Bethe Learning of Graphical Models via MAP Decoding

Kui Tang
Columbia University

Nicholas Ruozzi
UT Dallas

David Belanger
UMass Amherst

Tony Jebara
Columbia University

Abstract

Many machine learning tasks require fitting probabilistic models over structured objects, such as pixel grids, matchings, and graph edges. Maximum likelihood estimation (MLE) for such domains is challenging due to the intractability of computing partition functions. One can resort to approximate marginal inference in conjunction with gradient descent, but such algorithms require careful tuning. Alternatively, in frameworks such as the structured support vector machine (SVM-Struct), discriminative functions are learned by iteratively applying efficient maximum a posteriori (MAP) decoders. We introduce MLE-Struct, a method for learning discrete exponential family models using the Bethe approximation to the partition function. Remarkably, this problem can also be reduced to iterative (MAP) decoding. This connection emerges by combining the Bethe approximation with the Frank-Wolfe (FW) algorithm on a convex dual objective, which circumvents the intractable partition function. Our method can learn both generative and conditional models and is substantially faster and easier to implement than existing MLE approaches while still relying on the same black-box interface to MAP decoding as SVM-Struct. We perform competitively on problems in denoising, segmentation, matching, and new datasets of roommate assignments and news and financial time series.

1 INTRODUCTION

Learning graphical model parameters using regularized maximum likelihood estimation (MLE) is a ubiquitous problem in machine learning and related fields (Lafferty, 2001). Despite the availability of alternative estimators, MLE remains a primary goal for many practitioners, since it may yield superior predictive accuracy, more interpretable parameters, and quantification of uncertainty.

As the log-likelihood is concave, it can in principle be maximized by gradient ascent. However, this requires repeatedly computing gradients of the log-partition function, which is intractable in general. As a result, a common practice is to use an approximate marginal inference scheme (e.g., loopy belief propagation) to compute a surrogate partition function. However, this *double-loop* approach is slow and difficult to tune: as the inner inference problem performs continuous optimization, one must carefully set convergence thresholds. Moreover, the effect of the resulting inference errors on overall MLE performance is unclear. Further, for many structured objects, such as bipartite matchings, naïve applications of message passing are inefficient, so problem-specific algorithms are necessary (Huang & Jebara, 2009).

A variety of methods have been proposed to avoid the double-loop approach. First, one can interleave inference and learning by dualizing the inner problem to yield an objective jointly convex in parameters and "messages" (Hazan & Urtasun, 2010). This results in a coordinate update scheme for learning. But in practice, some coordinate updates cannot be computed exactly, so one resorts to mixing coordinate and gradient steps. This again requires deriving new updates for different structured problems and carefully tuning convergence thresholds. Second, one can backpropagate gradients through an approximate inference routine (Domke, 2013). Scalability of this method hinges on carefully choosing the number of inner loop iterations to run. Finally, one can avoid likelihoods entirely, and use methods such as the structured perceptron or structured support vector machine (SVM-Struct) that rely only on a repeated calls to a MAP solver (Collins, 2002; Taskar et al., 2004; Tsochantaridis, 2004; Lacoste-Julien, 2013)[1].

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.

[1] In this paper, MAP inference refers to predicting a

MAP-based methods are computationally and theoretically appealing because they consist of simple wrappers around fast and well-understood combinatorial optimization algorithms. They can be quite accurate and are often much faster than the methods above. Also, they offer users an attractive abstraction between the learning algorithm and the specific problem: to apply these methods to a new problem, the user needs only to link to a new MAP solver. We refer to these as *single-loop* techniques, since the black-box inner problem can be solved in polynomial time and does not require convergence thresholds.

To bring the benefits of MAP-based methods to learning probabilistic models, we introduce *MLE-Struct*, a user-friendly approximate MLE procedure that also only requires access to a black-box MAP (or approximate MAP) solver. Beginning with Bethe-style convex free energies, we derive a dual formulation of approximate MLE. The resulting *convex* objective can be efficiently minimized with the Frank-Wolfe (FW) method (Frank & Wolfe, 1956; Jaggi, 2013) using repeated (approximate) MAP calls. Stochastic subsampling (Lacoste-Julien, 2013) further improves scalability. Our method comes with an $O(1/T)$ runtime guarantee on the duality gap. We can also apply FW to perform test-time marginal inference, as first proposed by Sontag & Jaakkola (2007). This enables us to learn and infer exponential-family models for all structured outputs that SVM-Struct can, such as bipartite/general matchings (via max-flow and Blossom algorithms) (Goldberg & Kennedy, 1995; Kolmogorov, 2009), pairwise binary graphical models (via QPBO) (Rother, 2007), and others.

Compared to existing techniques, our method is faster, more reliable, and easier to implement and apply to new problems. It is equally applicable to *unsupervised* problems, where we observe only structured outputs for which we want to learn a probability distribution. The only tuning parameters in our algorithm are a convergence criterion and step-size schedule, for which a simple default suffices. We provide user-friendly code that is already being used in several applied projects.[2] MLE-Struct is comparable in performance to MAP-based training methods, but adds the benefits of probabilistic models. Our method applies equally to supervised and unsupervised problems, where in the latter, we observe only structured outputs and wish to learn a probability distribution. We demonstrate the superiority, both in speed and accuracy, of our test-time inference procedure versus an instance of the Perturb-and-MAP (Li, 2013) framework designed specifically for bipartite matchings. For CRFs, we outperform recent MLE methods (Domke, 2013; Hazan & Urtasun, 2010). Our experiments explore grid CRFs, high treewidth Markov random fields, and conditional exponential family models defined over bipartite and general matchings.

2 BACKGROUND

We consider discrete, log-linear conditional random fields where we observe independent samples $Y^{(1)}, \dots, Y^{(M)}$ from some discrete space $\mathcal{Y}$ as well as feature vectors $X^{(1)}, \dots, X^{(M)} \in \mathcal{X}$ (Lafferty, 2001). The joint distribution factors over a hypergraph $G = (V, \mathcal{A})$, where $\mathcal{A}$ is a collection of subsets of V, as

$$p(Y|X;\theta) = \frac{\exp\Big(\sum_{i\in V}\theta_i\phi_i(X,Y_i) + \sum_{\alpha\in\mathcal{A}}\theta_\alpha\phi_\alpha(X,Y_\alpha)\Big)}{Z(X;\theta)},$$

where $\phi_i(X, Y_i)$ is a vector of sufficient statistics for node i and the vector and $\phi_\alpha(X, Y_\alpha)$ is a vector of sufficient statistics for the hyperedge α. This formalism can also represent non-conditional (unsupervised) models by encoding unique indicator vectors for each node and clique in the $X^{(m)}$. We aim to learn θ by maximizing the ℓ_2-regularized log-likelihood

$$\ell(\theta; X, Y) = \sum_{m=1}^{M}\Big[\sum_{i\in V}\langle\theta_V, \phi_i(X^{(m)}, Y_i^{(m)})\rangle + \sum_{\alpha\in\mathcal{A}}\langle\theta_\mathcal{A}, \phi_\alpha(X^{(m)}, Y_\alpha^{(m)})\rangle - \log Z(X^{(m)};\theta)\Big] - \frac{\lambda}{2}\|\theta\|^2, \quad (1)$$

where the log-partition function is given by

$$\log Z(X^{(m)};\theta) = \log\sum_{Y\in\mathcal{Y}}\Big(\sum_{i\in V}\langle\theta_V, \phi_i(X^{(m)}, Y_i)\rangle + \sum_{\alpha\in\mathcal{A}}\langle\theta_\mathcal{A}, \phi_\alpha(X^{(m)}, Y_\alpha)\rangle\Big). \quad (2)$$

2.1 Convex Free Energy Approximation and Approximate MLE Objective

The central challenge in MLE is computing the log-partition function, $\log Z(X^{(m)}, \theta)$, at each iteration and for each observation m. We approximate this function in order to make learning tractable. We begin with the Bethe free energy, a standard approximation to the Gibbs free energy that is motivated by ideas from statistical physics. The approximation has been generalized to include different *counting numbers* that

discrete output Y given parameters θ and optionally input features X, while MLE learning refers to estimating θ (continuous parameters) given observations.

[2] `https://github.com/kuitang/fwmatch-public`

result in alternative entropy approximations (Weiss, 2007; Wainwright & Jordan, 2008). We focus on the restricted set of counting numbers that yield *convex* reweighted free energies. For reasons that will become clear in the sequel, we introduce a parameterized approximation to $x \log x$.

Definition 2.1. *For η in $[0, 1/2)$, define $g_\eta(x) = x \log x$ for $x \in [\eta, 1]$ and $g_\eta(x) = \eta \log(\eta) + (\log(\eta) + 1)(x - \eta) + \frac{(x-\eta)^2}{2\eta}$ for $x \in [0, \eta)$.*

The *ρ-reweighted free energy* is specified by a polytope approximation $\mathcal{T}$, a hypergraph $G = (V, \mathcal{A})$, an entropy approximation H_ρ^η, a parameter $\eta \in [0, 1/2)$, and a vector of counting numbers (a.k.a reweighting parameters) ρ.

$$\log F_\rho^\eta(\tau, X; \theta) \triangleq E(\tau, X; \theta) - H_\rho^\eta(\tau), \tag{3}$$

where the energy is given by

$$\begin{aligned} E(\tau, X; \theta) \triangleq &- \textstyle\sum_{i \in V} \langle \sum_{Y_i} \tau_i(Y_i) \phi_i(X, Y_i), \theta_V \rangle \\ &- \textstyle\sum_{\alpha \in \mathcal{A}} \langle \sum_{Y_\alpha} \tau_\alpha(Y_\alpha) \phi_\alpha(X, Y_\alpha), \theta_{\mathcal{A}} \rangle \end{aligned}$$

and the entropy approximation is given by

$$\begin{aligned} H_\rho^\eta(\tau) \triangleq &- \textstyle\sum_{i \in V} \sum_{y_i} \big(1 - \sum_{\alpha \supset i} \rho_\alpha\big) g_\eta(\tau_i(y_i)) \\ &- \textstyle\sum_{\alpha \in \mathcal{A}} \sum_{y_\alpha} \rho_\alpha g_\eta(\tau_\alpha(y_\alpha)), \end{aligned} \tag{4}$$

and τ is restricted to lie in the local polytope,

$$\begin{aligned} \mathcal{T} \triangleq &\{\tau \geq 0 : \forall i \in V, \textstyle\sum_{Y_i} \tau_i(Y_i) = 1, \\ &\forall \alpha \in \mathcal{A}, i \in \alpha, Y_i, \textstyle\sum_{Y_{\alpha \setminus \{i\}}} \tau_\alpha(Y_\alpha) = \tau_i(Y_i)\}. \end{aligned} \tag{5}$$

The reweighted log-partition function is then computed by minimizing (3) over $\mathcal{T}$

$$\log Z_\rho^\eta(X; \theta) \triangleq - \min_{\tau \in \mathcal{T}} F_\rho^\eta(\tau, X; \theta). \tag{6}$$

Setting $\rho_\alpha = 1$ for each $\alpha \in \mathcal{A}$ recovers the typical Bethe free energy approximation. The reweighting parameters can always be chosen so that the approximate free energy is convex (Heskes, 2006; Ruozzi & Tatikonda, 2013). For example, tree-reweighted belief propagation (TRW) chooses ρ so that its components correspond to (hyper)edge appearance probabilities of a collection of spanning (hyper)trees (Wainwright & Jordan, 2008). Later, we discuss how employing small $\eta > 0$ yields approximation error, but guarantees fast convergence of our proposed algorithm.

We replace the exact partition function in the MLE objective (1) with a reweighted free energy approximation of the form (3). This results in a *concave-convex* saddle point problem

$$\begin{aligned} \max_\theta \min_{\tau^{(1:M)}} \sum_m \Big[&\sum_{i \in V} \langle \theta_V, \phi_i(X^{(m)}, Y_i^{(m)}) \rangle + \\ &\sum_{\alpha \in \mathcal{A}} \langle \theta_{\mathcal{A}}, \phi_\alpha(X^{(m)}, Y_\alpha^{(m)}) \rangle - \\ &F_\rho^\eta(\tau^{(m)}, X^{(m)}, \theta) \Big] - \frac{1}{2} \|\theta\|^2 . \end{aligned} \tag{7}$$

3 CONVEX APPROXIMATION TO MLE

We now consider a convex *dual* of (7) that yields a new, fast learning algorithm. Since the objective is concave-convex and one set ($\mathcal{T}$) is constrained to a compact domain, we invoke Sion's minimax theorem (Sion, 1958) to swap the max and min operators while preserving equality. With θ on the inside, the optimal θ given fixed $\tau^{(1)}, \ldots, \tau^{(M)} \in \mathcal{T}$ can be found by setting the gradient with respect to θ equal to zero, yielding the *linear* maps

$$\begin{aligned} \theta_V^*(\tau^{(1:M)}) = &\frac{1}{\lambda} \big(\textstyle\sum_m \sum_{i \in V} \big[\phi_i(X^{(m)}, Y_i^{(m)}) \\ &- \textstyle\sum_{Y_i} \tau_i^{(m)}(Y_i) \phi_i(Y_i, X^{(m)}) \big] \big) \end{aligned} \tag{8}$$

$$\begin{aligned} \theta_{\mathcal{A}}^*(\tau^{(1:M)}) = &\frac{1}{\lambda} \big(\textstyle\sum_m \sum_{\alpha \in \mathcal{A}} \big[\phi_\alpha(X^{(m)}, Y_\alpha^{(m)}) \\ &- \textstyle\sum_{Y_\alpha} \tau_\alpha^{(m)}(Y_\alpha) \phi_\alpha(Y_\alpha, X^{(m)}) \big] \big). \end{aligned} \tag{9}$$

Substituting these back into (7) yields the following minimization over the local polytope:

$$\begin{aligned} \min_{\tau^{(1:M)} \in \mathcal{T}} &L_\rho^\eta(\tau^{(1:m)}) \triangleq \\ &\min_{\tau^{(1:M)} \in \mathcal{T}} \frac{1}{2\lambda} \|\theta^*(\tau^{(1:M)})\|^2 - \sum_m H_\rho^\eta(\tau^{(m)}) \end{aligned} \tag{10}$$

This is a convex minimization problem with linear constraints. We have shown that approximate MLE with a convex variational free energy is dual to an approximate max-entropy problem. However, it appears difficult to solve because all training examples are coupled by the quadratic term. Next, we present an algorithm that iteratively solves decoupled per-example problems given access to an approximate MAP solver.

3.1 Frank-Wolfe Algorithm For Maximum Likelihood Learning

The Frank-Wolfe algorithm minimizes a convex function f over linear constraints by iteratively minimizing its linearization over the same constraints (Frank & Wolfe, 1956; Jaggi, 2013). Each linear problem yields a vertex of the constraint set. Then, the iterate is moved one step in the direction of this vertex by a specified schedule. We can write this procedure as

$$s_t = \arg\min_{x \in \mathcal{X}} \langle x, \nabla f(x_{t-1}) \rangle \tag{11}$$

$$x_t = (1 - \gamma_t) x_{t-1} + \gamma_t s_t, \tag{12}$$

where the step-size, γ_t, can be set to $\frac{2}{2+t}$ or obtained via line search (Jaggi, 2013).

For the objective function in (10), each iteration requires computing

$$s_t = \arg\min_{\tau^{(1)},\ldots,\tau^{(M)} \in \mathcal{T}} \langle \tau^{(1:m)}, \nabla L^{\eta}_{\rho}(\tau^{(1:m)}_{t-1})\rangle. \quad (13)$$

This is a linear program over the *local polytope* $\mathcal{T}$, which is tractable. Since the constraints are separable across training examples, (13) decouples into M independent linear programs that can be solved in parallel. Alternatively, this this is a convex optimization problem over separable constraints, we can use block-coordinate Frank-Wolfe (BCFW) (Lacoste-Julien, 2013). BCFW performs the FW iterations over a randomly selected $m \in \{1, \ldots, M\}$ and leaves the remaining coordinates untouched. BCFW requires less work at each iteration, but the asymptotic rate of convergence remains the same (Lacoste-Julien, 2013). We describe both variants of MLE-Struct in Alg. 1. Line search can be accelerated by precomputing quadratic terms, as discussed in D.2.

The FW algorithm converges as $O(\frac{C}{t})$, where C is the objective's *curvature* (Jaggi, 2013). For any $\eta > 0$, the curvature is bounded and therefore Alg. 1 converges.

Theorem 3.1. *Let $|V|$ be the number of nodes in the model, $|\mathcal{A}|$ the number of factors, M the number of samples, R the maximum norm of any feature function ϕ, and $\eta \in (0, \frac{1}{2})$. Alg. 1 converges as $O(C/t)$ to the optimum of* (10)*, with curvature*

$$C < (|V| + |\mathcal{A}|)\, M \left(\frac{C_\rho}{\eta} + \frac{R^2}{\lambda} \right).$$

Here, C_ρ is a constant depending only on ρ and the graph structure of the problem.

The proof and detailed discussion appear in Appendix B. Using $\eta > 0$ results in approximation error for the Bethe-MLE problem but is necessary to use the above convergence guarantee. In Appendix B we discuss a data-dependent heuristic for choosing η. Although the curvature is technically unbounded at $\eta = 0$, experimentally, we have observed that the algorithm always converges, so our results use this setting.

3.2 Frank-Wolfe For Marginal Inference

Many applications require computing marginals at test time. We can use FW to perform marginal inference by maximizing (3) with respect to τ, which is a concave problem suitable for FW. Thus, at both train and test time, we only need to interact with the constraints through a MAP solver.

Algorithm 1 MLE-Struct: Frank-Wolfe Approximate Maximum-Likelihood Learning

```
Input: training data $\{(X^{(m)}, Y^{(m)}\}$, reweighting parameters $\rho \in [0,1]^n$, regularizer $\lambda$, $\eta \in [0, 1/2)$
Output: Approximate maximum likelihood estimate $\theta$.
Initialization: Set each $\tau^{(m)}$ uniformly.
repeat
  for all $m$ in parallel (batch) or $m$ chosen uniformly at random (block) do
    $s^{(m)}_t = \arg\min_{\tau^{(M)} \in \mathcal{T}} \langle \tau^{(m)}, \nabla^{(m)} L^{\eta}_{\rho}(\tau^{(1:m)}_{t-1})\rangle$
    Set $\gamma = \frac{2}{2+t}$ for batch and $\gamma = \frac{2M}{2M+t}$ for block or use line search.
    $\tau_t = (1-\gamma)\tau^{(m)}_{t-1} + \gamma s^{(m)}_t$
  end for
until duality gap $\langle \tau_t - s_t, \nabla L(\tau_t)\rangle < \epsilon$
Set $\theta$ using (8) and (9).
```

4 RELATED WORK

Although FW has been employed for marginal inference (Sontag & Jaakkola, 2007; Belanger, 2013), MAP inference (Schwing, 2014), and max-margin learning (Lacoste-Julien, 2013), our work is the first to apply it to approximate MLE. To do so, we simplify the saddle point problem (7) to a convex dual (10) that would be difficult to solve without FW, since (10) couples all training examples via quadratic terms. Linearization in FW is crucial for decoupling them.

In concurrent work, Krishnan et al. (2015) have introduced the *contraction polytope*, an inner bound to the marginal polytope. They propose two algorithms for approximate inference: one yields an $O(1/T)$ convergence rate but incurs approximation error (similar to MLE-Struct with $\eta > 0$), and the other has no approximation error but only converges at the rate $O(1/\sqrt{T})$. We expect their technique could be applied for learning as well but leave this for future work.

A related method for inference using MAP solvers is Perturb-and-MAP (Li, 2013). In Appendix F, we describe experiments on CRFs for bipartite matchings, demonstrating the favorable accuracy and speed of FW inference against both Perturb-and-MAP and belief propagation (Huang & Jebara, 2009). Another method for bipartite matchings is to obtain unbiased but noisy gradients using a strongly polynomial perfect sampler (Huber & Law, 2008). However, only experiments with graphs up to 20 nodes have been done (Petterson, 2009).

Wainwright (2006) investigated the use of TRW for learning in pairwise binary graphical models. They observe that the parameters learned via TRW are more robust to the addition of new data than those learned by BP, in a theoretically precise manner.

In the context of structured prediction, Kulesza & Pereira (2008) and Finley & Joachims (2008) have investigated potential failure cases of approximate MAP decoding for parameter learning. Our method only approximates the MAP problem by optimizing over the local polytope, a superset of the marginal polytope. This is an overgenerating approximation in the terminology of Finley & Joachims (2008), which has been shown to enjoy robust theoretical properties.

There are a variety of methods to solve (7) directly, all of which compute the partition function iteratively, yielding expensive double-loop algorithms. Ganapathi (2008) followed a maximum entropy (dual) approach using the Bethe entropy approximation (i.e., $\rho = 1$), but avoided convex entropy approximations, instead using the concave-convex procedure. Domke (2013) proposed performing MLE using a small, fixed number of TRW iterations to estimate the gradient. However, if TRW does not quickly converge, then the resulting procedure can diverge, as some of our experiments show. Vishwanathan (2006) proposed improving the convergence in the outer loop using accelerated gradient methods.

Most related to our method are techniques for rendering (7) more friendly by dualizing the inner maximization, yielding an objective that is jointly convex in θ and dual messages (i.e., beliefs) (Hazan & Urtasun, 2010; Meshi, 2010). However, these dual objectives explicitly retain iterates in θ and τ and the algorithms require alternating message passing and gradient steps. In practice, similar to Domke (2013), they rely on running the inner loop for a fixed number of iterations.

Finally, a key advantage of MLE-Struct over the above techniques is its ease of implementation. It interacts with the underlying problem only via an off-the-shelf MAP solver. This allows us to easily prototype applications on various problems without deriving new message-passing updates.

5 EXPERIMENTS

We demonstrate the versatility and competitiveness of MLE-Struct on grid CRFs, high treewidth Markov random fields, and probabilistic models of bipartite and general matchings. For grid CRFs, we compare against Hazan & Urtasun (2010) and Domke (2013), both of which attempt to optimize the same objective (7). MLE-Struct is significantly more scalable than both. For matchings, we compare against discriminative methods and obtain competitive test error. In each case, we use the same generic FW code, with only problem-specific gradient and MAP subroutines. Appendix A contains details and additional results.

5.1 Grid CRFs

For pairwise binary CRFs, MAP inference is intractable, but we can efficiently solve the LP relaxation over the local polytope using QPBO (Rother, 2007). Solutions over the local polytope are known to yield accurate predictions (Wainwright & Jordan, 2008). Of the related methods in Section 4, only Domke (2013) provides experimental results on the same scale as the problems below, with up to 40K nodes.

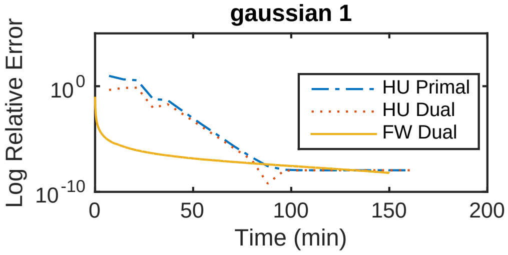
We study the binary denoising dataset of Kumar & Hebert (2003), which is the largest problem solved by Hazan & Urtasun (2010), and the Weizmann horses (Borenstein & Ullman, 2002), which is the largest binary problem solved by Domke (2013). We configure all algorithms to solve the same objective (7). Some methods solve the primal while others solve the dual, so to compare, we measure *relative error* in the objective value. This is the absolute difference between the objective value and the optimum (measured at the last iteration) over the optimum.

5.1.1 Binary Denoising

We denoise 64×64 images. Each node has a single feature—its noisy observation, and each edge a bias feature for every entry in the overcomplete node potential. This rich parameterization allows both methods to attain zero test error in one iteration. Therefore, we compare the methods' efficiency as optimization algorithms. Fig. 1b shows the speedup of our algorithm; the geometric mean of our speedup for reaching 1% relative error is **283x**. Fig. 1a plots the relative error vs. time of one problem; plots for the others are in Appendix A.1.1. We obtain the desired accuracy in less than one minute while the method of Hazan & Urtasun (2010) requires over an hour.

5.1.2 Image Segmentation

Next we learn to segment the Weizmann horses (Borenstein & Ullman, 2002) using the same setup as Domke (2013) and compare against their algorithm. In Appendix A.1.2 we discuss how the algorithm of Hazan & Urtasun (2010) could not scale to this dataset. In Fig. 2, we compare two variants of our algorithm with three variants of Domke (2013) in terms of objective value and test error over time. MLE-Struct curves are the BCFW version of our Alg. 1. MLE-Struct-wavg evaluates test error using a weighted average of the iterates as described by Lacoste-Julien (2013). The curve for averaged iterates is substantially smoother than the

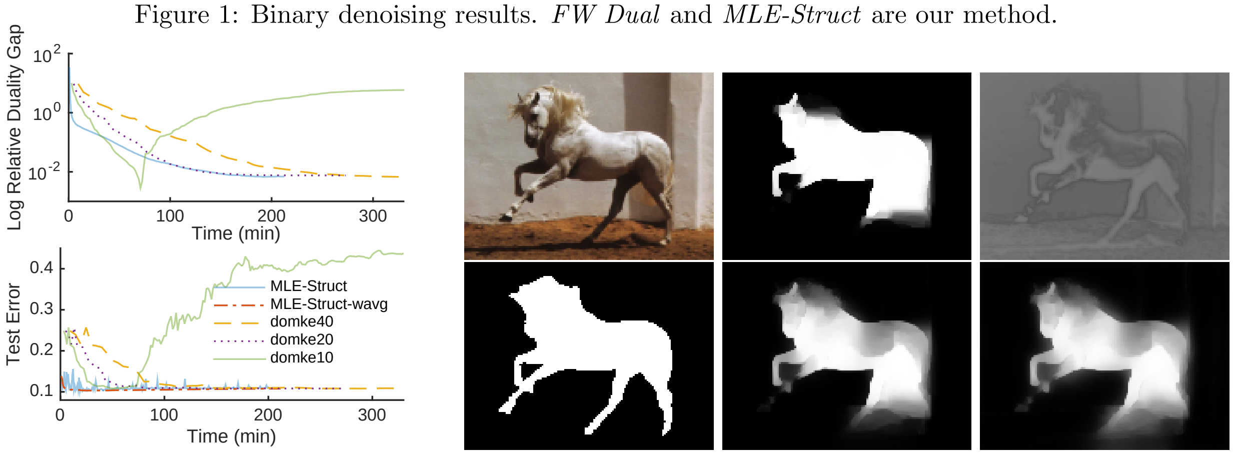


(a) Objective value vs. time.

	Gaussian				Bimodal			
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**
MLE-Struct	0.14	0.15	0.17	0.14	0.20	0.19	0.16	0.19
Hazan & Urtasun (2010)	45.7	45.7	45.1	46.4	46.6	46.2	48.6	48.1
Ratio	331	304	271	340	235	238	308	255

(b) Time (minutes) to reach 1% relative error.

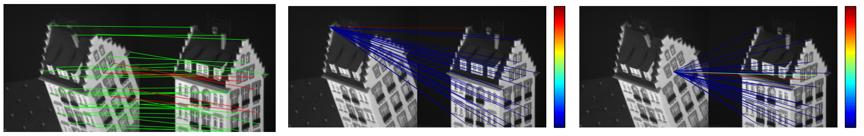
Figure 1: Binary denoising results. *FW Dual* and *MLE-Struct* are our method.



(a) *Top:* Log duality gap vs time. *Bottom:* Test error vs time. The -wavg suffix is for the weighted average iterates.

(b) *Left:* Raw image and ground truth segmentation. *Middle:* MLE-Struct prediction after 9.2 min. (top) and 3.7h (bottom). *Right:* domke40 prediction after 9.2 min. (top) and 3.7h (bottom).

Figure 2: Horse results. *MLE-Struct* curves are our method.

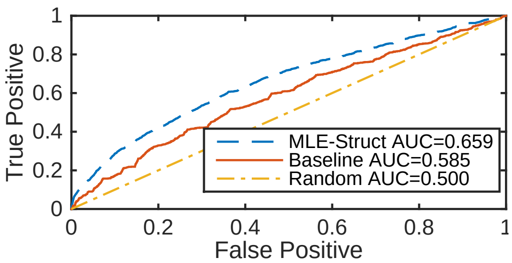


(a) *Left:* Approximate MAP assignment produced from the learned model. Red edges were incorrectly matched. *Mid:* Pseudomarginals for a correctly predicted edge. The correct edge has high probability (red) while all others have low probability (blue). *Right:* Pseudomarginals for a wrongly predicted edge. There are two edges with nontrivial probability (red and green). When the model is forced to pick one, it picked the wrong one.

	Hotel		**House**	
	FW	**lin.+l.**	**FW**	**lin.+l.**
0	0.0	0.0	0	0.0
10	0.0	0.0022	0.0040	0.0
20	0.0049	0.0049	0.0022	0.0022
30	0.020	0.020	0.0049	0.0
40	0.023	0.013	0.0	0.0
50	0.0614	0.050	0.017	0.022
60	0.13	0.12	0.041	0.051
70	0.17	0.15	0.051	0.067
80	0.24	0.19	0.080	0.14
90	0.33	0.30	0.12	0.11

(b) Test Hamming loss for MLE-Struct vs Caetano (2009)

Figure 3: Graph matching results. Left plot and *FW* column are our method.



(a) ROC curves for roommate matching of our algorithm and a constant baseline.

Profile Item	**Weight**
Smoking	-0.0484
Personality	-0.0370
I generally go to bed at...	-0.0296
I generally wake up at...	-0.0218
Study with audio/visual	-0.0133
Overnight Guests	-0.0097
Cleanliness	-0.0056

(b) Largest ℓ_1 distance features of roommate survey data. More negative values increasingly discourage features from differing.

Figure 4: Roommates results. *MLE-Struct* is our method.

raw MLE-Struct curve, and very quickly attains a low test error. The domke$x$ curves result from running their algorithm for x TRW inner loop iterations. Note that this is not guaranteed to converge for any finite x. A practitioner must run the algorithm for a sequence of increasing values of x to confirm convergence to the correct value. Overall, we reach 10% relative error 2.83x and 1% relative error 1.68x faster. We also reach 1% of the final test error 35.1x faster. In Appendix A.1.2 we provide intuition for why we reach low test error so quickly.

The efficiency of MLE-Struct is apparent when visualizing predicted marginals on a test image in Fig. 2b. Domke40 takes 9.2 minutes to complete one iteration. Their marginal estimates at this point only use local intensity data: light regions are classified as "horse" and dark regions are classified as "not horse." In about the same time, our BCFW method had already run 12K iterations and had made 60 passes over the training data. It essentially recovered the correct segmentation (except for difficult portions on the mane and hind legs, where the background texture is confusing) with mean Hamming loss of 0.068.

5.2 High Treewidth Markov Random Fields

We obtained a dataset of 48 financial indicators (such as market indices or macroeconomic indicators) and 19 news indicators (measuring intensity of various categories of events in each part of the world) for all hours in which the indicators were available from the years 2009 to 2014. This is an unsupervised, density estimation problem, since we have no additional features. We converted the data into a binary time series of whether the indicator was below or greater-or-equal-to its median value over the five year period. For simplicity, we ignored the temporal order and treated the data from each hour as an i.i.d. sample. We used the method of Ravikumar et al. (2010) to learn the structure, and then MLE-Struct using indicator features to learn the parameters. Figure 5 shows the model with highest approximate test likelihood of -42.1648, which we selected by holding out 20% of the samples and performing grid search (over one regularization parameter each for the structure and parameter learning procedures). We also computed, via the junction tree algorithm, the exact log-likelihood of -44.3310 for this model, which compares favorably with an independent model with log-likelihood -45.3448.

5.3 Permanents And Matchings

Consider the problem of learning distributions over perfect matchings of a given graph: given an adjacency matrix Y and a weight matrix W, the probability of observing a particular matching is

$$f(Y;W) = \frac{1}{Z(W)} \exp\left(\tfrac{1}{2}\operatorname{tr}(WY)\right). \tag{14}$$

In practice, W is unknown and must be learned from data. We can learn a generative model by estimating W directly, or a conditional model by assuming that W is the linear combination of K feature maps, e.g., $W = \sum_k \theta_k F_k$, and then estimating θ. This formulation can be relaxed to distributions over all matchings by allowing Y to correspond to the adjacency matrix of any (not necessarily perfect) matching.

When G is bipartite, the partition function is the permanent of the matrix of edge weights and is thus #P-hard to compute (Valiant, 1979). Although the partition function can be computed to any given accuracy using a fully polynomial randomized approximation scheme (Jerrum, 2004), such algorithms are impractical for graphs of any significant size. Instead, we will apply our strategy, noting that the reweighted free energy is convex over the local polytope for a wide range of parameter settings. We use max-flow as the MAP solver (Goldberg & Kennedy, 1995; Kolmogorov, 2009). Synthetic experiments in Appendix A.2.1 show that $\rho = 1$ yields very accurate parameter estimates.

Theorem 5.1. *For any $\rho \in [0,1]^{|V|}$, any graph (bipartite or general), and any matching (perfect or imperfect), the reweighted free energy* (3) *is convex over the local polytope.*

Theorem 5.1 is proven in Appendix C. By inclusion, it implies (3) is also convex over the marginal polytope. This generalizes earlier known results on the convexity of the Bethe free energy for bipartite perfect matchings (Vontobel, 2013; Chertkov & Yedidia, 2013) to general matchings; we use this result in our experiments below.

5.3.1 Bipartite Matchings: Stereo Vision

We apply the bipartite matching model to a graph matching problem on the CMU *house* and *hotel* image sequences, comparing against the linear+learning method of Caetano (2009). Both models use the same features and MAP predictors, but theirs minimizes a hinge loss. The methods achieve comparable test error, with ours doing slightly better on the houses and theirs doing slightly better on the hotels. MLE-Struct permits fast and simple approximate MLE in this problem where it was previously difficult. Experimentally, we found that using BP (Huang & Jebara, 2009; Bayati, 2011) to compute marginals for the standard double-loop MLE approach was very unstable.

Fig. 3a illustrates one advantage of learning a probabilistic model versus a discriminative model: the pseudomarginals indicate the model's confidence in a prediction. In many cases, when the algorithm made

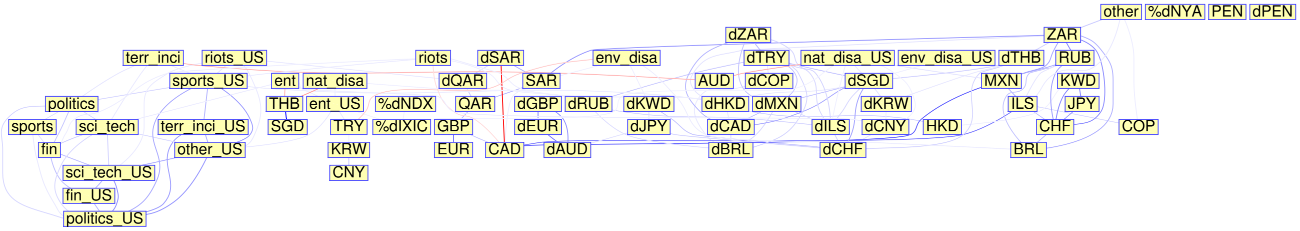


Figure 5: Learned graphical model of binary news and financial indicators. Nodes prefixed with d- encode hourly absolute differences and nodes with %d- encode hourly percentage changes. Blue and red lines denote attractive and repulsive edges, respectively, and darker shades denote larger magnitudes.

the wrong prediction, two edges incident to a specific node had relatively high pseudomarginal probabilities. Here, the errors are not unfounded: similar parts were matched, albeit incorrectly.

5.4 General Matchings

Many undergraduate institutions assign first-year students to roommates based on questionnaires, but allow returning students to pick their own roommates. We can use observed roommate matchings of returning students to train a model for students' preferences. Such a model can then be used to assign first-year students to roommates that they would have picked on their own, had they already met each other. Our data consists of 3 years of roommate assignments and survey responses of 14 questions for 2374–2504 students per year. As our data includes neither class nor gender, we treat the entire matching assignment for one year as one observation. We fit a model of the form (14) for a general matching with MLE-Struct.

MLE-Struct is, to our knowledge, the only MLE-based method to learn a model over general matchings in polynomial time and space. This because while all known descriptions of the general matching polytope require exponentially many constraints ot describe, there exists nevertheless a polynomial-time algorithm to solve MAP problems over the general matching polytope (Kolmogorov, 2009). Since we interact with the polytope only through the MAP solver, we inherit this polynomial runtime. Message-passing algorithms would need to explicitly represent the constraints of the matching polytope, leading to exponential space and runtime requirements.

Table 4b lists the distance features with largest magnitude; more negative values indicate closer agreement is required. Smoking, personality (introverted vs. extroverted), and sleeping habits require the strongest agreement. Additional results are in Appendix A.

As we are effectively classifying $\approx$ 2500 classes using only $\approx$ 14 features, we do not expect high accuracy in terms of Hamming error. Instead, we consider the use-case where we use the model to reject very bad roommate assignments. To evaluate this, we use our learned θ to form the cost matrix from features of the test year (2012), and use the entries of this cost matrix as scores for a binary classifier. We then plot ROC curves in Fig. 4a, where we demonstrate gains above random guesses and a constant baseline. We also evaluated a structured perceptron and structured SVM using the same MAP decoder, but even after extensive parameter tuning, they were unable to generalize, and obtained worse test AUCs than the constant baseline.

6 DISCUSSION

We have introduced a new approximate MLE method that relies only on a simple wrapper around a blackbox MAP solver and scales well to large datasets. The method employs convex free energy approximations with a convex dual learning objective that can be solved efficiently using Frank-Wolfe and block-coordinate Frank-Wolfe optimization. Previously, practitioners either employed expensive double-loop MLE procedures or abandoned MLE by resorting to structured SVMs and perceptrons. Our method is competitive with max-margin MAP-based estimation methods in terms of prediction error and faster than competing MLE methods, while being simple to implement. In the future, we will consider other combinatorial structures, incorporate structure learning with ℓ_1 regularization, and handle latent variable models.

Acknowledgements

This work is supported in part by DARPA grant N66001-15-2-4026 under the GRAPHS program, DARPA grant N66001-15-C-4032 under the SIMPLEX program, NSF grants III-1526914/1527312, IIS-1451500, CCF-1302269. This material is based upon work supported by the NSF Graduate Research Fellowship under grant DGE-11-44155. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation.

References

Bayati, M. et al. Belief propagation for weighted b-matchings on arbitrary graphs and its relation to linear programs with integer solutions. *SIDMA*, 25(2):989–1011, 2011.

Belanger, D. et al. Marginal inference in mrfs using frank-wolfe. In *NIPS Workshop*, 2013.

Borenstein, E. and Ullman, S. Class-specific, top-down segmentation. In *ECCV*, 2002.

Caetano, T. et al. Learning graph matching. *PAMI*, 31(6): 1048–1058, 2009.

Chertkov, M. and Yedidia, A. B. Approximating the permanent with fractional belief propagation. *JMLR*, 14 (1):2029–2066, 2013.

Collins, M. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *EMNLP*, 2002.

Domke, J. Learning graphical model parameters with approximate marginal inference. *PAMI*, 35(10):2454–2467, 2013.

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

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *NRL*, 3(1-2):95–110, 1956.

Ganapathi, V. et al. Constrained approximate maximum entropy learning of Markov random fields. In *UAI*, 2008.

Goldberg, A. and Kennedy, R. An efficient cost scaling algorithm for the assignment problem. *Math. Prog.*, 71 (2):153–177, 1995.

Hazan, T. and Urtasun, R. A primal-dual message-passing algorithm for approximated large scale structured prediction. In *NIPS*. 2010.

Heskes, T. Convexity arguments for efficient minimization of the bethe and kikuchi free energies. *JAIR*, 26:153–190, 2006.

Huang, B. and Jebara, T. Approximating the permanent with belief propagation. *arXiv:0908.1769*, 2009.

Huber, M. and Law, J. Fast approximation of the permanent for very dense problems. In *SODA*, 2008.

Jaggi, M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, 2013.

Jerrum, M. et al. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *JACM*, 51(4):671–697, 2004.

Kolmogorov, V. Blossom v: a new implementation of a minimum cost perfect matching algorithm. *Math. Prog. Comp.*, 1(1):43–67, 2009.

Krishnan, R., Lacoste-Julien, S., and Sontag, D. Barrier frank-wolfe for marginal inference. In *NIPS*, 2015.

Kulesza, A. and Pereira, F. Structured learning with approximate inference. In *NIPS*, 2008.

Kumar, S. and Hebert, M. Discriminative fields for modeling spatial dependencies in natural images. In *NIPS*, 2003.

Lacoste-Julien, S. et al. Block-coordinate Frank-Wolfe optimization for structural SVMs. In *ICML*, 2013.

Lafferty, J. et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

Li, K. et al. Efficient feature learning using perturb-and-map. *NIPS Workshop*, 2013.

Meshi, O. et al. Learning efficiently with approximate inference via dual losses. In *ICML*, 2010.

Petterson, J. et al. Exponential family graph matching and ranking. In *NIPS*, 2009.

Ravikumar, Pradeep, Wainwright, Martin J., and Lafferty, John D. High-dimensional ising model selection using ℓ_1-regularized logistic regression. *Ann. Statist.*, 38(3): 1287–1319, 06 2010.

Rother, C. et al. Optimizing binary MRFs via extended roof duality. In *CVPR*, 2007.

Ruozzi, N. and Tatikonda, S. Message-passing algorithms: Reparameterizations and splittings. *Trans. Inf. Theory*, 59(9):5860–5881, 2013.

Schwing, A. et al. Globally convergent parallel MAP LP relaxation solver using the Frank-Wolfe algorithm. In *ICML*, pp. 487–495, 2014.

Sion, M. On general minimax theorems. *Pac. J. Math.*, 8 (1):171–176, 1958.

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

Taskar, B., Guestrin, C., and Koller, D. Max-margin Markov networks. *NIPS*, 2004.

Tsochantaridis, I. et al. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

Valiant, L. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8(2):189–201, 1979.

Vishwanathan, S. et al. Accelerated training of conditional random fields with stochastic gradient methods. In *ICML*, pp. 969–976, 2006.

Vontobel, P. The Bethe permanent of a nonnegative matrix. *Trans. Inf. Theory*, 59(3):1866–1901, 2013.

Wainwright, M. Estimating the “wrong” graphical model. *JMLR*, 7:1829–1859, 2006.

Wainwright, M. and Jordan, M. Graphical Models, Exponential Families, and Variational Inference. *FnT Machine Learning*, 1(1-2):1–305, 2008.

Weiss, Y. et al. MAP estimation, linear programming and belief propagation with convex free energies. In *UAI*, 2007.