Approximating Marginals Using Discrete Energy Minimization

Filip Korč, Vladimir Kolmogorov and Christoph H. Lampert

IST Austria (Institute of Science and Technology Austria)

Am Campus 1

A-3400 Klosterneuburg

Technical Report No. IST-2012-0003

http://pub.ist.ac.at/Pubs/TechRpts/2012/IST-2012-0003.pdf

July 23, 2012

Approximating Marginals Using Discrete Energy Minimization

Filip Korč FILIP.KORC@IST.AC.AT
IST Austria, Am Campus 1, Klosterneuburg, Austria

Vladimir Kolmogorov VNK@IST.AC.AT
IST Austria, Am Campus 1, Klosterneuburg, Austria

Christoph H. Lampert CHL@IST.AC.AT
IST Austria, Am Campus 1, Klosterneuburg, Austria

Abstract

We consider the problem of inference in a graphical model with binary variables. While in theory it is arguably preferable to compute marginal probabilities, in practice researchers often use MAP inference due to the availability of efficient discrete optimization algorithms. We bridge the gap between the two approaches by introducing the *Discrete Marginals* technique in which approximate marginals are obtained by minimizing an objective function with unary and pairwise terms over a discretized domain. This allows the use of techniques originally developed for MAP-MRF inference and learning. We explore two ways to set up the objective function - by discretizing the Bethe free energy and by learning it from training data. Experimental results show that for certain types of graphs a learned function can outperform the Bethe approximation. We also establish a link between the Bethe free energy and submodular functions.

1. Introduction

We consider the problem of inference in a graphical model specified by the energy function

$$E(\boldsymbol{x}) = \sum_{i \in \mathcal{V}} \theta_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(x_i, x_j) \qquad (1)$$

with induced probability distribution

$$p(\boldsymbol{x}) = \frac{1}{Z} \exp\{-E(\boldsymbol{x})\}, \qquad (2)$$

for $Z = \sum_{\boldsymbol{x}} \exp\{-E(\boldsymbol{x})\}$. Here $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an undirected graph with $n = |\mathcal{V}|$ nodes, $\boldsymbol{x} = (x_1, \ldots, x_n)$ is a labeling of $\mathcal{V}$ where each x_i can take a finite number of states, and $\theta_i(\cdot)$, $\theta_{ij}(\cdot, \cdot)$ are unary and pairwise potentials. This problem has received a lot of attention as it has applications in many different areas, such as computer vision and natural language processing.

The two standard inference tasks are

- MAP prediction: find a state $\boldsymbol{x}$ of maximal likelihood $p(\boldsymbol{x})$ (or, equivalently, of minimal energy $E(\boldsymbol{x})$).
- Marginalization: compute marginal probabilities of the distribution p, e.g. $p(x_i)$ for some $i \in \mathcal{V}$.

The last decade has seen a tremendous growth in the popularity of the first approach. To a large extent, this can be attributed to the existence of efficient discrete energy minimization algorithms based on the min-cut/max-flow equivalence, such as *graph cuts* (Boykov et al., 2001).

In many situations marginalization is arguably the better inference approach. For example, the Bayes-optimal decision with respect to a Hamming loss consists of thresholding the marginal predictions. Unfortunately, it is much harder to tackle computationally, and this has somewhat hindered its practical use. One popular technique for approximate marginalization is (loopy) sum-product belief propagation (BP). However, BP has the well-known problem that it does not always converge, which makes it unsuitable for some applications. While provably convergent double-loop algorithms for computing fixed points of BP exist, they are typically rather slow in practice, and have not

Extended version of a paper presented at the International Conference on Machine Learning (ICML) workshop on *Inferning: Interactions between Inference and Learning*, Edinburgh, Great Britain, 2012.

found wide spread use, e.g. in the computer vision community.

In this paper we attempt to overcome this by combining the benefits of the two approaches: we compute (approximate) marginals using techniques developed originally for MAP-MRF inference. We do it for the important special case of binary variables, i.e. when $x_i \in \{0,1\}$ for each $i \in \mathcal{V}$. Our goal is thus to compute unary marginals $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ where

$$\alpha_i = p(x_i = 1) \in [0,1]. \tag{3}$$

For this we explore approximation schemes in which $\boldsymbol{\alpha}$ is obtained by minimizing a function of the form

$$f(\boldsymbol{\alpha}) = \sum_{i \in \mathcal{V}} f_i(\alpha_i) + \sum_{(i,j) \in \mathcal{E}} f_{ij}(\alpha_i, \alpha_j). \tag{4}$$

The motivation for using functions of the form (4) comes from the *belief optimization* framework (Welling & Teh, 2001); as shown in (Welling & Teh, 2001), the popular *Bethe free energy* approximation can be expressed in the form (4) where $\alpha_i \in [0,1]$. However, in order to use discrete optimization algorithms, we deviate from (Welling & Teh, 2001) by discretizing the allowed labelings $\boldsymbol{\alpha}$, i.e. we add a restriction $\alpha_i \in D \subset [0,1]$ where D is a fixed finite set. While this obviously limits the accuracy to some extent, it also adds two advantages:

- It allows the use of efficient techniques developed for MAP-MRF inference. One of our results shows a connection between the Bethe free energy and the submodularity theory; this suggests that one can use graph cuts for approximate marginalization.
- It allows to go beyond limitations of the Bethe approximation by *learning* terms $f_i(\cdot)$, $f_{ij}(\cdot,\cdot)$ from training data. Again, the discretization is essential here since it allows to apply standard techniques for structured output learning, such as structured support vector machines.

In this paper we investigate both approaches for setting terms $f_i(\cdot)$, $f_{ij}(\cdot,\cdot)$ – by using the discretized Bethe approximation, and by learning these terms from training data. Our experiments show that when the Bethe approximation does not work, the learning can indeed improve the accuracy of marginalization.

The rest of the paper is organized as follows. In section 2 we review previous work which is relevant as background for our contribution. In section 3 we discuss in detail the two techniques for setting terms of function f, and describe a link between the Bethe free energy and the submodularity theory. We then present experimental results section 4 and discuss conclusions and future work in section 5.

2. Background

Our contribution builds on several, seemingly unrelated earlier concepts. First, in section 2.1 we will review the Bethe free energy approximation. In section 2.2 we give some definitions related to submodularity, and in section 2.3 we discuss two MAP-MRF inferences approaches which are relevant in our setting. Finally, in section 2.4, we introduce structure output learning using structured support vector machines.

2.1. Belief optimization

This section describes the *belief optimization* approach (Welling & Teh, 2001) which inspired our work. Its idea is to express the *Bethe free energy* as a function of unary marginals only by minimizing out pairwise marginals, and then to apply a general-purpose technique for unconstrained non-linear optimization such as a gradient descent.

Bethe free energy. Let $\boldsymbol{\mu} = \{\mu_i(a), \mu_{ij}(a,b)\}$ be the vector of unary and pairwise marginals:

$$\mu_i(a) = p(x_i = a) \qquad \mu_{ij}(a,b) = p(x_i = a, x_j = b) \tag{5}$$

where $i \in \mathcal{V}$, $(i,j) \in \mathcal{E}$ and $a, b \in \{0,1\}$. It is known that $\boldsymbol{\mu}$ can be computed by minimizing the free energy:

$$\min_{\boldsymbol{\mu} \in \mathbb{M}} \; F(\boldsymbol{\mu}) \quad \text{for} \quad F(\boldsymbol{\mu}) = \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle - H(\boldsymbol{\mu}) \tag{6}$$

where $\mathbb{M}$ is the *marginal polytope*, i.e. the set of all realizable marginals $\boldsymbol{\mu}$, and $H(\boldsymbol{\mu})$ is the maximal entropy of distributions with marginals $\boldsymbol{\mu}$. Unfortunately, solving (6) is in general intractable: we cannot evaluate $H(\boldsymbol{\mu})$ efficiently, and furthermore set $\mathbb{M}$ is characterized by exponentially many inequalities. One popular approach is to replace $\mathbb{M}$ with the *local polytope*

$$\mathbb{L} = \left\{ \boldsymbol{\mu} \geq 0 \;\middle|\; \begin{array}{ll} \mu_{ij}(a,0) + \mu_{ij}(a,1) = \mu_i(a) & \forall (i,j), a \\ \mu_{ij}(0,b) + \mu_{ij}(1,b) = \mu_j(b) & \forall (i,j), b \\ \mu_i(0) + \mu_i(1) = 1 & \forall i \end{array} \right\} \tag{7}$$

and also replace the entropy term $H(\boldsymbol{\mu})$ with its *Bethe approximation*

$$H^{\texttt{Bethe}}(\boldsymbol{\mu}) = \sum_{i \in \mathcal{V}} (1 - n_i) H(\boldsymbol{\mu}_i) + \sum_{(i,j) \in \mathcal{E}} H(\boldsymbol{\mu}_{ij}) \tag{8}$$

where n_i is the degree of node i, i.e. the number of incident edges in $(\mathcal{V}, \mathcal{E})$. We then arrive at the problem of minimizing the *Bethe free energy* over the local

polytope:

$$\min_{\boldsymbol{\mu}\in\mathbb{L}} \; F^{\texttt{Bethe}}(\boldsymbol{\mu}) \quad \text{for} \quad F^{\texttt{Bethe}}(\boldsymbol{\mu}) = \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle - H^{\texttt{Bethe}}(\boldsymbol{\mu}) \tag{9}$$

The problem of minimizing the Bethe free energy has received a considerable attention. Most popular is the (loopy) belief propagation algorithm (BP) algorithm, but unfortunately BP does not always converge. More exactly, it is known that there is a one-to-one correspondence between fixed points of BP and stationary points of the Bethe free energy (Yedidia et al., 2000). Furthermore, *stable* fixed points of BP correspond to local *minima* of the Bethe free energy, but the converse is not necessarily true (Heskes, 2002). Provably convergent double-loop algorithms (Heskes, 2002; Yuille, 2002) exist, but they are often rather slow in practice.

Belief optimization. A different approach was proposed by Welling and Teh (Welling & Teh, 2001). They observed that for given vector $\boldsymbol{\alpha} \in [0,1]^n$ one can derive a closed-form expression for a vector $\boldsymbol{\mu} \in \mathbb{L}$ that minimizes $F^{\texttt{Bethe}}(\boldsymbol{\mu})$, subject to additional constraints $\mu_i(1) = \alpha_i$; this optimal solution is given by

$$\mu_i(0) = 1 - \alpha_i \quad \mu_i(1) = \alpha_i \tag{10a}$$

$$\begin{array}{ll} \mu_{ij}(0,0) = \xi_{ij} + 1 - \alpha_i - \alpha_j & \mu_{ij}(0,1) = \alpha_j - \xi_{ij} \\ \mu_{ij}(1,0) = \alpha_i - \xi_{ij} & \mu_{ij}(1,1) = \xi_{ij} \end{array} \tag{10b}$$

where

$$\xi_{ij} = \frac{1}{2\beta_{ij}} \left(Q_{ij} - \sqrt{Q_{ij}^2 - 4\beta_{ij}(1+\beta_{ij})\alpha_i\alpha_j} \right) \tag{10c}$$

$$\begin{aligned} \beta_{ij} = \exp\{ & -\theta_{ij}(0,0) + \theta_{ij}(0,1) \\ & + \theta_{ij}(1,0) - \theta_{ij}(1,1) \} - 1 \end{aligned} \tag{10d}$$

$$Q_{ij} = 1 + \beta_{ij}(\alpha_i + \alpha_j) \tag{10e}$$

Plugging $\boldsymbol{\mu}$ into (9) leads to the problem of minimizing a function of the form (4) over $\boldsymbol{\alpha} \in [0,1]^n$. This allows the use of techniques for unconstrained non-linear optimization such as a gradient descent (Welling & Teh, 2001). However, because (4) typically has many local minima, this approach alone is also not sufficient to solve the marginalization problem.

2.2. {Sub,super}modular functions

Let $D \subseteq [0,1]$ be a totally ordered set. A function $g : D^m \to \mathbb{R}$ is called *submodular* on D if

$$g(\boldsymbol{x} \wedge \boldsymbol{y}) + g(\boldsymbol{x} \vee \boldsymbol{y}) \leq g(\boldsymbol{x}) + g(\boldsymbol{y}) \tag{11}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in D^m$, where $\wedge, \vee$ denote the componentwise min and max operations, respectively. A function g is called *supermodular* if its negative is submodular. We now introduce the following class of functions.

Definition 1. *A function $f : D^n \to \mathbb{R}$ of the form (4) is called* {sub,super}modular *if each term $f_{ij}(\cdot,\cdot)$ is either submodular on D or supermodular on D.*

Such functions will play an important role in this paper: as we will show in Section 3, any function f obtained from the Bethe free energy satisfies the condition of definition 1.

2.3. Discrete energy minimization

For MAP-MRF inference, i.e. minimizing the function (4) over a discrete domain D^n, many more methods have been developed than for marginalization. In this section we concentrate on two techniques that solve linear programming (LP) relaxations of the problem. Following (Kohli et al., 2008), we call them LP-1 and LP-2.

LP-1 This is the most frequently used relaxation for MAP inference; it is also known as *Schlesinger's LP*:

$$\min_{\boldsymbol{\tau}} \quad \sum_{\substack{i\in\mathcal{V} \\ a\in D}} f_i(a)\boldsymbol{\tau}_i(a) + \sum_{\substack{(i,j)\in\mathcal{E} \\ a,b\in D}} f_{ij}(a,b)\boldsymbol{\tau}_{ij}(a,b) \tag{12a}$$

$$\text{s.t.} \quad \sum\nolimits_{a\in D} \boldsymbol{\tau}_{ij}(a,b) = \boldsymbol{\tau}_j(b) \qquad \forall (i,j), b \tag{12b}$$

$$\sum\nolimits_{b\in D} \boldsymbol{\tau}_{ij}(a,b) = \boldsymbol{\tau}_i(a) \qquad \forall (i,j), a \tag{12c}$$

$$\sum\nolimits_{a\in D} \boldsymbol{\tau}_i(a) = 1 \qquad \forall i \tag{12d}$$

It can be solved with general-purpose LP solvers, e.g. interior point methods, or solved approximately with specialized algorithms that exploit the special structure of the problem, see e.g. (Werner, 2007). Note that for submodular functions this relaxation is tight (Werner, 2007), so all solutions are integral. In general, however, the optimal solution may have fractional entries.

LP-2 This relaxation proposed in (Kohli et al., 2008) is used less frequently in the literature, but as we will see later it is quite appropriate in our context. It assumes that set D is ordered: $D = \{d_1, \ldots, d_K\}$, $d_1 < \ldots < d_K$. LP-2 can be described algorithmically as follows:

1. For each variable $\alpha_i \in D$ introduce $K-1$ binary variables $z_i = (z_{i2}, \ldots, z_{iK})$ with the following correspondence:

$$\begin{aligned} \alpha_i = d_1 &\quad\Leftrightarrow\quad z_i = (0,0,\ldots,0) \\ \alpha_i = d_2 &\quad\Leftrightarrow\quad z_i = (1,0,\ldots,0) \\ &\quad\ldots \\ \alpha_i = d_K &\quad\Leftrightarrow\quad z_i = (1,1,\ldots,1) \end{aligned}$$

This is known as the *Ishikawa representation* (Ishikawa, 2003).
2. Construct function $g(\boldsymbol{z})$ with unary and pairwise terms such that $g(\boldsymbol{z}) = f(\boldsymbol{\alpha})$ if $\boldsymbol{z}$ corresponds to $\boldsymbol{\alpha}$, and $g(\boldsymbol{z}) = \infty$ if $\boldsymbol{z}$ is not a "valid" labeling, i.e. $z_{ik} < z_{ik+1}$ for some i, k. We refer to (Schlesinger & Flach, 2006; Kohli et al., 2008) for details of this construction.
3. Apply the *roof duality* relaxation (Kolmogorov & Rother, 2007) to function $g(\boldsymbol{z})$.

In general, LP-2 is less tight than LP-1, i.e. the lower bound on $\min_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha})$ given by LP-2 is not greater than that of LP-1. However, there are several reasons to use LP-2 in our context due to the following properties:

Theorem 2 ((Kohli et al., 2008)). *(a) If f is a {sub,super}modular function then LP-1 and LP-2 coincide.*
(b) LP-2 can be solved in polynomial time by computing a maximum flow in an appropriately constructed graph.
(c) The LP-2 relaxation possesses the persistency, or partial optimality property. Namely, solving LP-2 gives labelings $\boldsymbol{\alpha}^{\min}$, $\boldsymbol{\alpha}^{\max}$ such that $\boldsymbol{\alpha}^{\min} \leq \boldsymbol{\alpha}^* \leq \boldsymbol{\alpha}^{\max}$ for some optimal solution $\boldsymbol{\alpha}^* \in \arg\min_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha})$. If all terms f_{ij} are submodular then $\boldsymbol{\alpha}^{\min} = \boldsymbol{\alpha}^{\max}$.

Remark 1 Since the number of edges in the graph of (b) grows quadratically with the number of labels, it is best suited for the case of few discretization steps. We conjecture, however, that –at least for {sub,super}modular functions– techniques for solving LP-2 that scale better in practice exist. One possibility could be to use the TRW-S algorithm (Kolmogorov, 2006), as it can be shown using arguments from (Kolmogorov & Wainwright, 2005; Kohli et al., 2008), for {sub,super}modular functions TRW-S converges to the solution of the LP (we omit a proof). Furthermore, the "distance transform" operations in TRW-S for submodular and supermodular edges can be implemented in time linear in the number of labels (Aggarwal et al., 1987).

2.4. Structured Support Vector Machines

Recently developed *structured support vector machines (SSVMs)* (Tsochantaridis et al., 2006), allow the learning of prediction functions $g : \mathcal{X} \to \mathcal{Y}$ between nearly arbitrary input and outputs structures, as long as we are able to efficiently optimize over the output set $\mathcal{Y}$.

SSVMs are based on the principle of *structural risk minimization* (Vapnik, 1998): given a loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ between outputs they aim for a prediction function g that results in mimimal expected Δ-loss between correct and predicted outputs. The prediction function is parameterized as $g(x) = \operatorname{argmax}_{y \in \mathcal{Y}} F(x, y)$ for $F(x, y) = \langle w, \phi(x, y) \rangle$, where $\phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^D$ is a fixed joint feature map. Training the SSVM consists of determining the weight vector $w \in \mathbb{R}^D$ from a training set $\{(x^1, y^1), \ldots, (x^N, y^N)\}$ by minimizing the regularized risk functional

$$\frac{1}{2}\|w\|^2 + C \sum_{\nu=1}^{N} \max_{y \in \mathcal{Y}} \Big\{ \Delta(y^\nu, y) - F(x^\nu, y) + F(x^\nu, y^\nu) \Big\}. \tag{13}$$

The constant $C \in \mathbb{R}$ is a regularization parameter, which typically needs to be determined by a model-selection step. The objective (13) is non-differentiable but convex, and efficient working-set techniques are available for finding its minimum (Joachims et al., 2009; Teo et al., 2010).

3. Discrete Marginals

We now return to the problem of computing discrete marginals for a fixed discretization $D = \{d_1, \ldots, d_K\} \subset [0, 1]$. As stated in the introduction, we would like to compute the marginals by minimizing function $f(\boldsymbol{\alpha}) = \sum_i f_i(\alpha_i) + \sum_{(i,j)} f_{ij}(\alpha_i, \alpha_j)$ over $\boldsymbol{\alpha} \in D^n$. In general, this problem is NP-hard, so we have to resort to an approximation. In this paper we employ the LP-1 relaxation of the energy. Solving it gives a fractional vector $\boldsymbol{\tau}$; we then compute the marginals via

$$\alpha_i = \sum_{d \in D} \boldsymbol{\tau}_i(d) d \tag{14}$$

We emphasize, however, that other techniques for MAP-MRF inference can be used as well, e.g. the LP-2 relaxation.

Below we discuss two ways to set terms $f_i(\cdot)$, $f_{ij}(\cdot, \cdot)$:
• restrict the Bethe free energy from $[0, 1]^n$ to D^n;
• learn $f_i(\cdot)$, $f_{ij}(\cdot, \cdot)$ from training data.
We will assume without loss of generality that function (1) has been converted to the form

$$E(\boldsymbol{x}) = \sum_{i \in \mathcal{V}} \eta_i x_i + \sum_{(i,j) \in \mathcal{E}} \eta_{ij} x_i x_j + const \tag{15}$$

This will be useful for the learning part. Note, coefficients η_i, η_{ij} are uniquely determined from $\boldsymbol{\theta}$.

3.1. Bethe Discrete Marginals

For this case we use terms derived in section 2.1:

$$f_i(\alpha_i) = \eta_i \alpha_i + h_1(\alpha_i) - d_i h_1(\alpha_i) \tag{16a}$$

$$f_{ij}(\alpha_i, \alpha_j) = \eta_{ij} \mu_{ij}(1, 1) + \sum_{a,b \in \{0,1\}} h(\mu_{ij}(a, b)) \tag{16b}$$

where $h(z) = z \log z$, $h_1(z) = h(z) + h(1-z)$ and $\mu_{ij}(\cdot,\cdot)$ are given by (10); in eq. (10d) we have $\beta_{ij} = \exp\{-\eta_{ij}\} - 1$. Note, values $\mu_{ij}(\cdot,\cdot)$ depend on α_i, α_j and η_{ij}. As a result, term $f_i(\cdot)$ depends on η_i while $f_{ij}(\cdot,\cdot)$ depends on η_{ij}; for brevity, we omitted this dependence. We now observe the following.

Theorem 3. *If a term $\theta_{ij}(\cdot,\cdot)$ is submodular (supermodular) on $\{0,1\}$ then the term $f_{ij}(\cdot,\cdot)$ defined by* (16b) *is submodular (supermodular) on* $[0,1]$.

A proof is given in Appendix A. This theorem has several implications. First, it means that for submodular functions $E(\boldsymbol{x})$ we can efficiently compute the global minimum of the Bethe free energy up to a given discretization. This adds to the understanding of the complexity of minimizing the Bethe free energy. Results known so far include various sufficient conditions for the uniqueness of the BP fixed point, e.g. (Mooij & Kappen, 2007; Watanabe & Fukumizu, 2009). However, existing conditions usually break for a sufficiently low temperature, i.e. when the energy is multiplied by some large constant. Furthermore, it is known (Watanabe, 2011) that for most types of graphs (with the exception of trees, single cycles, and several others) there always exist a submodular function $E(\boldsymbol{x})$ with multiple BP fixed points. Also note that for binary submodular functions, the Bethe free energy evaluated at any feasible point always bounds the log partition function, see (Ruozzi, 2012). Theorem 3 implies that the tightest Bethe bound can be up to a given discretization efficiently computed.

For non-submodular functions $E(\boldsymbol{x})$ it is not clear whether the global minimum of the Bethe free energy can be computed efficiently (with or without discretization). However, theorem 3 combined with theorem 2 imply two interesting facts: (i) the standard LP-1 relaxation of the (discretized) Bethe free energy can be computed efficiently via graph cuts, and (ii) the solution of this LP gives intervals $[\alpha_i^{\min}, \alpha_i^{\max}]$ which are guaranteed to contain a global minimum.

Convergence Using continuity of f, we can show some convergence results when the quantization step goes to zero. In the theorem below we denote $\mathcal{X} = [0,1]^n$, $f^* = \min_{\boldsymbol{\alpha}\in\mathcal{X}} f(\boldsymbol{\alpha})$ and $\mathcal{X}^* = \{\boldsymbol{\alpha} \in \mathcal{X} \mid f(\boldsymbol{\alpha}) = f^*\}$. For a point $\boldsymbol{\alpha} \in \mathcal{X}$ and a subset $\mathcal{X}' \subseteq \mathcal{X}$ we also define $\texttt{dist}(\boldsymbol{\alpha}, \mathcal{X}') = \inf_{\boldsymbol{\alpha}'\in\mathcal{X}'} ||\boldsymbol{\alpha} - \boldsymbol{\alpha}'||$ where $||\cdot||$ is the Euclidean norm. (A proof is the suppl. material.)

Theorem 4. *Let $\mathcal{X}_1, \mathcal{X}_2, \ldots$ be a sequence of subsets of $\mathcal{X}$ s.t. $\lim_{k\to\infty} \epsilon_k = 0$ where $\epsilon_k \doteq \max_{\boldsymbol{\alpha}\in\mathcal{X}} \texttt{dist}(\boldsymbol{\alpha}, \mathcal{X}_k)$. Let $\boldsymbol{\alpha}^k$ be a minimizer of $f(\boldsymbol{\alpha})$ over $\boldsymbol{\alpha} \in \mathcal{X}_k$.*
(a) If function $f : \mathcal{X} \to \mathbb{R}$ is continuous then $\lim_{k\to\infty} f(\boldsymbol{\alpha}^k) = f^*$ and $\lim_{k\to\infty} \texttt{dist}(\boldsymbol{\alpha}^k, \mathcal{X}^*) = 0$.
(b) If in addition f is twice continuously differentiable on $(0,1)^n$ and $\mathcal{X}^* \cap (0,1)^n \neq \varnothing$ then $f(\boldsymbol{\alpha}^k) - f^* \leq c \cdot \epsilon_k^2$ for some constant $c > 0$.

A proof is given in Appendix B.

We refer to the method of computing Bethe Discrete Marginals as to BDM.

3.2. Learned Discrete Marginals

The structure of f in Equation (4) and the fact that prediction is performed by minimization suggest a second possibility for obtaining discrete marginals: By learning a suitable f from training data using a structured support vector machine (SSVM) framework (see Section 2.4).

Our goal is to learn a prediction function from binary-valued pairwise MRFs to their marginals, so we set $\mathcal{P}$ to be the set of binary pairwise MRFs as defined in Section 1. For $\mathbb{L}$, we choose a construction that is *over-generating* in the sense of Finley and Joachims (Finley & Joachims, 2008), namely the set of all possible output of the discrete marginalization step, i.e. the set of vectors $\boldsymbol{\tau} = (\boldsymbol{\tau}_i)_{i\in\mathcal{V}} \oplus (\boldsymbol{\tau}_{ij})_{(i,j)\in\mathcal{E}}$, where $\boldsymbol{\tau}_i \in [0,1]^{|D|}$ and $\boldsymbol{\tau}_{ij} \in [0,1]^{|D|\times|D|}$ fulfill the constraints of LP-1, and $\oplus$ indicates the concatenation of vectors. In particular, this set contains any discrete value $a \in D$ for any $i \in \mathcal{V}$ through their corresponding integer-valued indicator vectors. For any $\boldsymbol{\tau} = (\boldsymbol{\tau}_i)_i \oplus (\boldsymbol{\tau}_{ij})_{ij}$ and $\boldsymbol{\tau}' = (\boldsymbol{\tau}'_i)_i \oplus (\boldsymbol{\tau}'_{ij})_{ij}$ we set as loss function

$$\Delta(\boldsymbol{\tau}, \boldsymbol{\tau}') = \sum\nolimits_{i\in\mathcal{V}} |\sum\nolimits_{d\in\mathcal{D}} (\boldsymbol{\tau}_i(d)d - \boldsymbol{\tau}'_i(d)d)|, \quad (17)$$

i.e. we penalize mistakes in the unary predictions $\boldsymbol{\tau}_i$ proportionally to their strength. Note that for non-fractional $\boldsymbol{\tau}_i$, the inner sum is the L^1-distance between the corresponding marginals, making Δ compatible with earlier work that had to analyze the quality of predicted marginals (Mooij, 2010; Welling, 2004).

For any input MRF, p, with node degrees n_i, unary weights η_i and pairwise weights η_{ij}, and for any output $\boldsymbol{\tau} = (\boldsymbol{\tau}_i)_{i\in\mathcal{V}} \oplus (\boldsymbol{\tau}_{ij})_{(i,j)\in\mathcal{E}}$, let $\phi_1 = \sum_{i\in\mathcal{V}} \boldsymbol{\tau}_i \psi_i^\top$ for $\psi_i = (\eta_i, n_i, 1)^\top \in \mathbb{R}^3$, and $\phi_2 = \sum_{(i,j)\in\mathcal{E}} \boldsymbol{\tau}_{ij} \psi_{ij}^\top$, where $\psi_{ij} \in \{0,1\}^{K'}$ denotes the indicator vector of discretizing η_{ij} into a set of predefined values, $D' = \{d'_1, \ldots, d'_{K'}\} \subset \mathbb{R}$. We form the SSVM's joint feature function as, $\phi(p, \boldsymbol{\tau}) = \text{vec}(\phi_1) \oplus \text{vec}(\phi_2)$, where $\text{vec}(\cdot)$ denotes row-major order vectorization of a matrix. The joint feature map $\phi(p, \boldsymbol{\tau})$, and thereby also the SSVM quality function $F(p, \boldsymbol{\tau}) = \langle w, \phi(p, \boldsymbol{\tau})\rangle$, are linear in $\boldsymbol{\tau}$. Therefore the SSVM prediction, $\text{argmax}_{\boldsymbol{\tau}} F(p, \boldsymbol{\tau})$, as well as the loss-augmented prediction steps needed during training,$\text{argmax}_{\boldsymbol{\tau}} \Delta(\boldsymbol{\tau}, \boldsymbol{\tau}') +$

$F(p, \boldsymbol{\tau})$ can be performed using LP-1. Note that our construction generalizes the BDM situation: for a suitably chosen weight vector, $F(p, \boldsymbol{\tau})$ becomes the Bethe discrete marginal function f (or rather its negative), up to quantization of η_{ij}.

For any regularization parameter C, we can now train an SSVM to obtain a weight vector w such that allows prediction of discrete marginals by maximizing $F(p, \boldsymbol{\tau})$ for fixed p, or equivalently, minimizing $f(\boldsymbol{\tau}) = -\langle w, \phi(p, \boldsymbol{\tau})\rangle$. Note that the learned f is only designed to yield good marginal predictions when minimized over. It is not necessarily a good approximation of the free energy (6), and it might also differ significantly from the Bethe free energy (9). From this additional flexibility one can expect better predicted marginals compared to the Bethe case, especially when the true free energy and the Bethe free energy differ significantly. Further insight comes from the fact that in the objective (13), any prediction $\boldsymbol{\tau}$ that contains fractional values is penalized by the loss term $\Delta(\boldsymbol{\tau}, \boldsymbol{\tau}')$, because the $\boldsymbol{\tau}$ come from the ground truth of the training data and are therefore non-fractional. Consequently, the SSVM will try to learn a weight vector that results in few fractional solutions if minimized, and this can also be expected to lead to overall better discrete marginal predictions. We refer to the method of computing Learned Discrete Marginals as to LDM.

4. Empirical Comparison

We compare BDM, LDM and BP as a baseline.

4.1. Datasets

For the comparison we generated datasets with six distribution prototypes. We recall that a distribution in equation (2) is specified by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the number of graph nodes is denoted by the scalar n, $n = |\mathcal{V}|$, and by the energy function $E(\mathbf{x})$ in equation (1).

We chose two graph prototypes and three energy prototypes. Their combinations yield our six prototypes of distributions. We chose complete graphs K_n and $n \times n$ lattice graphs L_n with 4-neighborhood as prototypes of the graph $\mathcal{G}$. We chose three prototypes of the energy function $E(\mathbf{x})$ that we call the Ising energy $E_{\text{Ising}}(\mathbf{x})$, the easy energy $E_{\text{easy}}(\mathbf{x})$ and the hard energy $E_{\text{hard}}(\mathbf{x})$. The unary potentials of the energies were parameterized as $[-\theta_i(0), -\theta_i(1)] = c_1[0, 1]$ and the pairwise potentials of the energies were parameterized as $[-\theta_{ij}(0,0), -\theta_{ij}(1,0), -\theta_{ij}(0,1), -\theta_{ij}(1,1)] = c_2[0, -0.5, -0.5, 0]$. For each prototype we chose different values of the scalar parameters c_1 and c_2. For the Ising energy we sample the parameter c_1 uniformly from the closed interval $[-2, 2]$ and we set the parameter c_2 to the value 1. For the easy energy we sample the parameter c_1 uniformly from the closed interval $[-2, 2]$ and the parameter c_2 uniformly from the set $\{-1, 1\}$. For the hard energy we sample the parameter c_1 uniformly from the closed interval $[-1, 1]$ and the parameter c_2 uniformly from the set $\{-4, 4\}$. We obtain six distribution prototypes (K_n,E_{Ising}), (L_n,E_{Ising}), (K_n,E_{easy}), (L_n,E_{easy}), (K_n,E_{hard}), (L_n,E_{hard}).

For each distribution prototype we generated a dataset of 200 pairs of distributions $(\mathcal{G}, E)$ and of the corresponding true singleton marginals $\boldsymbol{\alpha}$ computed by the junction tree algorithm. We used the junction tree implemented in the libDAI software. We split each data set randomly into two disjoint sets, namely the training set of 150 instances and the test set of 50 instances. To be able to repeat a particular experiment multiple times, we for each dataset specify five such random splits.

4.2. Experiments

For each dataset and test set we evaluated marginals of distributions with BDM, LDM and BP as baseline. We evaluated both types of discrete marginals by solving the LP-1 with interior-point methods implemented in the Mosek software. We learned parameters of the discrete energy specified in section 3.2 from the training set using the cutting plane algorithm for SSVMs implemented in the SVMstruct software. For the discretization D' of the parameter $\boldsymbol{\eta}_{ij}$ we used two discrete levels, namely one level for the parameter being positive and one level for the parameter being negative. During learning we solve the loss augmented LP-1 with interior-point methods implemented in the Mosek software. To evaluate the BP marginals we used the implementation in the libDAI software.

For each method and test set we compute the empirical mean l_1-distance of a predicted singleton marginal from the true singleton marginal. We refer to this quantity as to the l_1-error or simply as to the error. For BDM and LDM we used 1, 2, 4 and 8 equidistant discrete marginal values. This means that for 4 discrete levels we used the discretization $D = \{d_1, d_2, d_3, d_4\} = \{0.2, 0.4, 0.6, 0.8\} \subset [0, 1]$. We repeat each experiment five times with random dataset splits into the training set and the test set. We report the mean and the standard deviation computed from the five repetitions.

4.3. Results

In figures 1, 2, 3, 4 we report our results for six distribution prototypes, for BDM, LDM, BP, and for four

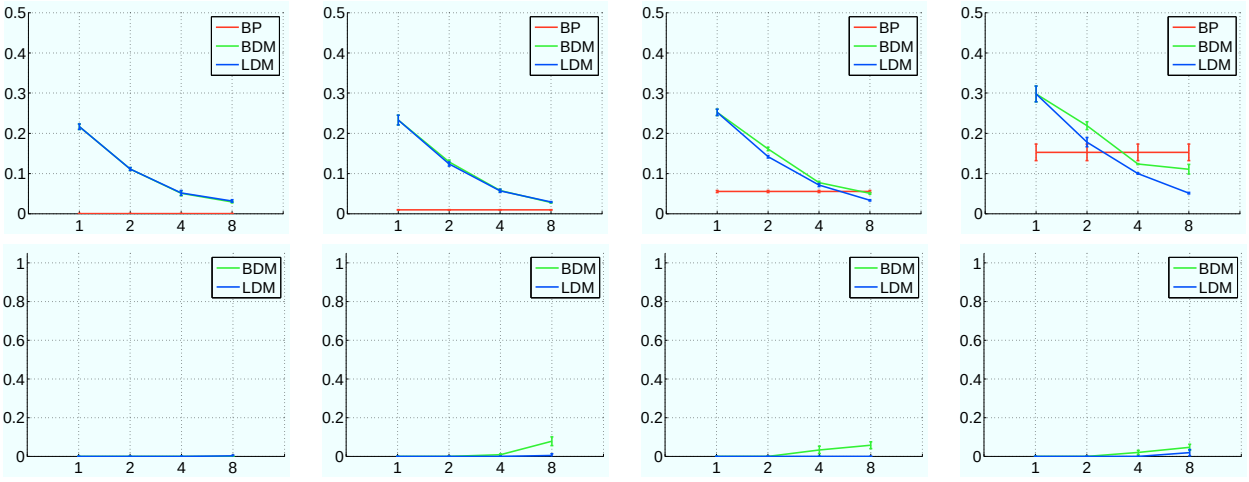

Figure 1. Complete graph, Ising energy. Top row: Mean error. Bottom row: Portion of fractional singleton marginals. Columns from left to right: Distributions with two, four, six and eight variables.

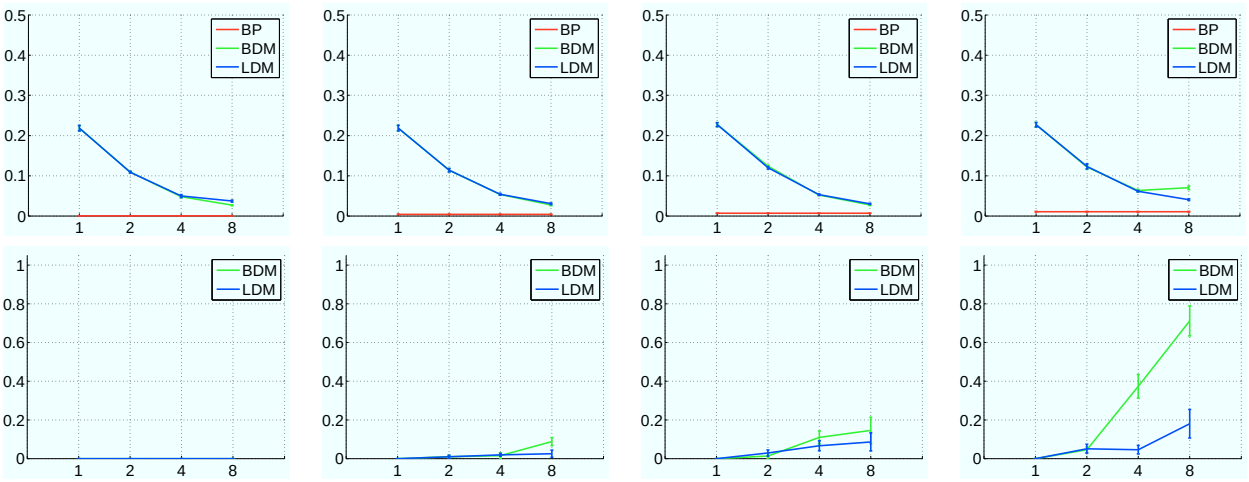

Figure 2. Complete graph, easy energy. Top row: Mean error. Bottom row: Portion of fractional singleton marginals. Columns from left to right: Distributions with two, four, six and eight variables.

discretizations of BDM and LDM. Top rows in figures 1, 2, 3, 4 report the mean errors. Bottom rows in figures 1, 2, 3, 4 report the portion of singleton marginal variables, labelings of which have been identified as fractional in our experiments. We say that a value is fractional if its distance from 0 or 1 is greater than 10^{-5}.

In figures 1, 2, 3 we report results on datasets with distributions specified by the complete graphs. In figure 1 we report results on a dataset with distributions specified by the Ising energies. Columns from left to right report datasets of distributions with two, four, six and eight variables. In figure 2 we report results on a dataset with distributions specified by the easy energies. Columns from left to right report datasets of distributions with two, four, six and eight variables. In figure 3 we report results on a dataset with distributions specified by the hard energies. Columns from left to right report datasets of distributions with three, four, five and six variables.

In figure 4 we report results on datasets with distributions specified by the lattice graphs. Columns from left to right report datasets of distributions with the Ising, the easy, the hard and the hard energies. Columns from left to right report datasets of distributions with 64, 64, four and nine variables.

4.4. Discussion of the Results

Let us recall that for the Ising (submodular) energy the LP-1 in the BDM is tight and hence solutions are ex-

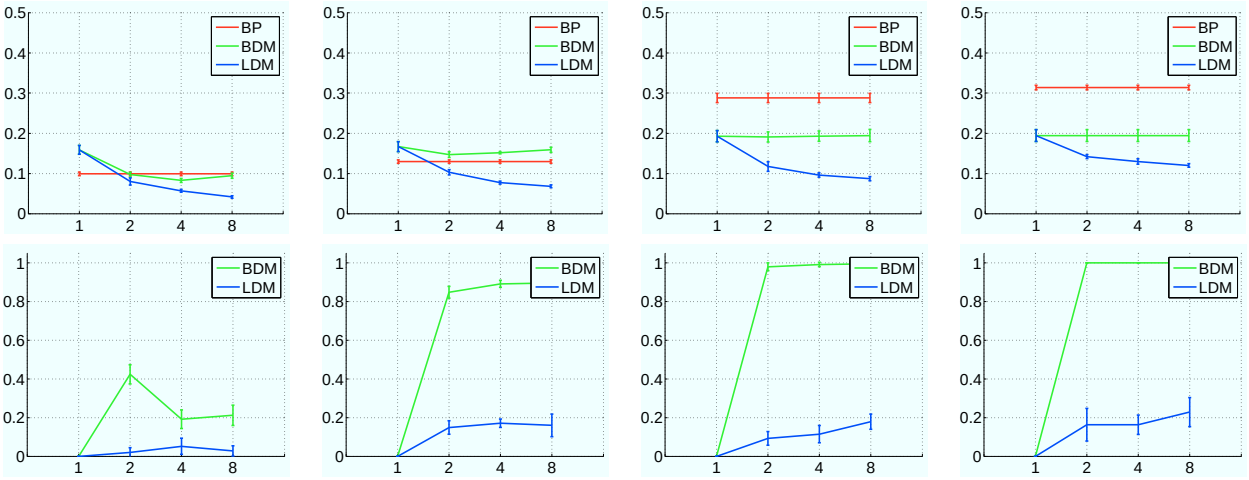

Figure 3. Complete graph, hard energy. Top row: Mean error. Bottom row: Portion of fractional singleton marginals. Columns from left to right: Distributions with three, four, five and six variables.

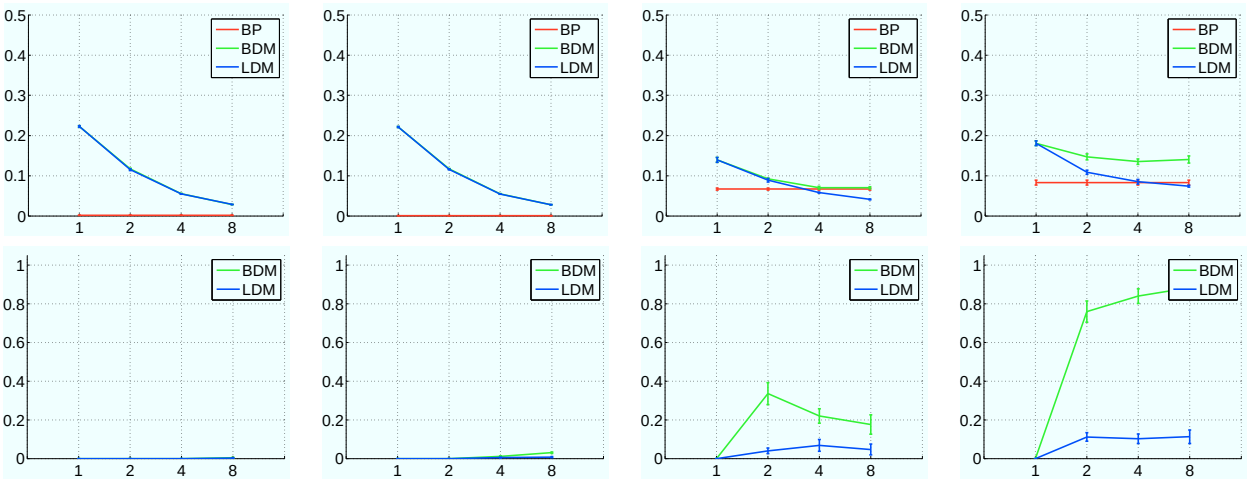

Figure 4. Lattice graph, Ising/easy/hard energy. Top row: Mean error. Bottom row: Portion of fractional singleton marginals. Columns from left to right: Distributions specified by the Ising, the easy, the hard and the hard energies. Columns from left to right: Distributions with 64, 64, four and nine variables.

pected to be non-fractional. The portion of fractional singleton marginals in the bottom row in figure 1 is not zero due to the fact that the employed interior-point method has not always converged to sufficient precision before it has reached its predefined maximum number of iterations. This resulted in small portion of the returned values exceeding slightly the threshold distance of 10^{-5} from 0 or 1. The bottom row of figure 1 shows that the LDM has learned an objective that avoids fractional solutions. In summary the bottom row shows that both BDM and LDM are in this case based on an objective function that practically avoids fractional solutions. This means that we for both objectives find global minimizers. In the top row of figure 1 we observe that LDM significantly reduces the error of BDM as we increase the number of discretization levels and the number of variables in the distribution. We argue that this is an indication of the ability of LDM to overcome some of the limitations of the Bethe approximation.

The BP error in the top row of figure 1 is not 0 possibly due to the in general wrong objective. And since we proved the convergence of the BDM objective to the Bethe free energy, we do not expect the BDM error to converge in that case to 0 either. The BDM error will not necessarily even converge to the BP error due to possibly local fixed point of the BP (Watanabe, 2011) and due to possibly BP not having converged at all.

The easy energy and the hard energy are cases where

the LP-1 in BDM is no longer tight. The bottom row in figure 2 confirms our expectation of observing some portion of fractional solutions. However the proportion suggests that in this case we can still solve the problem with some success. On the other hand the bottom row in figure 3 shows a significant portion of fractional solutions. Small BP error in the top row in figure 2 suggests that the Bethe free energy is in this case an appropriate objective that can also by minimized by BP. The top row confirms our expectation of BDM also converging to a small error as the number of discretizations is increased. On the contrary portions of fractional solutions and errors in figure 3 suggest that the hard energy poses a difficult problem both for BDM and BP. Finally we observe that even in the hard energy cases LDM consistently learned an objective that relative to BDM yields considerably smaller portion of fractional solutions and that increasing the number of discrete levels leads to lower marginal error.

4.5. Conclusion of the Empirical Comparison

In our experiments the LDM seems to have succeeded in achieving two goals. First in our experiments we observe that LDM consistently learns an objective the minimization of which yields significantly less fractional solutions than BDM. Our observation adheres to the observation made also by authors in (Finley & Joachims, 2008). Second in our experiments we observe that for the adopted discretizations LDM learns an objective the minimization of which yields marginals that approach true marginals as the number of discretizations is increased.

5. Conclusions and future work

We introduced the Discrete Marginals approach, in which the approximate marginals are obtained by minimizing an objective function of discrete variables with unary and pairwise terms. This allows the use of techniques developed for MAP-MRF inference and learning. Experiments suggest that if BP does not perform well, learning the suitable function from training data may have significant benefits.

Graphical models with energy $E(\boldsymbol{x}) = \sum_i \eta_i x_i + \sum_{(i,j)} \eta_{ij} x_i x_j$ of binary variables appear frequently in various applications, e.g. for binary image segmentation. In many cases researchers restrict themselves to the MAP estimation due to the availability of efficient discrete optimization algorithms (graph cuts). Our work opens the possibility of going beyond MAP in such cases: DM can produce approximate marginals while still using discrete optimization algorithms. It is generally accepted that marginals can be very useful: they provide a measure of the uncertainty of the solution and can lead to better predictions for the Hamming loss function. Marginals are also needed for CRF training.

The future work includes several directions. One of them is to replace LP-1 relaxation with LP-2 to allow the use of graph cuts; we conjecture that this would bias the learned function towards being {sub,super}modular. An important question is whether it is possible to learn a function that works for different graph topologies; potentially, this could be achieving by adding features that depend on the graph structure. We also plan to look at how we can learn functions f_i, f_{ij} for finer discretizations; potential techniques include interpolation from a coarser discretization and penalizing non-smooth functions.

Appendix A: Proof of theorem 3

The theorem will follow from

Lemma 5. *Denote* $\Delta = \theta_{ij}(0,0) - \theta_{ij}(0,1) - \theta_{ij}(1,0) + \theta_{ij}(1,1)$. *For each* $\alpha_i, \alpha_j \in (0,1)$ *there holds* $\mathtt{sign} \frac{\partial^2 f_{ij}(\alpha_i,\alpha_j)}{\partial \alpha_i \partial \alpha_j} = \mathtt{sign}\, \Delta$.

Proof. We will use the following notation. First, we assume that $i = 1$ and $j = 2$. We also introduce the third "dummy" variable x_3 whose value is always 1. Symbol $\boldsymbol{x}$ will denote a labeling (x_1, x_2, x_3) where $x_1, x_2 \in \{0,1\}$ and $x_3 = 1$. For such labeling $\boldsymbol{x}$ we denote $\theta(\boldsymbol{x}) = \theta(x_1, x_2, 1) = \theta_{ij}(x_1, x_2)$. It is easy to see that $f_{ij}(\alpha_i, \alpha_j) = g(\alpha_i, \alpha_j, 1)$ where function g is defined by

$$g(\boldsymbol{\alpha}) = \min_{\boldsymbol{\mu}} \sum_{\boldsymbol{x}} [\theta(\boldsymbol{x})\mu(\boldsymbol{x}) + h(\mu(\boldsymbol{x}))] \quad (18a)$$

$$\text{s.t.} \sum_{\boldsymbol{x}: x_i = 1} \mu(\boldsymbol{x}) = \alpha_i \quad \forall i \in \{1,2,3\} \quad (18b)$$

The Lagrangian of this constrained optimization problem is

$$L_{\boldsymbol{\alpha}}(\boldsymbol{\mu}, \boldsymbol{\lambda}) = \sum_{\boldsymbol{x}} [(\theta(\boldsymbol{x}) - \langle \boldsymbol{x}, \boldsymbol{\lambda} \rangle)\mu(\boldsymbol{x}) + h(\mu(\boldsymbol{x}))] + \langle \boldsymbol{\alpha}, \boldsymbol{\lambda} \rangle$$

Denote $G(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = \min_{\boldsymbol{\mu}} L_{\boldsymbol{\alpha}}(\boldsymbol{\mu}, \boldsymbol{\lambda})$. It is easy to check that

$$G(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = -\sum_{\boldsymbol{x}} \mu_{\boldsymbol{\lambda}}(\boldsymbol{x}) + \langle \boldsymbol{\alpha}, \boldsymbol{\lambda} \rangle \quad (19a)$$

$$\mu_{\boldsymbol{\lambda}}(\boldsymbol{x}) = \exp\{-\theta(\boldsymbol{x}) + \langle \boldsymbol{x}, \boldsymbol{\lambda} \rangle - 1\} \quad (19b)$$

Since the objective (18a) is strictly convex and constraints (18b) are linear, we have strong duality:

$$g(\boldsymbol{\alpha}) = \max_{\boldsymbol{\lambda}} \min_{\boldsymbol{\mu}} L_{\boldsymbol{\alpha}}(\boldsymbol{\mu}_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}} G(\boldsymbol{\alpha}, \boldsymbol{\lambda})$$

We define $D_i(\boldsymbol{\alpha},\boldsymbol{\lambda}) = \frac{\partial G(\boldsymbol{\alpha},\boldsymbol{\lambda})}{\partial \lambda_i}$. Let $\boldsymbol{\lambda}(\boldsymbol{\alpha})$ be the value of $\boldsymbol{\lambda}$ that maximizes $G(\boldsymbol{\alpha},\boldsymbol{\lambda})$, then $D_i(\boldsymbol{\alpha},\boldsymbol{\lambda}) = \frac{\partial G(\boldsymbol{\alpha},\boldsymbol{\lambda})}{\partial \lambda_k} = 0$ at $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\boldsymbol{\alpha})$. Using this fact, we get

$$\frac{\partial g}{\partial \alpha_i} = \frac{\partial G(\boldsymbol{\alpha},\boldsymbol{\lambda}(\boldsymbol{\alpha}))}{\partial \alpha_i} = \frac{\partial G}{\partial \alpha_i} + \sum_k \frac{\partial G}{\partial \lambda_k}\frac{\partial \lambda_k}{\partial \alpha_i} = \frac{\partial G}{\partial \alpha_i} = \lambda_i$$

Thus, $\frac{\partial^2 g}{\partial \alpha_i \partial \alpha_j} = \frac{\partial \lambda_i}{\partial \alpha_j}$. Differentiating equation $D_i(\boldsymbol{\alpha},\boldsymbol{\lambda}(\boldsymbol{\alpha})) = 0$ gives

$$\frac{\partial D_i(\boldsymbol{\alpha},\boldsymbol{\lambda}(\boldsymbol{\alpha}))}{\partial \alpha_k} = \frac{\partial D_i}{\partial \alpha_k} + \sum_j \frac{\partial D_i}{\partial \lambda_j}\frac{\partial \lambda_j}{\partial \alpha_k} = 0 \qquad (20)$$

We have $\frac{\partial D_i}{\partial \alpha_k} = \delta_{\{i=k\}}$ and $\frac{\partial D_i}{\partial \lambda_j} = C_{ij} \doteq \frac{\partial^2 G}{\partial \lambda_i \partial \lambda_j}$. Thus, equation (20) for $k = 1,2,3$ leads to

$$I + C\frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{\alpha}} = 0 \quad \Rightarrow \quad \frac{\partial \lambda_i}{\partial \alpha_j} = -B_{ij}, \;\; B = C^{-1}$$

We showed that $\frac{\partial^2 g}{\partial \alpha_1 \partial \alpha_2} = \frac{\partial \lambda_1}{\partial \alpha_2} = -B_{12}$. By the Cramer's rule

$$\frac{\partial^2 g}{\partial \alpha_1 \partial \alpha_2} = -B_{12} = \frac{1}{\det C}\det\begin{pmatrix} C_{21} & C_{23} \\ C_{31} & C_{33} \end{pmatrix}$$

From now on we fix $\boldsymbol{\alpha}$ with $\alpha_1, \alpha_2 \in (0,1)$, $\alpha_3 = 1$ and denote $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\boldsymbol{\alpha})$, $\boldsymbol{\mu} = \boldsymbol{\mu}_{\boldsymbol{\lambda}}$. We have

$$C_{ij} = \frac{\partial^2 G(\boldsymbol{\alpha},\boldsymbol{\lambda})}{\partial \alpha_i \partial \alpha_j} = -\sum_{\boldsymbol{x}: x_i = x_j = 1} \mu(\boldsymbol{x})$$

Note that $\sum_{\boldsymbol{x}} \mu(\boldsymbol{x}) = \alpha_3 = 1$. Thus, $C_{ij} = -\mathbb{E}_{\boldsymbol{\mu}}[x_i x_j]$ implying that C is a negative definite matrix; this means that $\det C < 0$. Denote $p_{ab} = \mu(a,b,1)$, then

$$\det\begin{pmatrix} C_{21} & C_{23} \\ C_{31} & C_{33} \end{pmatrix} = \det\begin{pmatrix} \mathbb{E}_{\boldsymbol{\mu}}[x_1 x_2] & \mathbb{E}_{\boldsymbol{\mu}}[x_2] \\ \mathbb{E}_{\boldsymbol{\mu}}[x_1] & 1 \end{pmatrix}$$

$$= \det\begin{pmatrix} p_{11} & p_{01}+p_{11} \\ p_{10}+p_{11} & p_{00}+p_{01}+p_{10}+p_{11} \end{pmatrix} = p_{11}p_{00} - p_{01}p_{10}$$

We showed that

$$\mathtt{sign}\frac{\partial^2 g}{\partial \alpha_1 \partial \alpha_2} = -\mathtt{sign}\left(\frac{p_{11}p_{00}}{p_{01}p_{10}} - 1\right)$$

From (19b) we get $\frac{p_{11}p_{00}}{p_{01}p_{10}} = e^{-\Delta}$, and therefore the above expression equals $-\mathtt{sign}\left(e^{-\Delta} - 1\right) = \mathtt{sign}\,\Delta$. □

Appendix B: Proof of theorem 4

Proof. **(a)** Consider point $\boldsymbol{\alpha}^* \in \mathcal{X}^*$. From theorem's conditions, there exists a sequence $\bar{\boldsymbol{\alpha}}^1, \bar{\boldsymbol{\alpha}}^2, \ldots$ such that $\bar{\boldsymbol{\alpha}}^k \in \mathcal{X}_k$ and $\lim_{k\to\infty} \bar{\boldsymbol{\alpha}}^k = \boldsymbol{\alpha}^*$. We can write

$$f(\boldsymbol{\alpha}^*) \le \lim_{k\to\infty} f(\boldsymbol{\alpha}^k) \le \lim_{k\to\infty} f(\bar{\boldsymbol{\alpha}}^k) = f(\boldsymbol{\alpha}^*)$$

where the last equality holds since f is continuous on $\mathcal{X}$. This proves the first claim of (a).

Let us prove that $\lim_{k\to\infty} \mathtt{dist}(\boldsymbol{\alpha}^k, \mathcal{X}^*) = 0$. Consider $\delta > 0$; we need to show that the set $\mathcal{I}_\delta \doteq \{k \mid \mathtt{dist}(\boldsymbol{\alpha}^k, \mathcal{X}^*) \ge \delta\}$ is finite. Suppose not, there exists a converging subsequence $\boldsymbol{\alpha}^{k(1)}, \boldsymbol{\alpha}^{k(2)}, \ldots$ with $k(\ell) \in \mathcal{I}_\delta$ for $\ell \ge 1$ (since $\mathcal{X}$ is bounded). Let $\bar{\boldsymbol{\alpha}}$ be its limit. The continuity of function $\mathtt{dist}(\cdot, \mathcal{X}^*)$ implies that $\mathtt{dist}(\bar{\boldsymbol{\alpha}}, \mathcal{X}^*) \ge \delta$, while continuity of f and the first claim of the theorem imply that $f(\bar{\boldsymbol{\alpha}}) = f^*$ and so $\bar{\boldsymbol{\alpha}} \in \mathcal{X}^*$. We get a contradiction.

(b) Let $\boldsymbol{\alpha}^*$ be a point in $\mathcal{X}^* \cap (0,1)^n$. Let us pick $\bar{\epsilon} > 0$ such that the set $\bar{\mathcal{X}} = \{\boldsymbol{\alpha} \in \mathbb{R}^n \mid ||\boldsymbol{\alpha} - \boldsymbol{\alpha}^*|| \le \bar{\epsilon}\}$ satisfies $\bar{\mathcal{X}} \subseteq (0,1)^n$. For a unit vector $\boldsymbol{u} \in S^1$ we denote $f_{\boldsymbol{uu}}(\boldsymbol{\alpha})$ to be the second derivative at $\boldsymbol{\alpha}$ in the direction $\boldsymbol{u}$, and $c = \max_{\boldsymbol{\alpha}\in\bar{\mathcal{X}}, \boldsymbol{u}\in S^n} f_{\boldsymbol{uu}}(\boldsymbol{\alpha}) < \infty$.

Since $\lim_{k\to\infty} \epsilon_k = 0$, there exists $\bar{k}$ such that $\epsilon_k \le \bar{\epsilon}$ for all $k > \bar{k}$. Consider fixed $k \ge \bar{k}$. From the definition of ϵ_k, there exists $\boldsymbol{\alpha} \in \mathcal{X}_k$ such that $r = ||\boldsymbol{\alpha} - \boldsymbol{\alpha}^*|| \le \epsilon_k$, and so $\boldsymbol{\alpha} \in \bar{\mathcal{X}}$. Let $\boldsymbol{u}$ be the direction from $\boldsymbol{\alpha}^*$ to $\boldsymbol{\alpha}$. Using the Taylor expansion formula with an explicit residual (along direction $\boldsymbol{u}$), we get

$$f(\boldsymbol{\alpha}^k) \le f(\boldsymbol{\alpha}) = f(\boldsymbol{\alpha}^*) + r f_{\boldsymbol{u}}(\boldsymbol{\alpha}^*) + \frac{1}{2} r^2 f_{\boldsymbol{uu}}(\boldsymbol{\alpha}')$$

for some $\boldsymbol{\alpha}' = \lambda\boldsymbol{\alpha}^* + (1-\lambda)\boldsymbol{\alpha}$, $\lambda \in [0,1]$. Since $\boldsymbol{\alpha}^*$ is a minimizer of f, we get $f_{\boldsymbol{u}}(\boldsymbol{\alpha}^*) = 0$. Therefore, $f(\boldsymbol{\alpha}^k) = f^* + \frac{1}{2} r^2 f_{\boldsymbol{uu}}(\boldsymbol{\alpha}') \le f^* + \frac{1}{2}\epsilon_k^2 c$. This implies part (b). □

References

Aggarwal, A., Klawe, M., Moran, S., Shor, P., and Wilber, R. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2:195–208, 1987.

Boykov, Y., Veksler, O., and Zabih, R. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11), November 2001.

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

Heskes, T. Stable fixed points of loopy belief propagation are minima of the Bethe free energy. In *NIPS*, 2002.

Ishikawa, H. Exact optimization for Markov Random Fields with convex priors. *PAMI*, 25(10):1333–1336, October 2003.

Joachims, T., Finley, T., and Yu, C.N.J. Cutting-plane training of structural svms. *Machine Learning*, 77 (1):27–59, 2009.

Kohli, P., Shekhovtsov, A., Rother, C., Kolmogorov,

V., and Torr, P. On partial optimality in multi-label MRFs. In *ICML*, 2008.

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

Kolmogorov, V. and Rother, C. Minimizing non-submodular functions with graph cuts - a review. *PAMI*, 29(7):1274–1279, 2007.

Kolmogorov, V. and Wainwright, M. On the optimality of tree-reweighted max-product message passing. In *UAI*, 2005.

Mooij, J. and Kappen, H. Sufficient conditions for convergence of the sum–product algorithm. *IEEE Transactions on Information Theory*, 53(12):4422–4437, 2007.

Mooij, J.M. libdai: A free and open source c++ library for discrete approximate inference in graphical models. *JMLR*, 11:2169–2173, 2010.

Ruozzi, N. The bethe partition function of log-supermodular graphical models. *CoRR*, abs/1202.6035, 2012.

Schlesinger, D. and Flach, B. Transforming an arbitrary minsum problem into a binary one. Technical Report TUD-FI06-01, Dresden University of Technology, 2006.

Teo, C.H., Vishwanthan, SVN, Smola, A.J., and Le, Q.V. Bundle methods for regularized risk minimization. *JMLR*, 11:311–365, 2010.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *JMLR*, 6(2):1453, 2006.

Vapnik, V. *Statistical learning theory*. Wiley, New York, 1998.

Watanabe, Y. Uniqueness of belief propagation on signed graphs. In *NIPS*, 2011.

Watanabe, Y. and Fukumizu, K. Graph zeta function in the Bethe free energy and loopy belief propagation. In *NIPS*, 2009.

Welling, M. On the choice of regions for generalized belief propagation. In *UAI*, pp. 585–592, 2004.

Welling, M. and Teh, Y. W. Belief optimization for binary networks: A stable alternative to loopy belief propagation. In *UAI*, 2001.

Werner, T. A linear programming approach to max-sum problem: A review. *PAMI*, 29(7):1165–1179, 2007.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. Generalized belief propagation. In *NIPS*, pp. 689–695, 2000.

Yuille, A. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14:1691–1722, 2002.