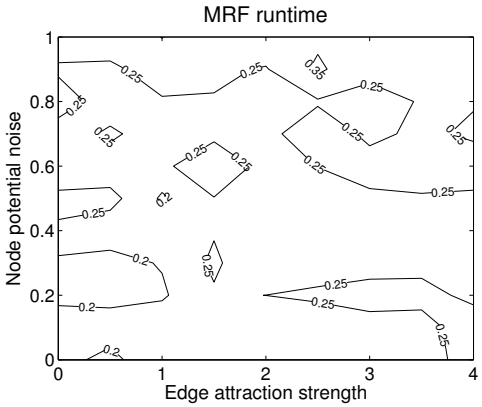

Figure 15. *MRF runtime in seconds.*

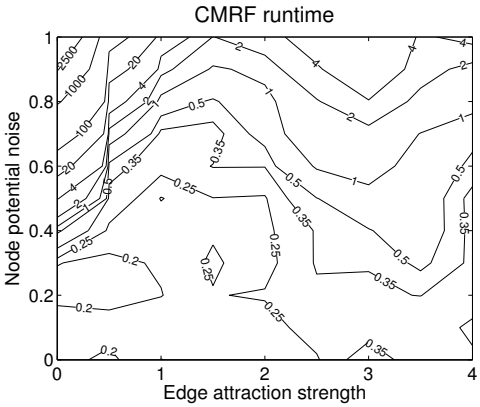

Figure 16. *CMRF runtime in seconds.*

both NP-hard. Hence, it is likely that no polynomial-time approach can provide the guaranteed optimum. In theory, a logical step within our approach would be to prove properties of the fractional solutions, for example, that they satisfy half-integrality or can be rounded with optimality guarantees in order to obtain a polynomial-time approximation algorithm. In practice, the approach already works very well.

4.2. Runtime analysis. We compare the runtime overhead of the connectivity constraint on the MAP-MRF inference in Figures 15 and 16.

The first observation is that the LP relaxation (MRF) is solved in time independent of the edge attraction strength and node potential noise. For the connected CMRF relaxation, this is not the case: the runtime increases with node potential noise and edge attraction strength. It remains efficient, with a runtime within a small factor of the MRF relaxation runtime for large parts of the parameter space.

One exception to this is the high node potential noise region with low edge attraction strength, shown in the upper-left of Figure 16. There, many pixels around the image prefer

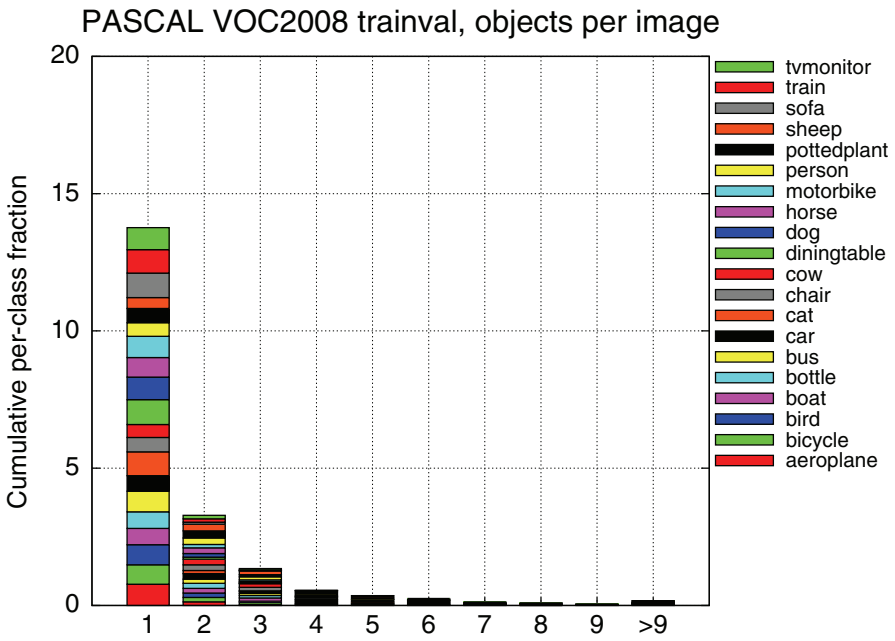

Figure 17. *Number of objects of individual classes per image in the PASCAL VOC* 2008 `trainval` *data set for the object detection task.*

to be labeled positive. The connectivity potential then connects these pixels, requiring many outer iterations in Algorithm 1, and a large number of constraints are generated. The solution obtained is of bad quality, as most pixels being connected are actually in the background. This same effect causes the bad performance of the CMRF method in this regime and can also be seen in the top left part of Figure 11.

5. 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 [12]. 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, 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 in an image? Figure 17 shows the number of objects of individual classes per image in the PASCAL VOC 2008 `trainval` data set. The statistics confirms that if an object is present in an image, in 70% of the cases there is no other object of the same class in the image. For some classes, like `aeroplane`, `cat`, and `diningtable`, this is more often the case than for classes like `bottle`, `chair`, `person`, and `sheep`.

5.1. Experimental setup. In our setting, we let $\boldsymbol{x} = (V, E)$ be the graph resulting from a superpixel segmentation [39] of an image, where each $i \in V$ is a superpixel. The superpixel segmentation is obtained using the method[1] of Mori [36], where we use ≈ 100 superpixels.

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

Example segmentations are shown in the left-hand images of Figures 18 to 20. Using superpixels has three advantages: (i) the information in each superpixel is more discriminative because all image information in the region can be used to describe it, (ii) the complexity of the inference is drastically reduced with only a negligible approximation error, and (iii) the notion of connectivity becomes more meaningful if larger, equal-sized parts are considered.

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. We prefer the normalized-cuts based superpixels over superpixels generated by using mean-shift clustering [9] and spanning-tree heuristics [13] because the normalized-cuts superpixels are approximately of the same size and each superpixel has a similar number of neighboring superpixels. This behavior fits our notion of connectivity and prevents a single superpixel from being connected to a large number of other superpixels.

For each image, we extract 50,000 speeded-up robust features features [2] 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 20 classes separately as a binary problem. That is, for each image showing an object of the class, a class-versus-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 the $\frac{TP}{TP+FP+FN}$ ratio, where TP, FP, and FN are true positives, false positives, and false negatives, respectively, per pixel labeling for the object class [12]. Because the VOC 2008 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 described later, we use the following feature functions.

- Node features, $\phi_i^{(1)}(y_i, \boldsymbol{x}) = \mathrm{vec}(x_i y_i^\top)$.
  Thus the output of $\phi_i^{(1)}(y_i, \boldsymbol{x})$ is an $(H, 2)$-matrix of two weighted replications of the node histogram x_i. The matrix is stacked columnwise.
- Edge features, $\phi_{i,j}^{(2)}(y_i, y_j, \boldsymbol{x}) = \mathrm{vec}_\Delta(y_i y_j^\top)$.
  This is the upper-triangular part including the diagonal of the outer product $y_i y_j^\top$. By making this feature available, the CRF can learn the weights for the interclass and intraclass Potts potentials separately.

We test three CRFs: (i) a CRF with these feature functions, (ii) the same CRF with $\psi_V^{\mathrm{hard(class)}}$, and (iii) the same CRF with $\psi_V^{\mathrm{soft(class)}}$. All three models are trained using the structured support vector machine (SVM) algorithm, and all models have access to exactly the same features.

5.2. Learning the parameters $\boldsymbol{w}$. For learning the parameters of the model, we use the structured SVM framework [46], recently also used in computer vision [6, 34, 45]. It minimizes the following regularized risk function:

$$\min_{\boldsymbol{w}} \quad \|\boldsymbol{w}\|^2 + \frac{C}{\ell} \sum_{n=1}^{\ell} \max_{\boldsymbol{y} \in \mathcal{Y}} \left(\Delta(\boldsymbol{y}_n, \boldsymbol{y}) + E(\boldsymbol{y}_n; \boldsymbol{x}_n, \boldsymbol{w}) - E(\boldsymbol{y}; \boldsymbol{x}_n, \boldsymbol{w}) \right), \tag{5.1}$$

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 $\Delta(\boldsymbol{y}^1, \boldsymbol{y}^2) = \sum_{i \in V} \frac{r_i}{\sum_{j \in V} r_j} \left(y_i^1 + y_i^2 - 2 y_i^1 y_i^2\right)$, where r_i is the size in pixels of the region i in the superpixel segmentation.

Note that this definition (i) is symmetric, $\Delta(\boldsymbol{y}^1, \boldsymbol{y}^2) = \Delta(\boldsymbol{y}^2, \boldsymbol{y}^1)$, (ii) is zero-based, $\Delta(\boldsymbol{y}, \boldsymbol{y}) = 0$, and nonnegative, (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 [14, 45]. We train with $C \in \{.00001, .0001, \dots, 10, 100\}$ and report the highest achieved performance of each model.

The objective (5.1) is convex, but nondifferentiable. Still, it can be solved efficiently by iteratively solving the following quadratic program (see [46, 19]):

$$\min_{\boldsymbol{w}, \boldsymbol{\xi}} \quad \|\boldsymbol{w}\|^2 + \frac{C}{\ell} \sum_{n=1}^{\ell} \xi_n \tag{5.2}$$

$$\text{s.t.} \quad E(\boldsymbol{y}_n; \boldsymbol{x}_n, \boldsymbol{w}) + \Delta(\boldsymbol{y}_n, \boldsymbol{y}) \le E(\boldsymbol{y}; \boldsymbol{x}_n, \boldsymbol{w}) + \xi_n \quad \forall n = 1, \dots, \ell, \ \forall \boldsymbol{y} \in \mathcal{Y}, \tag{5.3}$$

$$\xi_n \ge 0, \quad n = 1, \dots, N.$$

The set (5.3) of linear inequalities describes an intersection of half-spaces. In our case both N and $|\mathcal{Y}|$ are finite in (5.3), so the constraints describe a polyhedron. Despite being finite, $|\mathcal{Y}|$ is of exponential size in the length of the input representation. Therefore, (5.3) cannot be explicitly optimized over.

Instead, we use *delayed constraint generation* where we *start* with no constraints (5.3) and solve (5.2) to obtain a candidate solution. We then verify whether the candidate solution violates any of the inequalities (5.3). If it does, the violated inequality is explicitly generated and added to the problem and the problem is resolved. If the candidate solution turns out not to violate any inequality, then by the above reasoning the candidate solution is also the optimal solution. The incrementally growing problem is the *restricted master problem*, and the problem of finding violated inequalities is the *separation problem*.

The overall procedure is summarized in Algorithm 2. The algorithm iterates between solving the restricted master problem and generating violated constraints. The constraints found are used to tighten the master problem which is then resolved. If no violated constraints can be found, the procedure terminates. In each iteration, the maximum violation magnitude can be used as a convergence criterion and usually in practice one stops training once it is small enough. Because in our case $|\mathcal{Y}|$ is finite, the algorithm is finitely convergent, a fact proved in Tsochantaridis et al. [46].

In each algorithm iteration we are given a candidate parameter vector $\boldsymbol{w}$, and for each sample $(\boldsymbol{x}_n, \boldsymbol{y}_n)$ we need to solve the separation problem

$$\max_{\boldsymbol{y} \in \mathcal{Y}} \left(\Delta(\boldsymbol{y}_n, \boldsymbol{y}) + E(\boldsymbol{y}_n; \boldsymbol{x}_n, \boldsymbol{w}) - E(\boldsymbol{y}; \boldsymbol{x}_n, \boldsymbol{w})\right).$$

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 $\boldsymbol{y}_n^*$. This maximizer defines a new constraint, and by

Algorithm 2. Structured SVM training.

$\boldsymbol{w} = \textsc{StructuredSVM}(X, Y, C)$
Input:
  $\{(x_n, y_n)\}_{n=1,\dots,N}$ training set, $(x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$
  $C > 0$ regularization parameter
  $\epsilon \geq 0$ convergence tolerance
Output:
  $\boldsymbol{w} \in \mathbb{R}^F$ learned weight vector
Algorithm:
$D_{\boldsymbol{w},\boldsymbol{\xi}} \leftarrow \mathbb{R}^F \times \mathbb{R}^N_+$ {Initially: no constraints}
loop
  $(\boldsymbol{w}^*, \boldsymbol{\xi}^*) \leftarrow \left\{ \begin{array}{ll} \underset{\boldsymbol{w},\boldsymbol{\xi}}{\text{argmin}} & \|\boldsymbol{w}\|_2^2 + C \sum_{n=1}^N \xi_n \\ \text{s.t.} & (\boldsymbol{w}, \boldsymbol{\xi}) \in D_{\boldsymbol{w},\boldsymbol{\xi}} \end{array} \right.$ {Solve master}
  maxviol $\leftarrow -\infty$
  for $n = 1, \dots, N$ **do**
    $(\text{viol}, \boldsymbol{y}_v) \leftarrow (\max, \underset{\boldsymbol{y} \in \mathcal{Y}}{\text{argmax}}) \, [E(\boldsymbol{y}_n; \boldsymbol{x}_n, \boldsymbol{w}^*) - E(\boldsymbol{y}; \boldsymbol{x}_n, \boldsymbol{w}^*)$
      $+ \Delta(\boldsymbol{y}_n, \boldsymbol{y}) - \xi_n^*]$ {Solve separation problem}
    if viol > 0 **then**
      $D_{\boldsymbol{w},\boldsymbol{\xi}} \leftarrow D_{\boldsymbol{w},\boldsymbol{\xi}} \cap \{\boldsymbol{w}, \boldsymbol{\xi} : E(\boldsymbol{y}_n; \boldsymbol{x}_n, \boldsymbol{w}) + \Delta(\boldsymbol{y}_n, \boldsymbol{y}_v) \leq E(\boldsymbol{y}_v; \boldsymbol{x}_n, \boldsymbol{w}) + \xi_n\}$
    end if
    maxviol $\leftarrow \max\{\text{viol}, \text{maxviol}\}$
  end for
  if maxviol $> \epsilon$ **then**
    break
  end if
end loop

iterating between generating constraints and solving the quadratic program, we can obtain successively better parameter vectors $\boldsymbol{w}$.

Finley and Joachims [14] 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 (2.3) with the MAP-MRF LP local polytope defines an exactly solvable relaxation.

5.3. 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 often have more than one object in 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

Table 1

Results of the VOC 2008 segmentation experiment. Cases in bold are where a method outperforms the others.

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

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 behavior of the hard-connectedness CRF on test images is shown in Figures 18–20 for the aeroplane class. In the first two segmentations, connectedness helps by completing a discontinuous segmentation and by removing clutter. Figure 20 shows a hopeless case: if the CRF segmentation is that wrong, connectedness cannot help.

6. Conclusions. We have shown how the limitation of considering only 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 (2.3) can be accommodated into recent efficient dual LP MAP solvers [16, 29, 31, 43, 35] or graph-cut based algorithms [8, 7].

In this work we have considered constraints enforcing labeling with only one connected component. It is an open question how to generalize this to constraints that enforce or bias the solution toward a given number of connected components. The general polyhedral approach outlined in this work still applies, but we believe this multiple-component case has considerably more complex polyhedral structure.

In a wider sense, most computer vision research into MRF models have focused attention only on low-order interactions in sparsely connected graphs. Although even for this setting the general case is already hard, the conditional independence embodied in the Markov assumption allowed the development of tractable inference procedures. But there is additional structure possible which does not fit well in this standard setting: the global potential function we considered in this paper does not have a factorizable structure. Still, efficient approximate inference is possible by exploiting the *combinatorial structure*. In this work we have achieved this by combining the LP MAP-MRF relaxation with a suitable polytope derived from the global potential function. Whether there are more efficient ways to achieve the same effect is an open question.

All software is available as open-source at http://www.kyb.mpg.de/bs/people/nowozin/tuwo/.

Figure 18. *Image/CRF/CRF + conn. Case where connectedness helps: the local evidence is scattered; enforcing connectedness (right) helps.*

Figure 19. *Image/CRF/CRF + conn. Connectedness can remove clutter: local evidence (edges on the runway) is overridden.*

Figure 20. *Image/CRF/CRF+conn. Failure case: the CRF segmentation is bad (middle); connectedness does not help (right).*

Appendix A. Proofs.

Proof of Lemma 2.3. Every single node k constitutes a connected subgraph. By setting $y_k = 1$, $y_h = 0$ for $h \neq k$, a feasible solution is obtained. All these solutions are affinely independent. Furthermore, the empty graph is also a feasible subgraph. It follows that $\dim(Z) = |V|$; i.e., the connected subgraph polytope has full dimension. ∎

Proof of Lemma 2.4. First, $y_i \geq 0$. For each i, we construct $|V|$ affinely independent points in C with $y_i = 0$. Fix i; then one solution is obviously $\boldsymbol{x} = \mathbf{0}$, the empty subgraph. Next, for all $p \neq i$, obtain one solution by setting only $y_p = 1$, and for all $j \neq p$ set $y_j = 0$. Clearly, $y_j = 0$ and the $|V| - 1$ solutions thus obtained are affinely independent. In total we have $|V|$ solutions with $y_i = 0$; thus $y_i \geq 0$ is facet-defining.

Second, $y_i \leq 1$. Again let i be arbitrary. We construct $|V|$ affinely independent points in C with $y_i = 1$. For this, set $y_i = 1$ and $y_j = 0$ for all $j \neq i$. This is obviously one solution. Now root a spanning tree in i and set one node k at a time to $y_k = 1$, respecting the order of the spanning tree; i.e., the subgraph of selected nodes j with $y_j = 1$ always remains a connected subgraph of the spanning tree. This constructs $|V| - 1$ solutions, all affinely independent. Adding the first solution yields $|V|$ solutions in total, completing the proof. ∎

Proof of Theorem 2.6. First, the direction "is feasible," implying "is connected." Consider an arbitrary feasible $\boldsymbol{y}$. Due to integrality we have $y_i \in \{0,1\}$ for all $i \in V$. If $\sum_i y_i \leq 1$, the resulting subgraph is trivially connected; hence assume $\sum_i y_i \geq 2$. For arbitrary $y_i = 1$, $y_j = 1$, $i \neq j$, assume i and j are not connected; that is, $(i,j) \notin E$, and, moreover, there exists no path on G with all vertex variables being one. Trivially, we construct a vertex-separator set $S = \{k \in V : y_k = 0\}$ with $S \in \mathcal{S}(i,j)$. The removal of S from V must disconnect i and j, as $(i,j) \notin E$. However, by (2.1) we must have $y_i + y_j - \sum_{k \in S} y_k - 1 = 2 - 0 - 1 = 1 \leq 0$, which is clearly violated. Thus, feasibility implies connectedness. Second, the direction "is connected," implying "is feasible." Take any $y_i = 1$, $y_j = 1$, $i \neq j$, and i, j connected in G by a path starting at i and ending at j such that all intermediate nodes k satisfy $y_k = 1$. For all separators $S \in \mathcal{S}(i,j)$, at least one node t of this path must satisfy $t \in S$. Therefore $y_i + y_j - \sum_{k \in S} y_k - 1 \leq y_i + y_j - y_t - 1 = 0 \leq 0$ is satisfied. Thus any connected subgraph is feasible. ■

Proof of Theorem 2.7. We will prove this for any $i, j \in V$ by constructing $|V|$ affinely independent points in C which satisfy the inequality nonstrictly; that is, they also satisfy the corresponding equality. By [52, section 9.2.3] this shows that the inequality is facet-defining.

For $i, j \in V$ arbitrarily chosen, for any $S \in \bar{\mathcal{S}}(i,j)$, let $S = \{s_1, \ldots, s_{|S|}\}$ be the set of nodes in the essential vertex-separator set. Further, let S induce a partitioning of the graph into the set S, the connected subgraphs P_i and P_j, containing i and j, respectively, and the connected subgraphs P_s connected to exactly one $s \in S$ (if any subgraph is connected to more than one $s \in S$, remove all but one edge arbitrarily). This is shown in Figure 21.

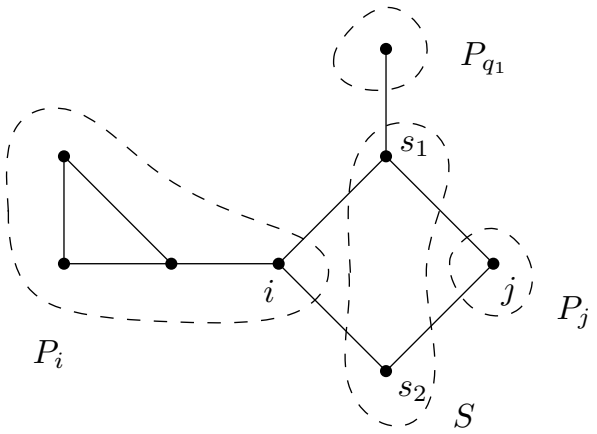

Figure 21. *The separator set S induces a graph partitioning.*

First, we construct $|P_i|+|P_j|$ affinely independent solutions in C which satisfy the equality.

1. For the connected subgraph P_i, root a spanning tree in i. Set $y_i = 1$, $y_k = 0$ for all $k \in P_i, k \neq i$. For each such $k \in P_i$, enlarge the subgraph incrementally by one node in an arbitrary ordering respecting the spanning tree; i.e., set $y_k = 1$. Each enlarged solution is a connected subgraph of P_i and G, and is affinely independent of all previous ones and satisfies the equality.
2. Likewise, do this for P_j, starting with just $y_j = 1$.

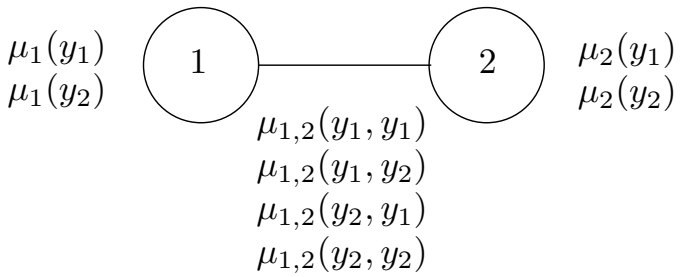

Figure 22. *Simple two-node MRF. The representation used in the LP relaxation defines four variables for the node states and four variables for the pairwise node states associated to the edge.*

Next, for each $s \in S$, we construct $|P_s|+1$ affinely independent solutions satisfying the equality as follows.

1. Set $y_k = 1$, for all $k \in P_i \cup P_j$, and $y_s = 1$. This solution is in C because S is essential and thus s connects P_i and P_j. Construct $|P_s|$ more solutions by building a spanning tree for P_s, rooted in the node connected to s. By incrementally setting $y_k = 1$ in an order respecting the spanning tree, $|P_s|$ affinely independent solutions in C are obtained.

We now consider the total number of solutions constructed:

$$|P_i| + |P_j| + \sum_{s \in S} (|P_s| + 1) = |V|.$$

We have constructed $|V|$ affinely independent solutions in C satisfying the equality. Therefore, by [52, section 9.2.3], the inequality defines a facet of $\mathrm{conv}(C)$. ∎

Appendix B. Solution integrality. We now discuss the integrality of the solutions to the constructed relaxation. The property we are interested in is the preservation of tightness of the relaxation: if we have two polytopes describing tight relaxations and we construct the intersection, do we still obtain a tight relaxation?

In general, the answer is no. By means of constructing a simple counterexample, we show that even if both the marginal polytope relaxation and the relaxation of the restricted feasible set in the node-label dimensions are tight, the intersection of both polytopes need not be. That is, it can contain new fractional vertices, even if both original polytopes contain only integral $\{0, 1\}$-vertices.

To see this, consider the simple two-node MRF shown as a graphical model in Figure 22. In the parametrization used by the linear programming relaxation (1.3), there are eight variables, four for the node states ($\mu_1(y_1)$, $\mu_1(y_2)$, $\mu_2(y_1)$, $\mu_2(y_2)$) and four for the pairwise node states at the edge ($\mu_{1,2}(y_1, y_1)$, $\mu_{1,2}(y_1, y_2)$, $\mu_{1,2}(y_2, y_1)$, $\mu_{1,2}(y_2, y_2)$).

The feasible set described by the constraints of the LP relaxation is given by the following

set of constraints:

$$
\begin{aligned}
(\text{B.1}) \qquad M = \{\mu : \; & \mu_1(y_1) + \mu_1(y_2) = 1, \\
& \mu_2(y_1) + \mu_2(y_2) = 1, \\
& \mu_{1,2}(y_1, y_1) + \mu_{1,2}(y_1, y_2) = \mu_1(y_1), \\
& \mu_{1,2}(y_2, y_1) + \mu_{1,2}(y_2, y_2) = \mu_1(y_2), \\
& \mu_{1,2}(y_1, y_1) + \mu_{1,2}(y_2, y_1) = \mu_2(y_1), \\
& \mu_{1,2}(y_1, y_2) + \mu_{1,2}(y_2, y_2) = \mu_2(y_2), \\
& \mu_1(y_1), \mu_1(y_2), \mu_2(y_1), \mu_2(y_2) \geq 0, \\
& \mu_{1,2}(y_1, y_1), \mu_{1,2}(y_1, y_2), \mu_{1,2}(y_2, y_1), \mu_{1,2}(y_2, y_2) \geq 0\}.
\end{aligned}
$$

The above constraints define the feasible set as a three-dimensional polytope embedded in eight dimensions. We can visualize the polytope partially by *projecting* it onto subspaces. For this, let us define the projection of a polytope.

Definition B.1 (projection of a polytope). *For a given polytope $Q \subseteq (\mathbb{R}^n \times \mathbb{R}^p)$, the* projection *of Q onto the subspace $\mathbb{R}^n$, denoted* $\text{proj}_x Q$*, is defined as*

$$\text{proj}_x Q = \{x \in \mathbb{R}^n : (x, w) \in Q \text{ for some } w \in \mathbb{R}^p\}.$$

Therefore, a point is in the projected set if there is at least one point in the higher-dimensional polytope which has identical coefficients in the projection dimensions. For additional properties of projected polytopes, see [1, 52, 41].

Figure 23 shows the projection $\text{proj}_{\mu_1(y_1), \mu_2(y_1), \mu_{1,2}(y_1, y_1)} M$ of the feasible set of the MRF shown in Figure 22. The full set of vertices of the polytope M is given as follows:

$$
\begin{aligned}
& \{(\mu_1(y_1), \mu_1(y_2), \mu_2(y_1), \mu_2(y_2), \;\; \mu_{1,2}(y_1, y_1), \mu_{1,2}(y_1, y_2), \mu_{1,2}(y_2, y_1), \mu_{1,2}(y_2, y_2))\} \\
= & \{(1, 0, 1, 0, \;\; 1, 0, 0, 0), \; (1, 0, 0, 1, \;\; 0, 1, 0, 0), \; (0, 1, 1, 0, \;\; 0, 0, 1, 0), \; (0, 1, 0, 1, \;\; 0, 0, 0, 1)\}.
\end{aligned}
$$

Therefore, all vertices are integral, and for this particular MRF the LP relaxation is tight. The feasible set defined by the LP relaxation is therefore identical to the true set, the *marginal polytope* [48].

Now suppose that we want to restrict the labelings such that both nodes are not labeled y_1 at the same time. Then, the only allowed combinations for $(\mu_1(y_1), \mu_2(y_1))$ are from the set $L = \{(0, 0), (0, 1), \; (1, 0)\}$. The convex hull $\text{conv}(L)$ is shown in Figure 24. The facet-defining constraints of the convex hull are simply $\mu_1(y_1) \geq 0$, $\mu_2(y_1) \geq 0$, and $\mu_1(y_1) + \mu_2(y_1) \leq 1$. We plan to add these new constraints to the feasible set of the MRF, defined by (B.1). Because the first two nonnegativity constraints are already in the constraint set, we only have to consider the new inequality $\mu_1(y_1) + \mu_2(y_1) \leq 1$.

Adding a constraint in the subspace of $\mu_1(y_1)$ and $\mu_2(y_1)$ is the same as first extending the set shown in Figure 24 to the full-dimensional space and then intersecting it with the marginal polytope. We show a three-dimensional projection of the extended feasible set in Figure 25.

The intersection of polytopes shown in Figures 25 and 23 is shown in Figure 26. The new polytope contains only points which satisfy $\mu_1(y_1) + \mu_2(y_1) \leq 1$ and (B.1). The polytope has

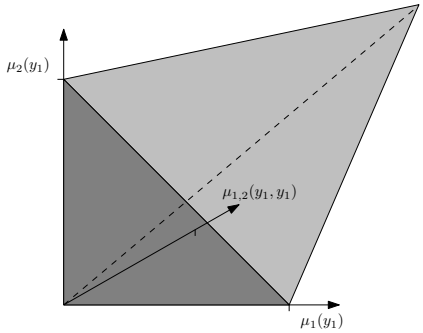

Figure 23. *Projection of the marginal polytope M onto the $\mu_1(y_1)$, $\mu_2(y_1)$, and $\mu_{1,2}(y_1, y_1)$ dimensions, i.e., $proj_{\mu_1(y_1),\mu_2(y_1),\mu_{1,2}(y_1,y_1)} M$.*

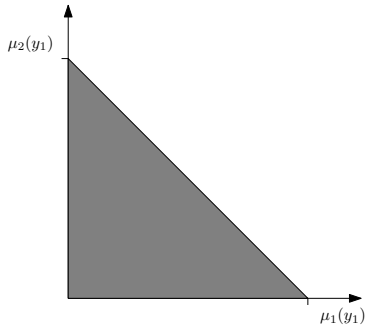

Figure 24. *Desired feasible set with respect to $\mu_1(y_1)$, $\mu_2(y_1)$. The nontrivial facet-defining inequality is $\mu_1(y_1) + \mu_2(y_1) \leq 1$.*

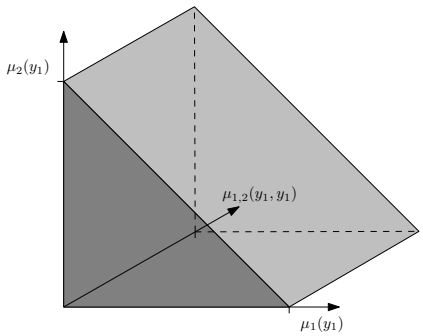

Figure 25. *Projected view of the extension to the full space of the desired feasible set with respect to $\mu_1(y_1)$, $\mu_2(y_1)$. Note that this polytope has only integral vertices.*

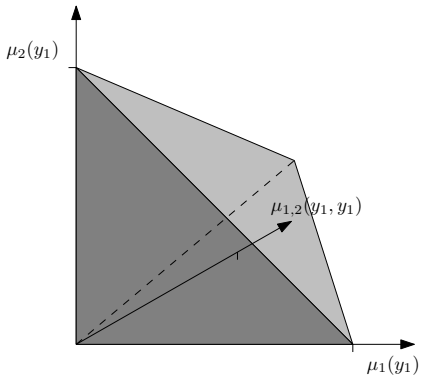

Figure 26. *Projected view of the resulting intersection with new fractional vertex $(\mu_1(y_1), \mu_2(y_1), \mu_{1,2}(y_1, y_1)) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$.*

the following set of vertices:

$$\begin{aligned}&\{(\mu_1(y_1), \mu_1(y_2), \mu_2(y_1), \mu_2(y_2),\ \mu_{1,2}(y_1, y_1), \mu_{1,2}(y_1, y_2), \mu_{1,2}(y_2, y_1), \mu_{1,2}(y_2, y_2))\} \\ &= \left\{(1,0,0,1,\ 0,1,0,0),\ (0,1,1,0,\ 0,0,1,0), (0,1,0,1,\ 0,0,0,1),\ \left(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2},\ \tfrac{1}{2}, 0, 0, \tfrac{1}{2}\right)\right\}.\end{aligned}$$

Therefore, although both polytopes have only integral vertices, their intersection has fractional ones. Note that the restriction of the intersection to the set of integral vertices still remains the exact set we are interested in: the subset of vertices of the marginal polytope satisfying $\mu_1(y_1) + \mu_2(y_1) \leq 1$.

In the above example, the simplified construction is qualitatively the same as the intersection of the connected subgraph polytope with the LP MAP-MRF relaxation LOCAL polytope [48]. Therefore, it is insightful in a number of ways.

First, having tight relaxations for both the connected subgraph polytope and the marginal polytope *does not guarantee* a tight relaxation for the convex hull of the integral vertices of their intersection.

Second, restricted to the set of integral solutions, the construction is exact. However, optimizing over only the integral solutions of the intersection is intractable, whereas optimizing over the intersection of two polytopes remains tractable if optimizing over the individual polytopes is tractable. Intersecting polytopes can therefore be thought as tractable relaxation to the intersection of their individual integral vertices: the new vertex set is a superset of the intersection of the individual polytopes' vertex sets.

To put this result into perspective, note the following three points. First, we never had a tight relaxation to start from. For general pairwise potentials, optimizing over the exact marginal polytope is NP-hard [48], so the LP relaxation is used. Optimizing over the exact subgraph polytope is NP-hard, so a relaxation is used. In order to remain tractable, both sets are relaxations and individually have fractional vertices. Whether the additional fractional vertices caused by intersection are an issue has to be settled empirically, as shown in Figure 14. Second, in general, finding inequalities which cut off fractional vertices of the intersection of two polytopes is hard; see [1, 52]. Third, as observed by Finley and Joachims in [14], structured learning of parameters in linear relaxations can "learn to avoid fractional solutions," as these always have a nonzero loss. In summary, intersecting polytopes weakens the overall relaxation.

Appendix C. Implementation details.

Separation routine. Our separation routine to find violated inequalities (2.3) is written in C++ and uses the boost 1.36 push-relabel max-flow solver.

MAP-MRF linear program. We solve (1.3) using the open-source COIN-OR Clp 1.8 solver[2] with the COIN-OR Osi 0.98.2 interface.[3] Instead of generating a single constraint at a time, we use *multiple pricing* and add as many violated constraints as we can find in each iteration, usually a few thousand. The cost of re-solving the LP relaxation is small compared to that of generating constraints. Finding additional violated constraints besides the most violating one incurs almost no additional cost.

Structured SVM. We solve (5.1) using the QP reformulation [46] in the dual by coordinate descent, similar to the approach in [17]. Unlike in that work, we need to ensure differentiability of the dual problem. Therefore, we add a small strictly convex proximal term in the primal, making it strictly convex in all variables. Strict convexity in the primal asserts dual differentiability everywhere [3], allowing our simple coordinate descent method to work. The advantage of the dual approach is the ability to rapidly warm-start once violating constraints have been found.

Acknowledgment. The authors would like to thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] E. Balas, *Projection, lifting and extended formulation in integer and combinatorial optimization*, Ann. Oper. Res., 2005, pp. 125–161.
[2] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool, *Speeded-up robust features (SURF)*, Comput. Vis. Image Underst., 110 (2008), pp. 346–359.
[3] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[2] Clp is available at https://projects.coin-or.org/Clp/.
[3] Osi is available at https://projects.coin-or.org/Osi/.

[4] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, Belmont, MA, 1997.
[5] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.
[6] M. B. Blaschko and C. H. Lampert, *Learning to localize objects with structured output regression*, in Proceedings of the 10th European Conference on Computer Vision (ECCV), Springer, Berlin, Heidelberg, 2008, pp. 2–15.
[7] Y. Boykov and V. Kolmogorov, *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision*, IEEE Trans. Pattern Anal. Mach. Intell., 26 (2004), pp. 1124–1137.
[8] Y. Boykov, O. Veksler, and R. Zabih, *Fast approximate energy minimization via graph cuts*, IEEE Trans. Pattern Anal. Mach. Intell, 23 (2001), pp. 1222–1239.
[9] D. Comaniciu and P. Meer, *Mean shift: A robust approach toward feature space analysis*, IEEE Trans. Pattern Anal. Mach. Intell, 24 (2002), pp. 603–619.
[10] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, McGraw–Hill, New York, 1990.
[11] P. Das, O. Veksler, V. Zavadsky, and Y. Boykov, *Semiautomatic segmentation with compact shape prior*, Image Vision Comput., 27 (2009), pp. 206–219.
[12] M. Everingham, L. V. Gool, C. K. Williams, J. Winn, and A. Zisserman, *The PASCAL Visual Object Classes Challenge Workshop* 2008, http://www.pascal-network.org/challenges/VOC/voc2008/workshop/.
[13] P. F. Felzenszwalb and D. P. Huttenlocher, *Efficient graph-based image segmentation*, Int. J. Comput. Vision, 59 (2004), pp. 167–181.
[14] T. Finley and T. Joachims, *Training structural SVMs when exact inference is intractable*, in Proceedings of the 25th International Conference on Machine Learning (ICML), ACM, New York, 2008, pp. 304–311.
[15] S. Geman and D. Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Trans. Pattern Anal. Mach. Intell., 6 (1984), pp. 721–741.
[16] A. Globerson and T. Jaakkola, *Fixing max-product: Convergent message passing algorithms for map lp-relaxations*, in Proceedings of Neural Information Processing Systems (NIPS), 2007.
[17] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, *A dual coordinate descent method for large-scale linear SVM*, in Proceedings of the 25th International Conference on Machine Learning (ICML), Vol. 307, 2008, pp. 408–415.
[18] T. Ideker, O. Ozier, B. Schwikowski, and A. F. Siegel, *Discovering regulatory and signalling circuits in molecular interaction networks*, Bioinformatics, 18 (2002), pp. S233–S240.
[19] T. Joachims, T. Finley, and C.-N. J. Yu, *Cutting-plane training of structural SVMs*, Mach. Learn., 77 (2009), pp. 27–59.
[20] R. M. Karp, *Maximum-Weight Connected Subgraph Problem*, 2002; available online from http://www.cytoscape.org/ISMB2002/nph.pdf.
[21] P. Kohli, M. P. Kumar, and P. Torr, *P3 & beyond: Solving energies with higher order cliques*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2007.
[22] P. Kohli, L. Ladický, and P. H. S. Torr, *Robust higher order potentials for enforcing label consistency*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
[23] P. Kohli, A. Shekhovtsov, C. Rother, V. Kolmogorov, and P. H. S. Torr, *On partial optimality in multi-label MRFs*, in Proceedings of the 25th International Conference on Machine Learning (ICML), Vol. 307, 2008, pp. 480–487.
[24] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, The MIT Press, Cambridge, MA, 2009.
[25] V. Kolmogorov, *Convergent tree-reweighted message passing for energy minimization*, IEEE Trans. Pattern Anal. Mach. Intell., 28 (2006), pp. 1568–1583.
[26] V. Kolmogorov and R. Zabih, *What energy functions can be minimized via graph cuts?*, IEEE Trans. Pattern Anal. Mach. Intell., 26 (2004), pp. 147–159.
[27] N. Komodakis and N. Paragios, *Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles*, in Proceedings of the 10th European Conference on Computer Vision (ECCV), 2008.

[28] N. Komodakis and N. Paragios, *Beyond pairwise energies: Efficient optimization for higher-order MRFs*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2009.
[29] N. Komodakis, N. Paragios, and G. Tziritas, *MRF optimization via dual decomposition: Message-passing revisited*, in Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV), 2007.
[30] M. P. Kumar, V. Kolmogorov, and P. Torr, *An analysis of convex relaxations for MAP estimation*, in Proceedings of Neural Processing Systems (NIPS), 2008.
[31] M. P. Kumar and P. Torr, *Efficiently solving convex relaxations for MAP estimation*, in Proceedings of the 25th International Conference on Machine Learning (ICML), 2008.
[32] J. Lafferty, A. McCallum, and F. Pereira, *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*, in Proceedings of the Eighteenth International Conference on Machine Learning (ICML), Morgan Kaufmann, San Francisco, 2001, pp. 282–289.
[33] S. Z. Li, *Markov random field models in computer vision*, in Computer Vision—ECCV'94, J.-O. Eklundh, ed., Lecture Notes in Comput. Sci. 801, Springer, Berlin, 1994, pp. 361–370.
[34] Y. Li and D. Huttenlocher, *Learning for stereo vision using the structured support vector machine*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
[35] T. Meltzer, A. Globerson, and Y. Weiss, *Convergent message passing algorithms - a unifying view*, in Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), Montreal, AUAI Press, Corvallis, OR, 2009.
[36] G. Mori, *Guiding model search using segmentation*, in Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV), 2005.
[37] S. Nowozin and C. H. Lampert, *Global connectivity potentials for random field models*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2009.
[38] S. Ramalingam, P. Kohli, K. Alahari, and P. H. S. Torr, *Exact inference in multi-label CRFs with higher order cliques*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
[39] X. Ren and J. Malik, *Learning a classification model for segmentation*, in Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV), 2003.
[40] C. Rother, P. Kohli, W. Feng, and J. Jia, *Minimizing sparse higher order energy functions of discrete variables*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2009.
[41] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, New York, 1998.
[42] D. Sontag and T. Jaakkola, *New outer bounds on the marginal polytope*, in Proceedings of Neural Information Processing Systems (NIPS), 2007.
[43] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss, *Tightening LP relaxations for MAP using message passing*, in Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), Helsinki, AUAI Press, Corvallis, OR, 2008.
[44] C. Sutton and A. McCallum, *An introduction to conditional random fields for relational learning*, in Introduction to Statistical Relational Learning, The MIT Press, Cambridge, MA, 2007, pp. 93–126.
[45] M. Szummer, P. Kohli, and D. Hoiem, *Learning CRFs using graph cuts*, in Proceedings of the 10th European Conference on Computer Vision (ECCV), 2008.
[46] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, *Large margin methods for structured and interdependent output variables*, J. Mach. Learn. Res., 6 (2005), pp. 1453–1484.
[47] S. Vicente, V. Kolmogorov, and C. Rother, *Graph cut based image segmentation with connectivity priors*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
[48] 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. Inform. Theory, 51 (2005), pp. 3697–3717.
[49] Y. Weiss, C. Yanover, and T. Meltzer, *Map estimation, linear programming and belief propagation with convex free energies*, in Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), Vancouver, AUAI Press, Corvallis, OR, 2007.

[50] T. Werner, *A linear programming approach to max-sum problem: A review*, IEEE Trans. Pattern Anal. Mach. Intell., 29 (2007), pp. 1165–1179.
[51] T. Werner, *High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF)*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
[52] L. A. Wolsey, *Integer Programming*, John Wiley & Sons, New York, 1998.
[53] C. Yanover, T. Meltzer, and Y. Weiss, *Linear programming relaxations and belief propagation—an empirical study*, J. Mach. Learn. Res., 7 (2006), pp. 1887–1907.
[54] 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*, Comput. Vis. Image Underst., 112 (2008), pp. 81–90.