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

Presented at the International Conference on Machine Learning (ICML) workshop on *Inferning: Interactions between Inference and Learning*, Edinburgh, Scotland, UK, 2012. Copyright 2012 by the author(s)/owner(s).

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.

2. Discrete energy minimization: Background

2.1. {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

$$g(\boldsymbol{x} \wedge \boldsymbol{y}) + g(\boldsymbol{x} \vee \boldsymbol{y}) \le g(\boldsymbol{x}) + g(\boldsymbol{y}) \tag{5}$$

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.

2.2. LP1 and LP2 relaxations

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*:

$$\min_{\boldsymbol{\tau}} \quad \sum_{\substack{i\in\mathcal{V}\\ a\in D}} f_i(a)\boldsymbol{\tau}_i(a) + \sum_{\substack{(i,j)\in\mathcal{E}\\ a,b\in D}} f_{ij}(a,b)\boldsymbol{\tau}_{ij}(a,b) \tag{6a}$$

$$\text{s.t.} \quad \sum\nolimits_{a\in D} \boldsymbol{\tau}_{ij}(a,b) = \boldsymbol{\tau}_j(b) \qquad \forall (i,j), b \tag{6b}$$

$$\sum\nolimits_{b\in D} \boldsymbol{\tau}_{ij}(a,b) = \boldsymbol{\tau}_i(a) \qquad \forall (i,j), a \tag{6c}$$

$$\sum\nolimits_{a\in D} \boldsymbol{\tau}_i(a) = 1 \qquad \forall i \tag{6d}$$

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 $\boldsymbol{\alpha}^{\min} = \boldsymbol{\alpha}^{\max}$.

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

$$\alpha_i = \sum_{d \in D} \boldsymbol{\tau}_i(d) d \tag{7}$$

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

$$E(\boldsymbol{x}) = \sum_{i \in \mathcal{V}} \eta_i x_i + \sum_{(i,j) \in \mathcal{E}} \eta_{ij} x_i x_j + const \tag{8}$$

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

The fact that the Bethe free energy in the binary case can be written in the form (4) has been observed in (Welling & Teh, 2001). This is achieved by minimizing out pairwise marginals for fixed unary marginals; we refer to (Welling & Teh, 2001) or to a technical report for details. We now observe the following.

Theorem 3. *If a term $\theta_{ij}(\cdot, \cdot)$ is submodular (supermodular) on $\{0, 1\}$ then the term $f_{ij}(\cdot, \cdot)$ that comes from the Bethe free energy is submodular (supermodular) on $[0, 1]$.*

A proof is given in the technical report. 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.

It is possible to show some convergence results when the quantization step goes to zero (see technical report).

We refer to the 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 using a structured support vector machine (SSVM) (Tsochantaridis et al., 2006).

We write $\mathcal{P}$ for set of binary pairwise MRFs and we denote by $\mathbb{L}$ the set of all possible outputs 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. This puts us into an *over-generating* setup in the sense of (Finley & Joachims, 2008): any discretized marginal value has a representation in $\mathbb{L}$ by its corresponding indicator vector but fractional solutions can be represented as well. 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

$$\Delta(\boldsymbol{\tau},\boldsymbol{\tau}') = \sum\nolimits_{i\in\mathcal{V}} \Big|\sum\nolimits_{d\in\mathcal{D}} (\boldsymbol{\tau}_i(d)d - \boldsymbol{\tau}'_i(d)d)\Big|, \tag{9}$$

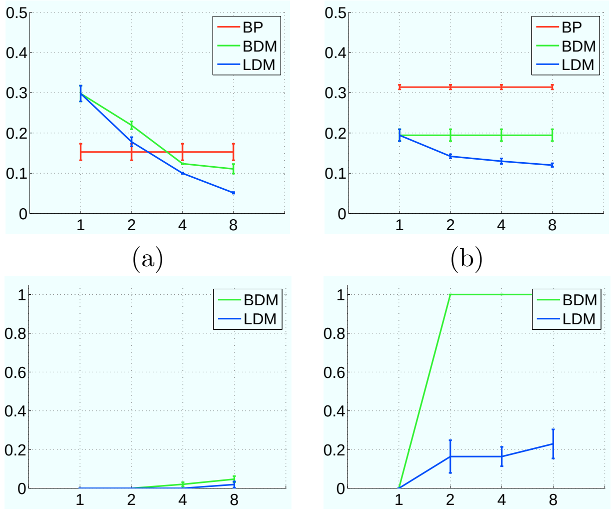
which penalizes mistakes in the unary predictions $\boldsymbol{\tau}_i$ proportionally to their strength. For non-fractional $\boldsymbol{\tau}_i$, the inner sum is the L^1-distance between the corresponding marginals, making Δ compatible with earlier work that had to analyze the quality of predicted 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, $D' = \{d'_1, \dots, d'_{K'}\} \subset \mathbb{R}$. We form the SSVM's joint feature function as, $\phi(p,\boldsymbol{\tau}) = \mathrm{vec}(\phi_1) \oplus \mathrm{vec}(\phi_2)$, where $\mathrm{vec}(\cdot)$ 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, $\mathrm{argmax}_{\boldsymbol{\tau}}\, F(p,\boldsymbol{\tau})$, as well as the loss-augmented prediction steps needed during training,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}. However, the learning aims neither at approximating the Bethe free energy nor the free energy itself. The sole criterion for the SSVM is selecting the energy function that yields good marginal predictions when minimized over. In particular, we can expect the resulting energy to be easier to minimize than the former two, because the overgenerating SSVM framework discourages terms that would result in many fractional solutions when minimized over.

We refer to the Learned Discrete Marginals as to LDM.



(c) (d)

Figure 1. Top row: Mean marginal error. Bottom row: Portion of fractional marginals. Left column: Eight variable distributions with submodular energies. Right column: Six variable distributions with non-submodular energies.

4. Empirical Comparison

We now give a short account of our experiments. For details we refer to the technical report. We generated two datasets with distributions specified over complete graphs. The first dataset, that we report in the left column of figure 1, involves distributions with submodular energies. In this case the LP-1 in BDM is tight and hence solutions are expected to be non-fractional. The second dataset, that we report in the right column of figure 1, involves distributions with non-submodular energies. In this case the LP-1 in BDM is no longer tight. For both datasets we evaluated marginal errors of BDM, LDM and BP as baseline. We report the marginal errors in the top row of figure 1. For BDM and LDM we in the bottom row of figure 1 also report the portion of fractional solutions.

The portion of fractional BDM marginals in figure 1(c) is not zero due to the fact that the employed interior-point method has not always converged to sufficient precision. The figure shows that LDM has learned an objective that yields almost no fractional solutions. Figure 1(d) shows that BDM marginals are all fractional. We observe that LDM learned an objective that yields relatively small portion of fractional solutions.

The BP error in the top row of figure 1 is not 0 possibly due to the in general wrong objective. For the same reason we do not expect the BDM error to converge 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), due to possibly BP not having converged at all or in figure 1(d) due to fractional solutions. In the top row of figure 1 we observe that as we increase the number of discrete levels LDM reduces the error of BDM. We argue that this is an indication of the ability of LDM to overcome some of the limitations of the Bethe approximation.

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.

References

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.

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

Kohli, P., Shekhovtsov, A., Rother, C., Kolmogorov, V., and Torr, P. On partial optimality in multi-label MRFs. In *ICML*, 2008.

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

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.

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

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.