Cooperative Graphical Models

Josip Djolonga  
Dept. of Computer Science, ETH Zürich  
josipd@inf.ethz.ch

Stefanie Jegelka  
CSAIL, MIT  
stefje@mit.edu

Sebastian Tschiatschek  
Dept. of Computer Science, ETH Zürich  
stschia@inf.ethz.ch

Andreas Krause  
Dept. of Computer Science, ETH Zürich  
krausea@inf.ethz.ch

Abstract

We study a rich family of distributions that capture variable interactions significantly more expressive than those representable with low-treewidth or pairwise graphical models, or log-supermodular models. We call these cooperative graphical models. Yet, this family retains structure, which we carefully exploit for efficient inference techniques. Our algorithms combine the polyhedral structure of submodular functions in new ways with variational inference methods to obtain both lower and upper bounds on the partition function. While our fully convex upper bound is minimized as an SDP or via tree-reweighted belief propagation, our lower bound is tightened via belief propagation or mean-field algorithms. The resulting algorithms are easy to implement and, as our experiments show, effectively obtain good bounds and marginals for synthetic and real-world examples.

1 Introduction

Probabilistic inference in high-order discrete graphical models has been an ongoing computational challenge, and all existing methods rely on exploiting specific structure: either low-treewidth or pairwise graphical models, or functional properties of the distribution such as log-submodularity. Here, we aim to compute approximate marginal probabilities in complex models with long-range variable interactions that do not possess any of these properties. Instead, we exploit a combination of structural and functional properties in new ways.

The classical example of image segmentation may serve to motivate our family of models: we would like to estimate a posterior marginal distribution over \( k \) labels for each pixel in an image. A common approach uses Conditional Random Fields on a pixel neighborhood graph with pairwise potentials that encourage neighboring pixels to take on the same label. From the perspective of the graph, this model prefers configurations with few edges cut, where an edge is said to be cut if its endpoints have different labels. Such cut-based models, however, short-cut elongated structures (e.g. tree branches), a problem known as shrinking bias. Jegelka and Bilmes \[1\] hence replace the bias towards short cuts (boundaries) by a bias towards configurations with certain higher-order structure: the cut edges occur at similar-looking pixel pairs. They group the graph edges into clusters (based on, say, color gradients across the endpoints), observing that the true object boundary is captured by few of these clusters. To encourage cutting edges from few clusters, the cost of cutting an edge decreases as more edges in its cluster are cut. In short, the edges “cooperate”. In Figure 1 each pixel takes on one of two labels (colors), and cut

Figure 1: Example cooperative model. Edge colors indicate the edge cluster. Dotted edges are cut under the current assignment.
While being expressive, such models are computationally very challenging: the nonlinear function

\[
\min_{y \in X}\, J(y) = \min_{y \in X}\, \sum_{i,j} \theta_{i,j}(y_{i,j}) + f(y(x))
\]

where we call \(y\) variables. With \(f\) factor, which is infeasible for our cooperative models. Some specialized tractable inference methods focus on pairwise models (in \([7]\). We will build on a selection of techniques that we discuss in the next section. Most existing models (when \(f \equiv 0\)) have no factorization assumptions, which is infeasible for our cooperative models. We develop new inference methods for these models; in particular, (2) upper bounds that are amenable to convex optimization, and (3) lower bounds that we optimize with \(\nu\). In contrast, we make one additional key assumption that will enable the development of efficiently computable variational lower and upper bounds: we henceforth assume that \(f: \{0, 1\}^E \to \mathbb{R}\) is submodular, i.e., it satisfies

\[
f(\min(y, y')) + f(\max(y, y')) \leq f(y) + f(y') \quad \text{for all } y, y' \in \{0, 1\}^E,
\]

where the \(\min\) and \(\max\) operations are taken element-wise. For example, the pairwise potentials \(\theta_{i,j}\) are submodular if \(\theta_{i,j}(0, 0) + \theta_{i,j}(1, 1) \leq \theta_{i,j}(0, 1) + \theta_{i,j}(1, 0)\). In our introductory example, \(f\) is submodular if \(h\) is concave. As opposed to \([3]\), we do not assume that \(f\) is monotone increasing. Importantly, even if \(f\) is submodular, \(P(x)\) neither has low treewidth, nor is its logarithm sub- or supermodular in \(x\), properties that have commonly been exploited for inference.

Contributions. We make the following contributions: (1) We introduce a new family of probabilistic models that can capture rich non-submodular interactions, while still admitting efficient inference. This family includes pairwise and certain higher-order graphical models, cooperative cuts \([1]\), and other, new models. We develop new inference methods for these models; in particular, (2) upper bounds that are amenable to convex optimization, and (3) lower bounds that we optimize with traditional variational methods. Finally, we demonstrate the efficacy of our methods empirically.

1.1 Related work

Maximumposteriori (MAP). Computing the mode of \([1]\) for binary models is also known as the cooperative cut problem, and has been analyzed for the case when both the pairwise interactions \(\theta_{i,j}\) are submodular and \(f\) is monotone \([1]\). While the general problem is NP-hard, it can be solved if \(f\) is defined by a piecewise linear concave function \([4]\).

Variational inference. Since computing marginal probabilities for \([1]\) is \#P-hard even for pairwise models (when \(f \equiv 0\)) \([5][6]\), we revert to approximate inference. Variational inference methods for discrete pairwise models have been studied extensively; a comprehensive overview may be found in \([7]\). We will build on a selection of techniques that we discuss in the next section. Most existing methods focus on pairwise models (\(f \equiv 0\)), and many scale exponentially with the size of the largest factor, which is infeasible for our cooperative models. Some specialized tractable inference methods exist for higher-order models \([8][9]\), but they do not apply to our family of models \([1]\).

The results presented in this paper can be easily extended to arbitrary binary-valued functions \(y(x)\).
Log-supermodular models. A related class of relatively tractable models are distributions $P(x) = \frac{1}{Z} \exp(-g(x))$ for some submodular function $g$; Djolonga and Krause [10] showed variational inference methods for those models. However, our models are not log-supermodular. While [10] also obtain upper and lower bounds, we need different optimization techniques, and also different polytopes. In fact, submodular and multi-class submodular [11] settings are a strict subset of ours: the function $g(x)$ can be expressed via an auxiliary variable $z \in \{0, 1\}$ that is fixed to zero using $\nu(x, z) = \lceil z = 0 \rceil$. We then set $f(y(x, z)) = g(x_1 \neq z, x_2 \neq z, \ldots, x_n \neq z)$.

2 Notation and Background

Throughout this paper, we have $n$ variables in a graph of $m$ edges, and the potentials $\theta_i$ and $\theta_{i,j}$ are stored in a vector $\theta$. The characteristic vector (or indicator vector) $1_A$ of a set $A$ is the binary vector which contains 1 in the positions corresponding to elements in $A$, and zeros elsewhere. Moreover, the vector of all ones is $1$, and the neighbours of $i \in V$ are denoted by $\delta(i) \subseteq V$.

Submodularity. We assume that $f$ in Eqn. (1) is submodular. Occasionally (in Sec. 4 and 5 where stated), we assume that $f$ is monotone: for any $y$ and $y'$ in $\{0, 1\}^E$ such that $y \leq y'$ coordinate-wise, it holds that $f(y) \leq f(y')$. When defining the inference schemes, we make use of two polytopes associated with $f$. First, the base polytope of a submodular function $f$ is

$$B(f) = \{ g \in \mathbb{R}^m \mid \forall y \in \{0, 1\}^E: g^T y \leq f(y) \} \cap \{ g \in \mathbb{R}^m \mid g^T 1 = f(1) \}.$$.

Although $B(f)$ is defined by exponentially many inequalities, an influential result [12] states that it is tractable: we can optimize linear functions over $B(f)$ in time $O(m \log m + mF)$, where $F$ is the time complexity of evaluating $f$. This algorithm is part of our scheme in Figure 2. Moreover, as a result of this (linear) tractability, it is possible to compute orthogonal projections onto $B(f)$. Projection is equivalent to the minimum norm point problem [13]. While the general projection problem has a degree polynomial time complexity, there are many very commonly used models that admit practically fast projections [14][15][16].

The second polytope is the upper submodular polyhedron of $f$ [17], defined as

$$U(f) = \{ (g, c) \in \mathbb{R}^{m+1} \mid \forall y \in \{0, 1\}^E: g^T y + c \geq f(y) \}.$$.

Unfortunately, $U(f)$ is not as tractable as $B(f)$: even checking membership in $U(f)$ is hard [17]. However, we can still succinctly describe specific elements of $U(f)$. In [4] we show how to efficiently optimize over those elements.

Variational inference. We briefly summarize key results for variational inference for pairwise models, following Wainwright and Jordan [7]. We write pairwise models as

$$P(x) = \exp \left( - \sum_{i \in V} \theta_i(x_i) - \sum_{\{i,j\} \in E} (g_{i,j}[x_i \neq x_j] + \theta_{i,j}(x_i, x_j)) - A(g) \right) \nu(x),$$

where $g \in \mathbb{R}^E$ is an arbitrary vector and $A(g)$ is the log-partition function. For any choice of parameters $(\theta, g)$, there is a resulting vector of marginals $\mu \in [0, 1]^{k|V| + k^2|E|}$. Specifically, for every $i \in V$, $\mu$ has $k$ elements $\mu_{i,x_i} = P(X_i = x_i)$, one for each $x_i \in X$. Similarly, for each $\{i, j\} \in E$, there are $k^2$ elements $\mu_{i,j,x_i,x_j}$ so that $\mu_{i,j,x_i,x_j} = P(X_i = x_i, X_j = x_j)$. The marginal polytope $\mathcal{M}$ is now the set of all such vectors $\mu$ that are realizable under some distribution $P(x)$, and the partition function can equally be expressed in terms of the marginals [7]:

$$A(g) = \sup_{\mu \in \mathcal{M}} \left( - \sum_{i \in V, x_i} \mu_{i,x_i} \theta_i(x_i) - \sum_{\{i,j\} \in E, x_i, x_j} \mu_{i,j,x_i,x_j} \theta_{i,j}(x_i, x_j) - \Delta(\mu)^T g \right) + H(\mu),$$

where $H(\mu)$ is the entropy of the distribution, $\Delta(\mu)$ is the vector of disagreement probabilities with entries $\Delta(\mu)_{i,j} = \sum_{x_i \neq x_j} \mu_{i,j,x_i,x_j}$, and stack$(\theta, g)$ adds the elements of $\theta$ and $g$ into a single

\footnote{This formulation is slightly nonstandard, but will be very useful for the subsequent discussion in [3].}
vector so that the sum can be written as an inner product. Alas, neither \( \overline{M} \) nor \( H(\mu) \) have succinct descriptions and we will have to approximate them. Because the vectors in the approximation of \( \overline{M} \) are in general not correct marginals, they are called pseudo-marginals and will be denoted by \( \tau \) instead of \( \mu \). Different approximations of \( \overline{M} \) and \( H \) yield various methods, e.g. mean-field [7], the semidefinite programming (SDP) relaxation of Wainwright and Jordan [18], tree-reweighted belief propagation (TRWBP) [19], or the family of weighted entropies [20, 21]. Due to the space constraints, we only discuss the latter. They approximate \( \overline{M} \) with the local polytope

\[
\mathbb{L} = \{ \{ \tau \geq 0 \mid (\forall i \in V) \sum_{x_i} \tau_{i,x_i} = 1 \text{ and } (\forall j \in \delta(i)) \tau_{i,x_i} = \sum_{x_j} \tau_{ij,x_i,x_j} \} \}.
\]

The approximations \( \overline{H} \) to the entropy \( H \) are parametrized by one weight \( \rho_{i,j} \) per edge and one \( \rho_i \) per vertex \( i \), all collected in a vector \( \rho \in \mathbb{R}^{|V|+|E|} \). Then, they take the following form

\[
\overline{H}(\tau, \rho) = \sum_{i \in V} \rho_i H_i(\tau_i) + \sum_{(i,j) \in E} \rho_{i,j} H_{ij}(\tau_{ij}), \quad H_i(\tau_i) = -\sum_{x_i} \tau_{i,x_i} \log \tau_{i,x_i}, \quad \text{and} \quad H_{ij}(\tau_{ij}) = -\sum_{x_{i},x_{j}} \tau_{ij,x_{i},x_{j}} \log \tau_{ij,x_{i},x_{j}}.
\]

The most prominent example is traditional belief propagation, i.e., using the Bethe entropy, which sets \( \rho_e = 1 \) for all \( e \in E \), and assigns to each vertex \( i \in V \) a weight of \( \rho_i = 1 - |\delta(i)| \).

### 3 Convex upper bounds

The above variational methods do not directly generalize to our cooperative models: the vectors of marginals could be exponentially large. Hence, we derive a different approach that relies on the submodularity of \( f \). Our first step is to approximate \( f(y(x)) \) by a linear lower bound, \( f(y(x)) \approx g^T y(x) \), so that the resulting (pairwise) linearized model will have a partition function upper bounding that of the original model. Ensuring that \( g \) indeed remains a lower bound means to satisfy an exponential number of constraints \( f(y(x)) \geq g^T y(x) \), one for each \( x \in \{0,1\}^V \). While this is hard in general, the submodularity of \( f \) implies that these constraints are easily satisfied if \( g \in B(f) \), a very tractable constraint. For \( g \in B(f) \), we have

\[
\log Z = \log \sum_{x \in \{0,1\}^V} \exp \left( -\sum_{i \in V} \theta_i(x_i) + \sum_{(i,j) \in E} \theta_{i,j}(x_i, x_j) + f(y(x)) \right) \\
\leq \log \sum_{x \in \{0,1\}^V} \exp \left( -\sum_{i \in V} \theta_i(x_i) + \sum_{(i,j) \in E} \left( \theta_{i,j}(x_i, x_j) + g_{i,j}[x_i \neq x_j] \right) \right) \equiv A(g).
\]

Unfortunately, \( A(g) \) is still very hard to compute and we need to approximate it. If we use an approximation \( A(g) \) that upper bounds \( A(g) \), then the above inequality will still hold when we replace \( A \) by \( \overline{A} \). Such approximations can be obtained by relaxing the marginal polytope \( \overline{M} \) to an outer bound \( \overline{M} \supseteq \overline{M} \), and using a concave entropy surrogate \( \overline{H} \) that upper bounds the true entropy \( H \). TRWBP [19] or the SDP formulation [18] implement this approach. Our central optimization problem is now to find the tightest upper bound, an optimization problem\(^1\) in \( g \):

\[
\text{minimize } \sup_{g \in B(f)} \{ \text{stack}(\theta, g), \tau \} + \overline{H}(\tau).
\]

Because the inner problem is linear in \( g \), this is a convex optimization problem over the base polytope. To obtain the gradient with respect to \( g \) (equal to the negative disagreement probabilities \(-\Delta(\tau)\)), we have to solve the inner problem. This subproblem corresponds to performing variational inference in a pairwise model, e.g. via TRWBP or an SDP. The optimization properties of the problem [3] depend on its Lipschitz continuity of the gradients (smoothness). Informally, the inferred pseudomarginals should not drastically change if we perturb the linearization \( g \). The formal condition is that there exists some \( \sigma > 0 \) so that \( \| \Delta(\tau) - \Delta(\tau') \| \leq \sigma \| \tau - \tau' \| \) for all \( \tau, \tau' \in \overline{M} \). We discuss below when this condition holds. Before that, we discuss two different algorithms for solving problem [3], and how their convergence depends on \( \sigma \).

\(^1\)If we compute the Fenchel dual, we obtain a special case of the problem considered in [22] with the Lovász extension acting as a non-smooth non-local energy function (in the terminology introduced therein).
Frank-Wolfe. Given that we can efficiently solve linear programs over \( B(f) \), the Frank-Wolfe [23] algorithm is a natural candidate for solving the problem. We present it in Figure 2. It iteratively moves towards the minimizer of a linearization of the objective around the current iterate. The method has a convergence rate of \( O(\sigma/t) \) [24], where \( \sigma \) is the assumed smoothness parameter. One can either use a fixed step size \( \gamma = 2/(t+2) \), or determine it using line search. In each iteration, the algorithm calls the procedure LINEAR-ORACLE, which finds the vector \( s \in B(f) \) that minimizes the linearization of the objective function in (3) over the base polytope \( B(f) \). The linearization is given by the (approximate) gradient \( \Delta(\tau) \), determined by the computed approximate marginals \( \tau \).

When taking a step towards \( s \), the weight of edge \( e_i \) is changed by \( s_{e_i} = f(\{e_1, e_2, \ldots, e_i\}) - f(\{e_1, e_2, \ldots, e_{i-1}\}) \). Due to the submodularity\(^4\) of \( f \), an edge will obtain a higher weight if it appears earlier in the order determined by the disagreement probabilities \( \Delta(\tau) \). Hence, in every iteration, the algorithm will re-adjusts the pairwise potentials, by encouraging the variables to agree more as a function of their (approximate) disagreement probability.

1: procedure FW-INFERENCES0, \( \theta \) 1: procedure LINEAR-ORACLE(\( f, \tau \))
2: \( g \leftarrow \text{LINEAR-ORACLE}(f, 0) \) 2: Let \( e_1, e_2, \ldots, e_{|E|} \) be the edges \( E \) sorted so that \( \Delta(\tau)_e \geq \Delta(\tau)_{e_2} \geq \ldots \geq \Delta(\tau)_{e_{|E|}} \)
3: for \( t = 0, 1, \ldots, \text{max_steps} \) do 3: for \( i = 0, 1, \ldots, |E| \) do
4: \( \tau \leftarrow \text{VAR-INFERENCE}(\theta, g) \) 4: \( f_{i+1} \leftarrow f(\{e_1, e_2, \ldots, e_i\}) \)
5: \( s \leftarrow \text{LINEAR-ORACLE}(f, \tau) \) 5: \( f_{i+1} \leftarrow f(\{e_1, e_2, \ldots, e_i\}) \)
6: \( \gamma \leftarrow \text{COMPUTE-STEP-SIZE}(g, s) \) 6: \( s_{e_i} \leftarrow f_{i+1} - f_i \)
7: \( g \leftarrow (1 - \gamma)g + \gamma s \) 7: return \( s \)

Figure 2: Inference with Frank-Wolfe, assuming that VAR-INFERENCE guarantees an upper bound.

Projected gradient descent (PGD). Since it is possible to compute projections onto \( B(f) \), and practically so for many submodular functions \( f \), we can alternatively use projected gradient or subgradient descent (PGD). Without smoothness, PGD converges at a rate of \( O(1/\sqrt{t}) \). If the objective is smooth, we can use an accelerated methods like FISTA [25], which has both a much better \( O(\sigma/t^2) \) rate and seems to converge faster than many Frank-Wolfe variants in our experiments.

Smoothness and convergence. The final question that remains to be answered is under which conditions problem (3) is smooth (the proof can be found in the appendix).

Theorem 1 Problem (3) is \( k^2\sigma \)-smooth over \( B(f) \) if the entropy surrogate \( -\bar{H} \) is \( \frac{1}{\sigma} \)-strongly convex.

This result follows from the duality between smoothness and strong convexity for convex conjugates, see e.g. [26]. It implies that the convergence rates of the proposed algorithms depend on the strong convexity of the entropy approximation \( -\bar{H} \). The benefits of strongly convex entropy approximations are known. For instance, the tree-reweighted entropy approximation is strongly convex with a modulus \( \sigma \) depending on the size of the graph; similarly, the SDP relaxation is strongly convex [27]. London et al. [28] provide an even sharper bound for the tree reweighted entropy, and show how one can strong-convexify any weighted entropy by solving a QP over the weights \( \rho \).

In practice, because the inner problem is typically solved using an iterative algorithm and because the problem is smooth, we obtain speedups by warm-starting the solver with the solution at the previous iterate. We can moreover easily obtain duality certificates using the results in [24].

Joint optimization. When using weighted entropy approximations, it makes sense to optimize over both the linearization \( g \) and the weights \( \rho \) jointly. Specifically, let \( \mathcal{W} \) be some set of weights that yield an entropy approximation \( \bar{H} \) that upper bounds \( H \). Then, if we expand \( \bar{H} \) in problem (3), we obtain

\[
\min_{g \in \mathcal{B}(f)} \sup_{\rho \in \mathcal{W}} \left( \text{stack}(\theta, g), \tau \right) + \sum_{i \in V} \rho_i H_i(\tau_i) + \sum_{\{i,j\} \in E} \rho_{i,j} H_{i,j}(\tau_{i,j}).
\]

Note that inside the supremum, both \( g \) and \( \rho \) appear only linearly, and there is no summand that has terms from both of them. Thus, the problem is convex in \( (g, \rho) \), and we can optimize jointly over

\(^4\)This is also known as the diminishing returns property.
both variables. As a final remark, if we already perform inference in a pairwise model and repeatedly tighten the approximation by optimizing over \( \rho \) via Frank-Wolfe (as suggested in [19]), then the complexity per iteration remains the same even if we use the higher-order term \( f \).

4 Submodular lower bounds

While we just derived variational upper bounds, we next develop lower bounds on the partition function. Specifically, analogously to the linearization for the upper bound, if we pick an element \( (g, c) \) of \( \mathcal{U}(f) \), the partition function of the resulting pairwise approximation always lower bounds the partition function of \( f \). Formally,

\[
\log Z \geq \log \sum_{x \in \{0,1\}^V} \exp \left( - (a^T x + \sum_{\{i,j\} \in E} \theta_{i,j,x_i,x_j} + \sum_{(i,j) \in E} g_{i,j}[x_i \neq x_j] + c) \right) = A(g) - c.
\]

As before, after plugging in a \( \text{lower} \) bound estimate of \( A \), we obtain a variational lower bound over the partition function, which takes the form

\[
\log Z \geq \sup_{(g,c) \in \mathcal{U}(f), \tau \in \mathcal{M}} -c + \langle \text{stack}(\theta, g), \tau \rangle + \mathcal{H}(\tau),
\]

for any pair of approximations of \( \mathcal{M} \) and \( \mathcal{H} \) that guarantee a lower bound of the pairwise model. We propose to optimize this lower bound in a block-coordinate-wise manner: first with respect to the pseudo-marginals \( \tau \) (which amounts to approximate inference in the linearized model), and then with respect to the supergradient \( (g, c) \in \mathcal{U}(f) \). As already noted, this step is in general intractable. However, it is well-known [29] that for any \( Y \subseteq E \) we can construct a point (so called \( \text{bar supergradient} \)) in \( \mathcal{U}(f) \) as follows. First, define the vectors \( a_{i,j} = f(1_{\{i,j\}}) \) and \( b_{i,j} = f(1) - f(1_{\{i,j\}}) \). Then, the vector \((g, c)\) with \( g = b \odot 1_Y + (1 - 1_Y) \circ a \) and \( c = f(Y) - b^T 1_Y \) belongs to \( \mathcal{U}(f) \), where \( \circ \) denotes element-wise multiplication.

Theorem 2 Optimizing problem (4) for a fixed \( \tau \) over all bar supergradients is equal to the following submodular minimization problem \( \min_{Y \subseteq E} f(Y) + (\Delta(\tau) \circ (b - a) - b)^T 1_Y \).

In contrast to computing the MAP, the above problem has no constraints and can be easily solved using existing algorithms. As the approximation algorithm for the linearized pairwise model, one can always use mean-field [7]. Moreover, if (i) the problem is binary with submodular pairwise potentials \( \theta_{i,j} \) and (ii) \( f \) is monotone, we can also use belief propagation. This is an implication of the result of Ruozi [30], who shows that traditional belief-propagation yields a lower bound on the partition function for binary pairwise log-supermodular models. It is easy to see that the above conditions are sufficient for the log-supermodularity of the linearized model, as \( g \geq 0 \) when \( f \) is monotone (because both \( a \) and \( b \) have non-negative components). Moreover, in this setting both the mean-field and belief propagation objectives (i.e. computing \( \tau \)) can be cast as an instance of continuous submodular minimization (see e.g. [31]), which means that they can be solved to arbitrary precision in polynomial time. Unfortunately, problem (4) will not be jointly submodular, so we still need to use the block-coordinate ascent method we have just outlined.

5 Approximate inference via MAP perturbations

For binary models with submodular pairwise potentials and monotone \( f \) we can (approximately) solve the MAP problem using the techniques in [1][4]. Hence, this opens as an alternative approach the perturb-and-MAP method of Papandreou and Yuille [32]. This method relies on a set of tractable first order perturbations: For any \( i \in V \) define \( \theta_i(x_i) = \theta_i(x_i) - \eta_i(x_i) \), where \( \eta = (\eta_i(x_i)) \in \mathbb{R}^V \) are a set of independently drawn Gumbel random variables. The optimizer \( \arg \min \theta \in \mathbb{R}^V \theta_i(x_i) + \sum_{(i,j) \in E} \theta_{i,j}(x_i, x_j) + f(\theta(x)) \) is then a sample from (an approximation to) the true distribution. If this MAP problem can be solved exactly (which is not always the case here), then it is possible to obtain an upper bound on the partition function [33].

6 Synthetic experiments

Synthetic experiments. Our first set of experiments uses a complete graph on \( n \) variables. The unary potentials were sampled as \( \theta_i(x_i) \sim \text{Uniform}(\pm \alpha) \). The edges \( E \) were randomly split
into five disjoint buckets $E_1, E_2, \ldots, E_5$, and we used $f(y) = \sum_{j=1}^{5} h_j(y_{E_i})$, where $y_{E_i}$ are the coordinates of $y$ corresponding to that group, and the functions $\{h_j\}$ will be defined below. To perform inference in the linearized pairwise models, we used: trwbp, jtree+ (exact inference, upper bound), jtree- (same, lower bound), sdp (SDP), mf (mean-field), bp (belief propagation), pmap (perturb-and-MAP with exact MAP) and epmap (perturb-and-MAP with approximate MAP). We used libDAI\textsuperscript{[34]} and implemented sdp using cvxpy\textsuperscript{[35]} and SCS\textsuperscript{[36]}. As a maxflow solver we used\textsuperscript{[37]} JTree. Errors bars denote three standard errors.

Figure 3 shows the results for $h_i(y_{E_i}) = w_i \sqrt{\sum_{e \in E_i} p_e / \sqrt{|E_i|}}$, with weights $w_i \sim \text{Uniform}(0, \beta)$. In panel (c) we use mixed (attractive and repulsive) pairwise potentials, chosen as $\theta_{i,j}(x_i, x_j) = w_{i,j}\|x_i \neq x_j\|$, where $w_{i,j} \sim \text{Uniform}(-\beta, \beta)$. First, the results imply that the methods optimizing the fully convex upper bound yield very good marginal probabilities over a large set of parameter configurations. The estimate of the log-partition function from trwbp is also very good, while sdp is much worse, which we believe can be attributed to the very loose entropy bound used in the relaxation. The lower bounds (bp and mf) work well for settings when the pairwise strength $\beta$ is small compared to the unary strength $\alpha$. Otherwise, both the bound and the marginals become worse, while jtree- still performs very well. This could be explained by the hardness of the pairwise models obtained after linearizing $f$. Finally, pmap (when applicable) seems very promising for small $\beta$.

To better understand the regimes when one should use trwbp or pmap, we compare their marginal errors in Figure 5. We see that for most parameter configurations, trwbp performs better, and significantly so when the edge interactions are strong.

Finally, we evaluate the effects of the approximate MAP solver for pmap in Figure 4. To be able to solve the MAP problem exactly (see \cite{1}), we used $h(y_{E_i}) = \max\{\sum_{e \in E_i} y_e v_e, \sum_{e \in E_i} v_e / 2\}$, where $v_e \sim \text{Uniform}(0, \beta)$. As evident from the figure, the gains from the exact solver seem minimal, and it seems that solving the MAP problem approximately does not strongly affect the results.

**An example from computer vision.** To demonstrate the scalability of our method and obtain a better qualitative understanding of the resulting marginals, we ran trwbp and pmap on a real world image segmentation task. We use the same setting, data and models as\textsuperscript{[1]}, as implemented in the pycoo package. Because libDAI was too slow, we wrote our own TRWBP implementation. Figure 6 shows the results for two specific images (size 305 × 398 and 214 × 320). The example in the first row is particularly difficult for pairwise models, but the rich higher-order model has no problem capturing the details even in the challenging shaded regions of the image. The second row shows results for two different model parameters. The second model uses a function $f$ that is closer to being linear, while the first one is more curved (see the appendix for details). We observe that trwbp requires lower temperature parameters (i.e. relatively larger functions $\theta_i, \theta_{i,j}$ and $f$) than pmap, and that the bottleneck of the complete inference procedure is running the trwbp updates. In other words, the added complexity from our method is minimal and the runtime is dominated by the message passing updates of TRWBP. Hence, any algorithms that speed up TRWBP (e.g., by parallelization or better message scheduling) will result in a direct improvement on the proposed inference procedure.

7 Conclusion

We developed new inference techniques for a new broad family of discrete probabilistic models by exploiting the (indirect) submodularity in the model, and carefully combining it with ideas from classical variational inference in graphical models. The result are inference schemes that optimize rigorous bounds on the partition function. For example, our upper bounds lead to convex variational inference problems. Our experiments indicate the scalability, efficacy and quality of these schemes.

**Acknowledgements.** This research was supported in part by SNSF grant CRSII2_147633, ERC StG 307036, a Microsoft Research Faculty Fellowship, a Google European Doctoral Fellowship, and NSF CAREER 1553284.

**References**


\[https://github.com/shelhamer/coop-cut\]
Figure 3: Results on several synthetic models. The methods that optimize the convex upper bound (trwbp, sdp) obtain very good marginals for a large set of parameter settings. Those maximizing the lower bound (bp, mf) fail when there is strong coupling between the edges. In the strong coupling regime the results of pmap also deteriorate, but not as strongly. In (c) bp, pmap, sdp are not applicable.

Figure 4: α = 2, K_{15}, model where epmap is applicable. Solving the MAP problem exactly only marginally improves over pmap. The other observations are similar to those in Fig. 3b.

Figure 5: \text{error}_{pmap} - \text{error}_{trwbp} on K_{15}. Missing entries were not significant at the 0.05 level.

Figure 6: Inferred marginals on an image segmentation task. The first row showcases an example that is particularly hard for pairwise models. In the second row we show the results for two different models (the cooperative function \( f \) is more curved for model 1).


