Supplementary Material

The following material is used to provide details about techniques employed in the paper. Part A presents details of experimental setup and additional results. Then in part B we discuss the convergence properties of FW for our problems. In part C we prove convexity of the Bethe free energy for general matchings. Part D derives an instance of our Alg. 1 for matchings, and part E derives the same for linear CRFs.

Then, we present a full derivation of the dual objective and line search objective for doing FW learning for graph matchings. This is presented in terms of matrix-valued terms (various weighted adjacency matrices for the graph), which facilitates easy implementation, since MAP solvers operate on such matrices.

A Details on Experiments

In this appendix, we explain further details of experiments.

A.1 Grid CRFs

A.1.1 Binary Denoising

We used code by Schwing[3] which implements the algorithm in Hazan & Urtasun (2010), and we configured it to solve the same objective as (7). Tuning parameters, such as number of inner loop iterations, message passing schedule, etc. were left at their default values. We set $\lambda = 100$ for both experiments and for all experiments. Test error was measured by Hamming error of the maximum posterior marginal (MPM) estimate, the same metric used in the horses experiments. We did not use linesearch for our algorithm.

Fig. 6 shows the objective value plots for all eight problems. The FW curves are for our algorithm, while the HU curves are the results for the algorithm of Hazan & Urtasun (2010).

Timings were performed on a dedicated 12 core (24 hyper threads) 2.00GHz Intel Xeon E5-2620 machine with 68 GB of physical RAM running Ubuntu 12.04.5 LTS and Matlab R2012b. Processes were restricted to a single thread, and we timed 8 problems simultaneously. Our algorithms were implemented in Matlab, interfacing with combinatorial solvers written in C or C++. Schwing's code was written in C++.

A.1.2 Image Segmentation

We set $\lambda = 3420$ in our parameterization which matches the setting for the published result of Domke (2013). We did not use linesearch.

Timings were performed a dedicated 8 core (16 hyper threads), 2.67GHz Intel Xeon X5550 machine with 24 GB of physical RAM running Ubuntu 12.04.5 LTS and Matlab R2012b. Computations were restricted to a single core, and at most two experiments were run at a time. Our algorithms were implemented in Matlab, interfacing with combinatorial solvers written in C or C++. We downloaded the code for Domke (2013) and Caetano (2009) from the authors' websites, which were implemented in C++ and Matlab with C extensions, respectively. We obtained the original experiment scripts for Domke (2013) through correspondence with the author.

Our denoising model had 4,096 nodes and 40,448 parameters and our horses model had 10,000–40,000 nodes and 368 parameters. Both had about twice as many edges as nodes. Because the horses had many features (between 1.5 to 11.9 million per image), we were unable to train Hazan & Urtasun (2010) on more than 18 images (out of 200) before running out of memory on our 64G machine.

In early iterations our algorithm achieves the lowest objective value and test error. We attain low test error even when the objective value is relatively poor. This phenomenon is a result of the dual formulation: we iteratively move τ to minimize the objective, but for each value of τ, we compute the optimal θ as a linear function of τ. While τ may initially be very inaccurate, contributing to a large objective value, θ is much lower dimensional, so some of the errors may cancel in computing θ, resulting in good predictions nevertheless. In contrast, Domke (2013) iteratively moves θ, and for each value of θ, computes the optimal value of τ using TRW. Prior to convergence, this may enable better fit to the training data at the expense of accurate estimation of θ.

[3] http://alexander-schwing.de/projectsGeneralStructuredPredictionLatentVariables.php

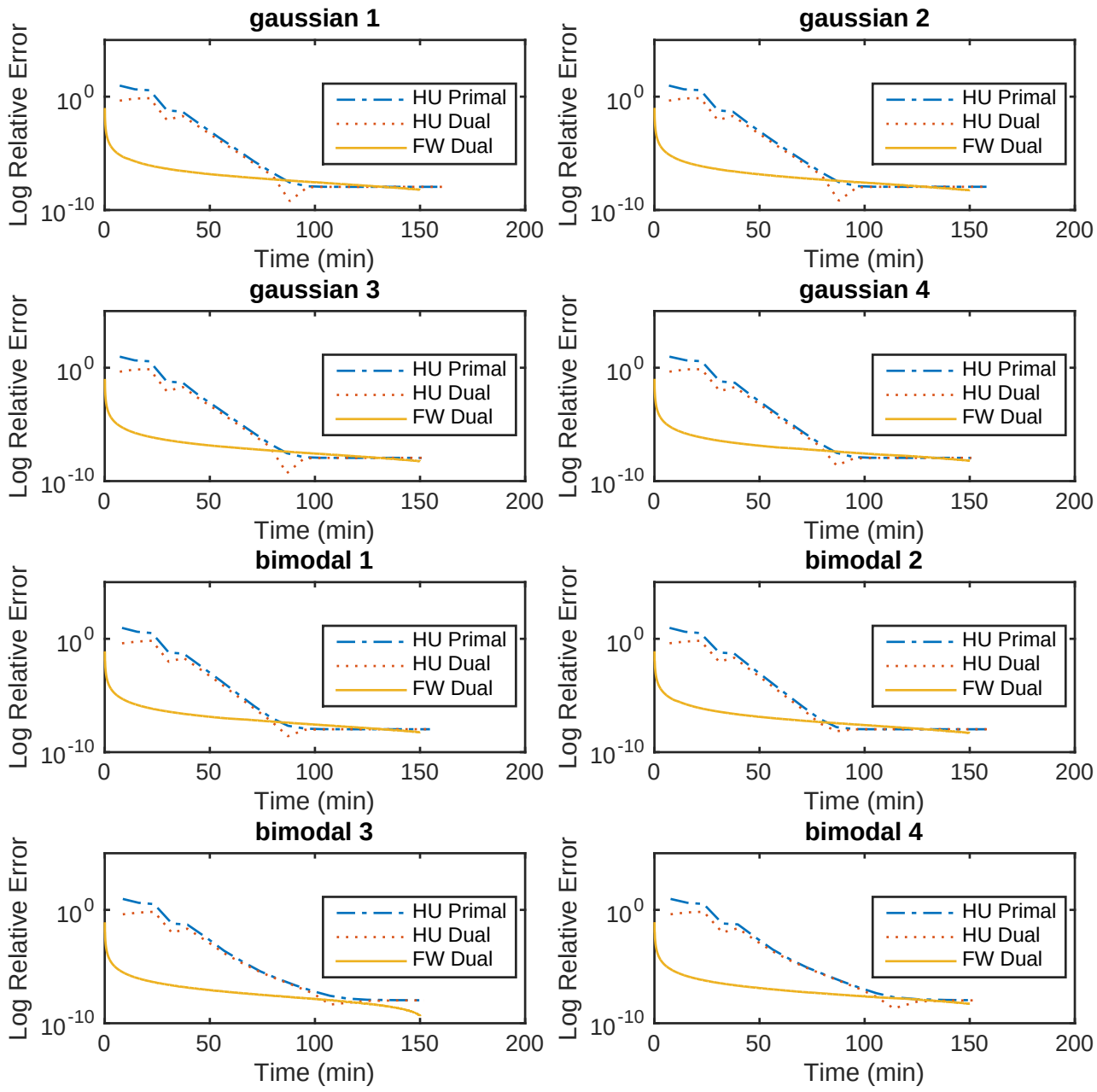


Figure 6: Objective value vs. time for all binary denoising problems.

A.2 Permanents and Matchings

A.2.1 Synthetic Bipartite Matchings

We begin with a synthetic experiment using the flexibility of MLE-Struct to analyzes the accuracy of various entropy approximations for matchings. We sample 10×10 bipartite matchings from the distribution (14). We explore two choices of the weight matrix W: one in the *high SNR regime* with -2 on the off-diagonals and 0 on the diagonals, and one in the *low SNR regime* -0.5 off-diagonal and 0 on-diagonal.

Our problems are small enough that we can compute exact partition functions and their gradients with Ryser's algorithm (Ryser, 1963). Hence, we can perform exact MLE with gradient descent. We can also evaluate the true (regularized) likelihood of our estimates. We ran Alg. 1 with $\rho = 1$ and called this result the *Bethe estimator.* In addition, setting $\rho = 0.5$ for this problem guarantees a concave entropy approximation and an upper bound on the partition function (Meltzer, 2009). We also ran Alg. 1 at this setting and denote the result as the *RW estimator.*

Fig. 7(a) displays the average regularized log-likelihood of each estimator, higher being better and the *Exact* curve being an upper bound. In both low and high SNR regimes, the Bethe estimator is superior to the RW estimator. Reweighted entropies such as the one chosen here are known to perform poorly as estimators of the true partition function as compared to belief propagation. Interestingly, although the objective values of the estimates are different in each case, in the low SNR regime, all estimation methods produce about the same likelihood.

In this experiment, the problems were small enough that we could compare against the true log-likelihood. For larger problem, our framework can be used to bound the value of the true log-likelihood from both above and below, allowing us to assess approximation quality for all problems. We defer the details and experiments to Appendix A.2.1.

Our framework can also be used to bound the value of the true likelihood. First, since Z_{RW} provides an upper bound on Z (Wainwright & Jordan, 2008), we have $Z_{\mathrm{RW}}(W) = Z_{0.5}(W) \geq Z(W)$ for any W. Second, for matchings, we have $Z_B(W) = Z_1(W) \leq Z(W)$ (Gurvits). Therefore, we have

$$\log \ell_{\mathrm{RW}}(W) \leq \log \ell(W) \leq \log \ell_B(W) \tag{15}$$

for all W. Moreover, ℓ_{RW} and ℓ_B are *global* bounds on the maximum likelihood, so the inequalities also hold at their respective optima. That is,

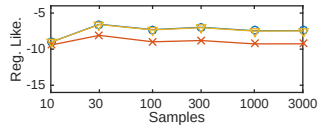
$$\log \ell_{\mathrm{RW}}(W^*_{\mathrm{RW}}) \leq \log \ell(W^*) \leq \log \ell_B(W^*_B) \tag{16}$$

where W^*_{RW} is the RW estimator, W^* is the regularized MLE, and W^*_B is the Bethe estimator. We plot the quantities of (16) in Fig. 7(b). We can also use (15) to obtain upper and lower bounds of $\ell(W^*_B)$ and $\ell(W^*_{\mathrm{RW}})$ by using FW for inference to compute $\ell_B(W^*_{\mathrm{RW}})$ and $\ell_{\mathrm{RW}}(W^*_B)$, since $\ell_B(W^*_B)$ and $\ell_{\mathrm{RW}}(W^*_{\mathrm{RW}})$ will already be available upon convergence of Alg. 1. Fig. 8 shows these results.

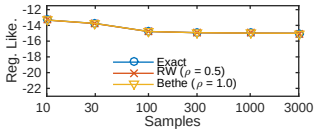
A.2.2 Bipartite Matchings (Stereo Vision)

The data consist of 111 frames of a toy house and 101 separate frames of a toy hotel, each rotated a fixed angle from its predecessor. Each frame was hand-labeled with the same 30 landmark points. We consider pairs of images at a fixed number of frames apart (the *gap*), which we divide into training, validation, and testing sets following the same splits as Caetano (2009). We measure the average Hamming error between the predicted matching (MAP estimate using our learned parameters) and the ground truth.

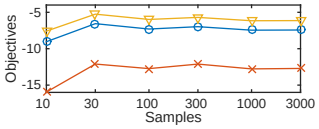
The results of the experiments with reweighting parameters $\rho = \vec{1}$ are described in Fig. 3b. For each subsequence, we chose the regularization via cross-validation. We also compared the performance of our algorithm with different reweighting parameters ρ. Fig. 9 shows the results for the house data when the gap is 50 for various choices of ρ. We observed little difference in test error as ρ varies: this was confirmed over synthetic as well as real data. As a result, we did not tune ρ for different data/problem setups.



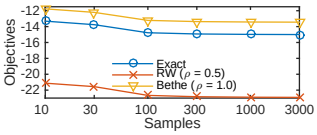
(a) *High SNR problem*: Average (divided by sample size) *true* regularized log-likelihood evaluated at the exact MLE as well as at the parameters that maximize the RW and Bethe approximate log-likelihoods. Higher is better, with the *Exact* curve being the upper bound. The data are nearly as probable under the Bethe estimator as they are under the exact MLE.



(b) *Low SNR problem:* Same plot as (a). All methods perform comparably in this setting.

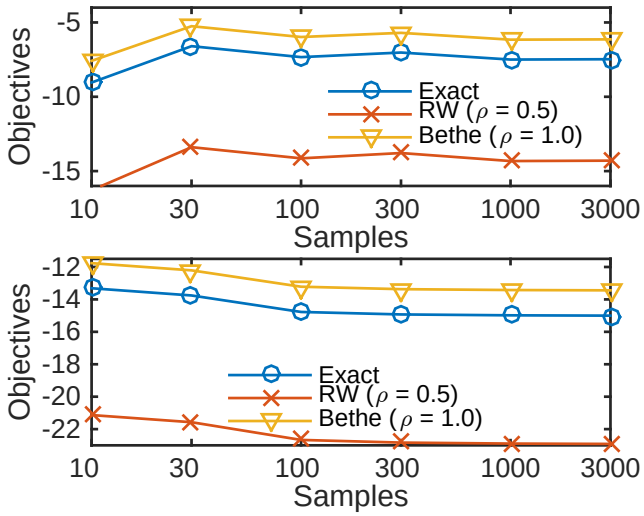


(c) *High SNR problem:* Optimal values of regularized true, and approximate log-likelihoods under the RW and Bethe approximations. The true likelihood is always bounded by the approximations.

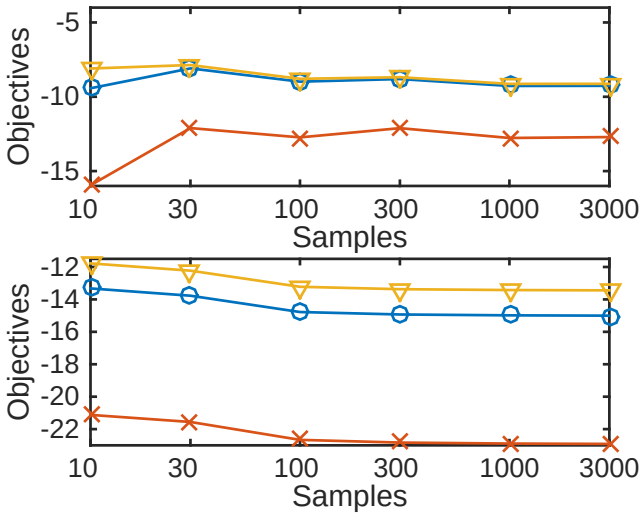


(d) *Low SNR problem:* Same as (c)

Figure 7: Exact likelihoods of the Bethe and RW estimates, and sandwich bounds on the likelihood.



(a) Exact and approximate likelihoods evaluated at the parameters that maximize the Bethe log-likelihood.



(b) Exact and approximate likelihoods evaluated at the parameters that maximize the RW log-likelihood.

Figure 8: Sandwich bounds on the likelihood at the Bethe and TRW estimators. Top is the high SNR problem, bottom is the low SNR problem.

$\boldsymbol{\rho}$	.5	.6	.7	.8	.9	1
Test Loss	0.013	0.013	0.017	0.017	0.020	0.017

Figure 9: Test loss for approximate MLE using FW on the house data set with a gap of 50 and different choices of uniform ρ vectors.

Profile Item Description	Weight	Rank
I generally go to bed at...	-0.0296	3
I generally wake up at...	-0.0218	4
Rising Sophomore	0	10
Cleanliness	-0.0056	7
Smoking	-0.0484	1
Sleeping Habits	-0.0025	8
Overnight Guests	-0.0097	6
Personality	-0.037	2
Usual Study Hours	0.0131	12
Study Location	-0.0006	9
Study with audio/visual	-0.0133	5
Single-sex floor (1)	0.2277	14
Single-sex floor (2)	0.0518	13
Allow in Brownstone	0	10

Figure 10: Distance features for the roommates experiments, their learned weights, and their relative importance ranking with regularization $\lambda = 100$.

A.2.3 General Matchings (Roommate Assignments)

We obtained an anonymized dataset from a major US university over a three year period from 2010-2012. The data consists of roommate assignments for pairs of students in each of the three years. In addition, each of the students was required to complete a brief housing survey that asked for their preferences in terms of cleanliness, sleeping schedule and habits, personality, study preferences, etc. Our questionnaire data consists of 2 binary features and 12 ordinal features of 5 levels each. For each pair of students and each questionnaire question, we created one feature of absolute differences and several interaction indicator features, one for each possible pair of answers to the questionnaire questions. For simplicity, we assumed symmetric interactions. For each student pair, the weighted score for their matching is a linear combination of features. The learned weights for each *distance* features and their relative rankings are described in Fig. 10. As we are using a log-linear model, the weights should be interpreted as log-odds ratio for a unit increase in absolute distance (assuming ordinal features).

A few qualitative observations about these results. First, *single-sex floor*, *rising sophomore*, and *allow in Brownstone* all received relatively low weights, indicating perhaps that the data was too noisy with respect to these survey responses. Second, *personality*, *smoking*, and *bedtime* were among the strongest predictors of a successful match while *cleanliness*, *study hours*, *study location* were among the least important.

The *constant baseline* sets $\theta = -1$ for distance features and $\theta = 0$ for interactions.

When comparing to the BCFW algorithm for a structured SVM, we employed the publicly-available code from the authors. We tried regularization parameter lambda values in the range [10e-4, 10e2]. Our best-performing configuration achieved an AUC of .504 and a hamming error of .9992, which outperforms random guessing, but significantly underperforms a model trained with MLE.

B Convergence of our Frank-Wolfe Learning Algorithm

Recall the following approximate max-entropy objective function introduced in (10):

$$L_\rho^\eta(\tau^{(1:m)}) = \frac{1}{2\lambda}\|\theta^*(\tau^{(1:M)})\|^2 - \sum_m H_\rho^\eta(\tau^{(m)}),$$

In this appendix, we discuss the convergence rate of the FW algorithm for the minimization of the convex function L_ρ^η over the local polytope.

Jaggi (2013) has shown that the error of the FW iterates, for objective L, is bounded by

$$L(\tau_t) - L(\tau^*) \leq \frac{2C_L}{t+2}(1+\delta), \tag{17}$$

where δ is the multiplicative error to which each of the linear subproblems (i.e., the optimization problem performed at each iteration) is solved and C_L is the *curvature* of the function L.

Curvature is a stronger notion of the function's geometry than its Lipschitz parameter, since it is affine-invariant, like the entire Frank-Wolfe algorithm (Jaggi, 2013). The curvature of a differentiable function $F : X \to \mathbb{R}$ is given by

$$C_F = \sup_{\substack{x,x' \in X,\ \gamma \in [0,1] \\ y = x + \gamma(x' - x)}} \frac{2}{\gamma^2} \left(F(y) - F(x) - \langle y - x, \nabla F(x) \rangle \right). \tag{18}$$

The curvature, C_F, quantifies how much F can differ from its linearization. For twice differentiable functions, the curvature can be upper bounded as follows.

$$C_F \leq \sup_{\substack{x,x' \in X,\ \gamma \in [0,1] \\ y = x + \gamma(x' - x)}} \frac{1}{2} (x' - x)^T \nabla^2 F(y)(x' - x) \tag{19}$$

For our objective function, the curvature is most heavily influenced by the entropy term as the curvature of the quadratic piece is simply a constant that depends on the features ϕ.

Unfortunately, the curvature of the entropy terms is unbounded as we approach the boundary of the polytope, and therefore the curvature of the Bethe-MLE objective, i.e., when $\eta = 0$, is unbounded. We instead analyze the case $\eta \in (0, \frac{1}{2})$, which corresponds to forming a quadratic approximation to the entropy terms near the boundary. Then we discuss a heuristic for setting η which guarantees $O(1/t)$ convergence in all cases and incurs no approximation error if all variables are independent. As in Krishnan et al. (2015), who obtain $O(1/t)$ convergence at the expense of introducing approximation error, using $\eta > 0$ also yields an approximation error in general. We expect this error to be very mild compared to that of the Bethe entropy approximation itself.

The modification necessary to make our algorithm have this behavior is trivial and does not change the performance in practice. Most importantly, we can still use the same black-box MAP oracle in the inner loop. All we do is return different gradients for the entropy outside of $[\eta, 1 - \eta]$, which are easy to compute as the function $g_\eta(x)$ is a quadratic on $[0, \eta]$ and $[1 - \eta, 1]$. Furthermore, in practice the modification is unnecessary as our FW iterations never travel outside of $[\eta, 1 - \eta]$.

These techniques can be applied naturally to the test-time inference algorithm described in Section 3.2. In fact, we can obtain tighter bounds because $||\theta||$ is directly available and does not need to be bounded when constructing η.

B.1 Obtaining $O(\frac{1}{t})$ Convergence

We show that the Frank-Wolfe method applied to our (ρ, η)-parameterized objective (10) enjoys $O(\frac{1}{t})$, so long as $\eta \in (0, \frac{1}{2})$. Recall that this objective is not exactly the ρ-weighted free energy: it instead becomes a quadratic approximation for any coordinate within η of the polytope boundary. Our proof begins by bounding the curvature of quadratic, data-dependent, terms. We then bound the curvature of the g_η terms, which are singleton entropy terms that become quadratic approximations near the boundary. We then show that if ρ is chosen to make L^0_ρ convex, then the approximate objective L^η_ρ is still convex. Putting these pieces together yields a proof of convergence.

Definition B.1. *Let R be the maximum possible ℓ_2 norm of all node features $\phi_i(X, Y_i)$ and clique features $\phi_\alpha(X, Y_\alpha)$. Let K be the maximum cardinality of any clique. Let M be the number of training examples. Let N be the maximum of the sum of the number of nodes and cliques in the training examples.*

Lemma B.2. *For the convex approximate MLE dual problem* (10)*, consider the dual-to-primal mappings from feasible iterates τ to corresponding parameters θ_V and $\theta_\mathcal{A}$ of the graphical model given in* (8) *and* (9)*. Then, for all feasible τ, we have that both θ_V and $\theta_\mathcal{A}$ have ℓ_2 norm no greater than $\frac{2}{\lambda} NMR$.*

Proof. In both (8) and (9), θ is $\frac{1}{\lambda}$ times a sum of NM terms, each of which has norm at most $2R$, since each term is a difference of vectors of norm at most R. Therefore, the lemma follows from the triangle inequality. □

Definition B.3 (Restatement of Definition 2.1)**.** *For η in $[0, 1/2)$, define*

$$g_\eta(x) = \begin{cases} x\log(x), & \text{for } x \in [\eta, 1], \\ \eta\log(\eta) + (\log(\eta)+1)(x-\eta) + \frac{(x-\eta)^2}{2\eta}, & \text{for } x \in [0, \eta) \end{cases}$$

g_η approximates $x\log x$ outside of $[\eta, 1]$ by its quadratic extension. Note that g_η is convex, twice continuously differentiable, and $g_\eta(x) \le x\log(x)$ for all $x \in [0, 1]$.

Lemma B.4. *For x such that $\sum_i x_i = 1$, $x_i \ge 0$, the curvature of $g(x) \triangleq \sum_i g_\eta(x_i)$ is less than $2/\eta$.*

Proof. Fix $x, x' \in [0,1]^n$ such that $\sum_i x_i = \sum_i x'_i = 1$. $\nabla^2 g(y)$ is a diagonal matrix whose diagonal elements are each smaller than $1/\eta$ for any $y \in [0,1]^n$. As a result, by (23),

$$C_g \le \frac{||x' - x||^2}{2\eta}.$$

By the triangle inequality,

$$||x' - x||^2 \le (||x'|| + ||x||)^2 \le 4.$$

As a result, $C_g \le \frac{2}{\eta}$. □

Recall (4), the reweighted entropy parameterized by ρ and η:

$$H^\eta_\rho(\tau) = -\sum_{i\in V}\sum_{y_i}\left(1 - \sum_{\alpha\supset i}\rho_\alpha\right) g_\eta(\tau_i(y_i)) - \sum_{\alpha\in\mathcal{A}}\sum_{y_\alpha}\rho_\alpha g_\eta(\tau_\alpha(y_\alpha)).$$

For $\eta = 0$, we can choose ρ to make this function concave (Heskes, 2006; Ruozzi & Tatikonda, 2013). We now show that for $\eta \in (0, \frac{1}{2})$, the entropy approximation is still concave:

Lemma B.5. *For any $\alpha \in \mathcal{A}$ and $i \in \alpha$ with $\eta \in (0, \frac{1}{2})$, the difference between the clique and node entropies*

$$\Delta_{\alpha i} = \sum_{y_\alpha} g_\eta(\tau_\alpha(y_\alpha)) - \sum_{y_i} g_\eta(\tau_i(y_i))$$

is convex under the marginalization constraints.

Proof. The case for $\eta = 0$ is proven in Heskes (2006) Lemma A.1. We extend the argument to consider cases when we use the quadratic approximation instead of $x\log x$.

The local marginalization constraints allow us to identify

$$\tau_i(y_i) = \sum_{y_{\alpha\setminus i}} \tau_\alpha(y_\alpha)$$

where $\alpha\setminus i$ denotes the variables in clique α excluding i. Thus we can write

$$\Delta_{\alpha i} = \sum_{y_\alpha} g_\eta\left(\tau_\alpha(y_\alpha)\right) - \sum_{y_\alpha} g_{\eta_1}\left(\sum_{y_{\alpha\setminus i}} \tau_\alpha(y_\alpha)\right) \tag{20}$$

We interpret the beliefs τ_α as a vector and the labels y_α as indices. Observe that (20) is separable in each coordinate $\tau_\alpha(y_\alpha)$, so its Hessian is diagonal. Specifically, the diagonal entry for y_α on the Hessian is given by

$$\left(\nabla^2\Delta_{\alpha i}\right)_{y_\alpha y_\alpha} = \begin{cases} (\tau_\alpha(y_\alpha))^{-1} & \text{for } \tau_\alpha(y_\alpha) \in [\eta, 1] \\ \eta^{-1} & \text{for } \tau_\alpha(y_\alpha) \in [0, \eta] \end{cases} - \begin{cases} \left(\sum_{y_{\alpha\setminus i}} \tau_\alpha(y_\alpha)\right)^{-1} & \text{for } \sum_{y_{\alpha\setminus i}} \tau_\alpha(y_\alpha) \in [\eta, 1] \\ \eta^{-1} & \text{for } \sum_{y_{\alpha\setminus i}} \tau_\alpha(y_\alpha) \in [0, \eta] \end{cases}$$

We show that $\Delta_{\alpha i}$ is positive semi-definite by showing that the former expression is nonnegative for each of four cases:

(1) If $\tau_\alpha(y_\alpha) \in [\eta, 1]$ and $\sum_{y_{\alpha\setminus i}} \tau_\alpha(y_\alpha) \in [\eta, 1]$, then the $g_\eta(x) = x\log x$ and Lemma A.1 of Heskes (2006) applies.

(2) If $\tau_\alpha(y_\alpha) \in [0, \eta]$ and $\sum_{y_{\alpha\setminus i}} \tau_\alpha(y_\alpha) \in [0, \eta]$, then $\left(\nabla^2 \Delta_{\alpha i}\right)_{y_\alpha y_\alpha} = 0$.

(3) If $\sum_{y_{\alpha\setminus i}} \tau_\alpha(y_\alpha) \in [0, \eta]$, then due to the marginalization constraint, we must have $\tau_\alpha(y_\alpha) \in [0, \eta]$ as well, so this case reduces to case 2.

(4) If $\tau_\alpha(y_\alpha) \in [0, \eta]$ and $\sum_{y_{\alpha\setminus i}} \tau_\alpha(y_\alpha) \in [\eta, 1]$, then we need that $\eta^{-1} \geq \left(\sum_{y_{\alpha\setminus i}} \tau_\alpha(y_\alpha)\right)$, or equivalently, $\eta \leq \sum_{y_{\alpha\setminus i}} \tau_\alpha(y_\alpha)$ which is true by assumption. □

Lemma B.6. *The approximate entropy* (4) *is concave over the local polytope if* $\eta \in (0, \frac{1}{2})$ *and the counting numbers* ρ *satisfy the conditions of Theorem 3.1 in Heskes (2006).*

Proof. The proof for Theorem 3.1 in Heskes (2006) decomposes the entropy (4) at $\eta = 0$ into three terms, each of which is shown to be convex. Two of the terms depend solely on ρ, so these terms in (4) are convex because we satisfy the same conditions. The third term requires $\Delta_{\alpha i}$ to be convex for all $\alpha \in \mathcal{A}$ and $i \in \alpha$, which Lemma B.5 provides. □

Lemma B.7. *There exists a constant* C_ρ*, which depends only on* ρ *and the graph structure of the problem, such that the curvature of the term* $-H_\rho^\eta(\tau)$ *in learning objective* (10) *is at most*

$$\frac{C_\rho(|V| + |\mathcal{A}|)M}{\eta}$$

.

Proof. We can consider each of the approximate entropy terms g_η separately, corresponding to the entropy for a variable or a factor in a single sample. In the Bethe approximation, each variable or factor entropy is preceded by a coefficient that depends only on ρ and the graph structure; let C_ρ be twice the largest absolute value of the coefficients. The curvature of each entropy term is thus bounded by C_ρ/η by Lemma B.4, and there are $(|V| + |\mathcal{A}|)M$ such terms, hence the result. □

Lemma B.8. *In objective function* (10)*, the first term*

$$\frac{\lambda}{2}\|\theta^*(\tau^{(1:M)})\|^2 \tag{21}$$

has curvature constant less than

$$\frac{1}{\lambda}(|V| + |\mathcal{A}|)MR^2. \tag{22}$$

Proof. First, recall that (21) decomposes into terms for the node and clique parameters:

$$\frac{\lambda}{2}\|\theta_V^*(\tau^{(1:M)})\|^2 + \frac{\lambda}{2}\|\theta_{\mathcal{A}}^*(\tau^{(1:M)})\|^2.$$

We first analyze $\frac{\lambda}{2}\|\theta_V^*(\tau^{(1:M)})\|^2$, given in (8), and the curvature of the second term follows by analogy. Since this is a quadratic function of τ, its Hessian H is constant in τ. Define Φ to be a wide matrix where each column contains the the features of for each node in the training set for each possible value that the node can take on. Then, (8) implies

$$H = \frac{1}{2\lambda}\Phi^\top\Phi.$$

Invoking the upper bound on the curvature given in (23), we have that the curvature is bounded above by

$$\sup_{\tau_1,\tau_2\in\mathcal{M}} \frac{1}{2}(\tau_1 - \tau_2)^T H(\tau_1 - \tau_2). \tag{23}$$

$$= \sup_{\tau_1,\tau_2\in\mathcal{M}} \frac{1}{4\lambda}(\tau_1 - \tau_2)^T\Phi^\top\Phi(\tau_1 - \tau_2) \tag{24}$$

$$= \sup_{\tau_1,\tau_2\in\mathcal{M}} \frac{1}{4\lambda}(\Phi(\tau_1 - \tau_2))^\top\Phi(\tau_1 - \tau_2) \tag{25}$$

$$\leq \frac{M|V|}{4\lambda}(2R)(2R) \tag{26}$$

$$= \frac{1}{\lambda}M|V|R^2. \tag{27}$$

Above, (26) follows from the line above because each τ term is a concatenation of marginals for $M|V|$ nodes and the maximum norm of each column of Φ is R by assumption, so the maximum norm of $\Phi(\tau_1 - \tau_2)$ is $2R$. Note that in the lines above, terms such as $\Phi(\tau_1 - \tau_2)$ do not denote feature function evaluation, but matrix-vector multiplication.

The proof follows by also applying the above analysis to $\frac{\lambda}{2}\|\theta^*_{\mathcal{A}}(\tau^{(1:M)})\|^2$ and adding the curvatures of the two terms. □

Theorem B.9 (Restatement of 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. Alg. 1 converges as $O(C/t)$ to the optimum of* (10)*, with curvature*

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

Proof. This follows from summing the bounds in Lemmas B.7 and B.8. □

Observation B.10. *At first glance, these bounds appear quite loose, since they scale with M and N. However, note that we set up our overall learning problem as total training loss plus a regularizer, rather than average training loss plus a regularizer. Therefore, in practice the choice of λ scales with M and N, though it won't scale linearly with these, since for smaller training sets one needs to regularize more heavily. This reduces the effect of these constants in the bounds.*

B.2 Choosing η

We present a technique for choosing a value of $\eta > 0$ by considering a relaxed version of (10). We show that *in this relaxed problem*, we can find a value of η^* such that no coordinate of the optimal belief vector will be η^*-close to the boundary, and therefore we suffer no approximation error.

Proposition B.11. *Consider a simplification of* (10) *where we remove the consistency constraints between beliefs for nodes and edges, yielding a collection of independent logistic regression problems that can be analyzed in closed form. Let τ^* be the optimum of this problem. Then for $\eta^{-1} = 1 + (K-1)\exp(\frac{4}{\lambda}NMR^2)$, we can guarantee that each coordinates τ^* is contained in $[\eta, 1-\eta]$. In this case, the optimum of L^η_ρ and L^0_ρ coincide.*

Proof. Consider a node-wise logistic regression for variable Y with cardinality n. Let $P(Y = j|X) = \frac{\exp(a_j)}{\sum_k \exp(a_k)}$, where $a_1, \ldots, a_n$ are functions of X. Since this is a simple multinomial distribution, its vector of marginals is simply the vector $P = [P(Y = 1|X), P(Y = 2|X), ...P(Y = n|X)]$. We seek η such that every coordinate of P is in $[\eta, 1-\eta]$, and thus it is sufficient to set η to a known lower bound on the minimum entry of P, since if every element of P is greater than η, we have that every element of P must also be less than $1-\eta$, since they sum to one.

Therefore, we construct such a lower bound for the elements of P. Let $A = [a_1, \ldots, a_n]$ and define a^- to be the minimum entry of A and a^+ to be the maximum entry of A. With these, we have that each of the coordinates of P is lower bounded by

$$\eta = \frac{\exp(a^-)}{\exp(a^-) + (n-1)\exp(a^+)}. \tag{28}$$

Namely, the value $Y = j$ receives the smallest possible probability mass if j has un-normalized mass $\exp(a^-)$ and all of the other values have un-normalized mass $\exp(a^+)$.

Therefore, we require bounds on a^- and a^+. Each coordinate of A is obtained by the inner product of θ and ϕ (either θ_V and ϕ_i or $\theta_{\mathcal{A}}$ and ϕ_α). By the Cauchy-Schwartz inequality and Lemma B.2, we have the bound on every element of A: $|a_i| < \frac{2}{\lambda}DMR^2$. In other words, $a^- > -\frac{2}{\lambda}NMR^2$ and $a^+ < \frac{2}{\lambda}NMR^2$. Next we substitute these into (28) and simplify terms. Finally we replace n in (28) with the maximum cardinality of any of these

local logistic regression problems, which is the maximum cardinality of a clique, which we established to be K in Definition B.1. This yields the desired bound.

□

C Convexity of the Bethe Free Energy for General Matchings

In this appendix, we argue that the Bethe free energy for the (not necessarily perfect) matching problem is convex over general graphs. The convexity of the Bethe approximation for the bipartite matching problem was investigated experimentally by Huang & Jebara (2009) and proven by Vontobel (2013). The same argument holds for any choice of reweighting parameters such that $\rho_i \in [0,1]$ for all i. The entropy and polytope approximations are formulated as follows.

$$H'_\rho(\tau) \triangleq \sum_{(i,j)\in E} \Big[(\rho_i+\rho_j-1)(1-\tau_{ij})\log(1-\tau_{ij}) - \tau_{ij}\log\tau_{ij}\Big] - \sum_{i\in V}\rho_i(1-\sum_{j\in\partial i}\tau_{ij})\log(1-\sum_{j\in\partial i}\tau_{ij})$$

where τ is restricted to $\mathcal{T}' = \{\tau \geq 0 : \text{for all } i \in V, \sum + j \in \partial i \tau_{ij} \leq 1\}$.

Theorem C.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.*

Proof. For simplicity we only argue the case $\rho_i = 1$ for all i. Proving the general case simply requires keeping track of additional ρ_i coefficients, as in Theorem 60 of Vontobel (2013).

For the case of the perfect matching problem on a graph G, and for $\rho = 1$, the entropy term of the Bethe free energy can be written as

$$\begin{aligned} H'_{\vec{1}}(\tau) =& \Big[\sum_{(i,j)\in E}(1-\tau_{ij})\log(1-\tau_{ij}) - \tau_{ij}\log\tau_{ij}\Big] - \sum_{i\in V}(1-\sum_{j\in\partial i}\tau_{ij})\log(1-\sum_{j\in\partial i}\tau_{ij}) \\ =& \sum_{i\in V}\Big[-(1-\sum_{j\in\partial i}\tau_{ij})\log(1-\sum_{j\in\partial i}\tau_{ij}) + \frac{1}{2}\sum_{j\in\partial i}\Big((1-\tau_{ij})\log(1-\tau_{ij}) - \tau_{ij}\log\tau_{ij}\Big)\Big] \\ =& \sum_{i\in V}\Big[\frac{1}{2}S(\tau_{i,\partial i}, 1-\sum_{j\in\partial i}\tau_{ij}) + \frac{1}{2}h(1-\sum_{j\in\partial i}\tau_{ij})\Big]. \end{aligned}$$

Here, $h(x) = -x\log x - (1-x)\log(1-x)$ is the entropy function. As proven in Vontobel (2013), the function

$$S_n(x_1,\ldots,x_n) = \sum_{i=1}^{n}(1-x_i)\log(1-x_i) - x_i\log x_i$$

is concave when restricted to the n-dimensional probability simplex for each $n \geq 1$ (above, we employ $n = 2$). Therefore as both S and h are concave functions, the entropy function is concave which implies that the free energy approximation is convex. □

D Frank-Wolfe and Matchings

In this appendix, we describe a conditional random field over perfect matchings, formulate the approximate learning problem in this context, and describe the linesearch procedure used as part of the FW algorithm.

Assume we have M observations, consisting of N items matched to N other items. We represent the m'th observation by $\left(W^{(m)}, X^{(m)}, Y^{(m)}\right)$ where the W and X are $N \times D_W$ and $N \times D_X$ data matrices and $Y^{(m)}$ is an $N \times N$ column permutation matrix [4]. Note that W and X contain the data for the two separate parts of the graph.

[4] That is, if i maps to j, then $Y_{ji} = 0$ and $Y_{ki} = 0$ for $k \neq j$.

In general, conditional random field features can be arbitrary functions of (W, X, Y). To produce a model whose MAP solution is a maximum-weight perfect matching, we require the features to be linear in Y. Since Y_{ji} denotes the presence or absence of edge (i, j), its coefficient ought to depend only on the data for items i and j. Therefore, we use the feature map $F_k(W, X, Y) = \langle G_k(W, X), Y\rangle$ where $(G_k(W, X))_{ij} = g_k(w_i, x_j)$. That is, the k'th feature is a linear function of Y with coefficients given by applying a single function $g_k : \mathbb{R}^{D_X} \times \mathbb{R}^{D_W}$ to every pair of rows in W and X. We will have K features in total. We now write $G_k^{(m)} = G_k\left(W^{(m)}, X^{(m)}\right)$ and dispense with W and X. The probability of one observation is thus

$$p(Y|G_{1:K};\theta) = \frac{1}{Z(\theta)} \exp\left(\sum_k \theta_k \langle G_k, Y\rangle\right)$$

So the log-likelihood for M i.i.d. observations is

$$\ell\left(\theta; Y^{(1:M)}, G_{1:K}^{(1:M)}\right) = \sum_m \sum_k \theta_k \left\langle G_k^{(m)}, Y^{(m)}\right\rangle - \log Z\left(\theta, G_{1:K}^{(m)}\right) \tag{29}$$

We focus on the case $K \le MN^2$.

D.1 Minimax Formulation

We now replace $\log Z$ with $\log Z_{B,\rho}$ and add an L_2 regularizer. Note that ρ is an N-vector reweighting parameter with entries between 0.5 and 1. Using the variational formulation of the (Bethe) free energy, we write the maximum Bethe likelihood problem as a minimax problem, which we further analytically reduce to a convex program with linear constraints. Begin with

$$-\log Z_B\left(\theta, G_{1:K}^{(m)}\right) = \min_{T\in\mathcal{M}} -\sum_k \theta_k \left\langle G_k^{(m)}, T\right\rangle - H_\rho(T). \tag{30}$$

To simplify subsequent derivations, let $y^{(m)} = \mathrm{vec}\left(Y^{(m)}\right)$, $\tau^{(m)} = \mathrm{vec}\left(T^{(m)}\right)$, and $G^{(m)}$ be an $N^2 \times K$ matrix whose k'th column is given by $\mathrm{vec}\left(G_k^{(m)}\right)$. In the sequel, we will use the reweighting parameters in the form of pairwise sums $\rho_i + \rho_j$. Thus, let R be $N \times N$ matrix where $R_{ij} = \rho_i + \rho_j$ and let $r = \mathrm{vec}(R)$. Additionally, define y, τ, and G by vertically stacking all M members of $y^{(m)}, \tau^{(m)}$, and $G^{(m)}$. Thus we can rewrite $\sum_m \theta_k \left\langle G_k^{(m)}, Y^{(m)}\right\rangle = \theta^\top\left(G^\top y\right)$. Plugging (30) into (29) and adding a quadratic penalty gives the problem

$$\begin{aligned}
& \max_\theta \theta^\top\left(G^\top y\right) - \frac{\lambda}{2}\|\theta\|_2^2 + \sum_m \min_{\tau^{(m)}\in\mathcal{M}} -\theta^\top\left(G^{(m)\top}\tau^{(m)}\right) - H_\rho\left(\tau^{(m)}\right) \\
= \;& \max_\theta \theta^\top\left(G^\top y\right) - \frac{\lambda}{2}\|\theta\|_2^2 + \min_{\tau\in\mathcal{M}^M} -\theta^\top\left(G^\top\tau\right) - \sum_m H_\rho\left(\tau^{(m)}\right) \\
= \;& \min_{\tau\in\mathcal{M}^M} \max_\theta \theta^\top\left(G^\top(y-\tau)\right) - \frac{\lambda}{2}\|\theta\|_2^2 - \sum_m H_\rho\left(\tau^{(m)}\right) \\
=: \;& \min_{\tau\in\mathcal{M}^M} \max_\theta f(\tau, \theta)
\end{aligned} \tag{31}$$

The second line is justified because the minimizations in $\tau^{(m)}$ are separable, so the min and sum operators commute. The cost is that we must now minimize over a larger product space $\mathcal{M}^M$, but we will see later why this is not a problem. The last line follows from Sion's minimax theorem: the minimization domain $\mathcal{M}^M$ is compact convex, and the objective is convex in the minimization variable τ and concave in the maximization variable θ (Sion, 1958). The theorem requires only *one* compact domain, so θ can remain unconstrained. Thus, for any τ, the concave function $f(\tau, \cdot)$ attains its maximum at the stationary point $0 = \nabla_\theta f = G^\top(y - \tau) - \lambda\theta$, e.g. $\theta = \lambda^{-1}G^\top(y - \tau)$. Moreover, $f(\tau, \cdot)$ is *strictly* concave for $\lambda > 0$, so the maximum is unique. Plugging in to (31) and simplifying, we get

$$\begin{aligned}
& \min_{\tau\in\mathcal{M}^M} \frac{1}{2\lambda}\left\|G^\top(y-\tau)\right\|_2^2 - \sum_m H_\rho\left(\tau^{(m)}\right) \\
=: \;& \min_{\tau\in\mathcal{M}^M} h(\tau)
\end{aligned} \tag{32}$$

D.2 Line search

To compute the next iterate of FW, τ_{t+1}, we use linesearch. Write $\tau_{t+1} = (1-\tau)\tau_t + \eta\tau^*_{t+1}$. Plugging in to (32), we get

$$\begin{aligned} h_t(\eta) &:= \frac{1}{2\lambda}\left\|G^\top\left(y-(1-\eta)\tau_t-\eta\tau^*_{t+1}\right)\right\|^2 - \sum_m H_{RW}\left((1-\eta)\tau_t^{(m)}+\eta\tau_{t+1}^{*(m)};\rho\right) \\ &= \frac{1}{2\lambda}\left\{\left\|G^\top(y-\tau_t)\right\|^2 + 2\eta\left(y-\tau_t\right)^\top GG^\top\left(\tau_t-\tau^*_{t+1}\right)+\eta^2\left\|G^\top\left(\tau_t-\tau^*_{t+1}\right)\right\|^2\right\} \\ &\quad -\sum_m H_{RW}\left((1-\eta)\tau_t^{(m)}+\eta\tau_{t+1}^{*(m)};\rho\right) \end{aligned} \tag{33}$$

Thus we can precompute the expensive matrix products in the quadratic term.

D.3 General Matchings

The above FW procedure only requires a few changes when switching from complete bipartite graphs to general graphs. The same equations and steps hold when we replace biadjacency feature matrices with adjacency features, and permutation matrices with matrices representing perfect matchings. There are only a few technical caveats. First, for general graphs we need to be able to allow some τ_{ij} to be zero. This can occur because either there is no edge between i and j, or i and j are neighbors, but there is no possible perfect matching in which they are linked. For both of these cases, we simply clamp τ_{ij} at zero. Similarly, some edges occur in every perfect matching, so we need to discover these a-priori and clamp τ_{ij} at one. Second, unlike for bipartite matching, initialization of τ is non-trivial, since the set of neighbors $\mathrm{Nb}(i)$ is different for every i. We cannot choose an integral τ from the local marginal polytope as an initial point, since the curvature is infinite there. Instead, for every edge in the graph, we can find one matching that contains that edge and one matching does not by solving a series of matching problems. We average all of these matchings to obtain an initial feasible point.

E Frank Wolfe and Linear CRFs

E.1 Notation

We work with a conditional random field of L labels over the graph $G = (N, E)$ in the standard overcomplete parameterization. That is, y_n is an $L \times 1$ indicator vector for the state of node v, and y_e is an $L \times L$ indicator matrix for the state of edge e. We will also treat y_e as a vector when convenient. We denote an element of a matrix or vector by parentheses. For node n, let u_n be its $C \times 1$ feature vector and for edge e, let v_e be its $D \times 1$ feature vector. Implicitly, these feature vectors are derived from applying some function to an input vector x. We refer to elements of a vector We will learn a linear map for the node and edge parameters:

$$\begin{aligned} \theta_n &= Fu_n \quad \forall n \in N \\ \theta_e &= Gv_e \quad \forall e \in E \end{aligned}$$

Now suppose we have M exchangeable samples, and let the superscript $\cdot^m$ denote the observation belonging to the mth sample. Our joint log-likelihood is thus

$$\ell\left(F,G;y,u,v\right) = \sum_m \left(\sum_n y_n^{m\top} F u_n + \sum_e y_e^{m\top} G v_e\right) - \log Z\left(F,G,u,v\right)$$

E.2 Minimax Formulation

We replace $\log Z$ with a parameterized surrogate likelihood $\log Z_\rho$ which interpolates between the TRW and Bethe approximations. We use the variational formulation of $\log Z_\rho$, over the *local* polytope $\mathcal{T}$. Note that the Bethe approximation is not convex in this setting, but TRW is. For grid MRFs, each edge has probability 0.5 of appearing in a spanning tree, so we set $\rho = 0.5$ for each edge.

Since we are estimating matrix parameters, we add a Frobenius penalty. The minimax formulation is thus

$$\begin{aligned}
&\max_{F,G} \sum_m \left(\sum_n y_n^{m\top} F u_n^m + \sum_e y_e^{m\top} G v_e^m \right) - \frac{\lambda}{2} \|F\|_F^2 - \frac{\lambda}{2} \|G\|_F^2 \\
&+ \sum_m \min_{\mu^m \in \mathcal{T}} - \left(\sum_n \mu_n^{m\top} F u_n^m + \sum_e \mu_e^{m\top} G v_e^m \right) - H_\rho(\mu) \\
= \quad & \min_{\mu \in \mathcal{T}^M} \max_{F,G} \sum_m \left(\sum_n \left(y_n^m - \mu_n^m\right)^\top F u_n^m + \sum_e \left(y_e^m - \mu_e^m\right)^\top G v_e^m \right) \\
&- \frac{\lambda}{2} \|F\|_F^2 - \frac{\lambda}{2} \|G\|_F^2 - \sum_m H_\rho\left(\mu^{(m)}\right)
\end{aligned} \tag{34}$$

where the reweighted approximate entropy is given by

$$\begin{aligned}
H_\rho(\mu) &:= \sum_{n \in N} H(\mu_n) - \sum_{nn' \in E} \rho_{nn'} I(\mu_{nn'}) \\
&= \sum_{n \in N} H(\mu_n) - \sum_{n \in N} \sum_{n' \in \mathrm{Ne}(n)} \rho_{nn'} \left[H(\mu_n) + H(\mu_{n'})\right] + \sum_{nn' \in E} \rho_{nn'} H(\mu_{nn'}) \\
&= \sum_{n \in N} \left(1 - \sum_{n' \in \mathrm{Ne}(n)} \rho_{nn'}\right) H(\mu_n) + \sum_{nn' \in N} \rho_{nn'} H(\mu_{nn'})
\end{aligned} \tag{35}$$

where $I(\mu_{nn'}) = \sum_{y_n y_{n'}} \mu_{nn'}(y_n, y_{n'}) \log\left[\mu_{nn'}(y_n, y_{n'}) / \mu_n(y_n) \mu_{n'}(y_{n'})\right]$ is the mutual information between variables n and n' and $H(\mu_n) = -\sum_{y_n} \mu_n(y_n) \log \mu_n(y_n)$ and $H(\mu_{nn'}) = -\sum_{y_n y_{n'}} \mu_{nn'}(y_n, y_{n'}) \log \mu_{nn'}(y_n, y_{n'})$ are singleton and pairwise entropies. We have used the identity $I(\mu_{nn'}) = H(\mu_n) + H(\mu_{n'}) - H(\mu_{nn'})$. We have implicitly used the *pairwise* marginalization constraints when using the mutual information identity, so these gradients are valid only on the local polytope—a fact that is important to remember when optimizing.

The stationary point of the objective in (34) is thus

$$\begin{aligned}
0 &= \sum_{mn} \left(y_n^m - \mu_n^m\right) u_n^{m\top} - \lambda F \\
\Rightarrow F &= \lambda^{-1} \sum_{mn} \left(y_n^m - \mu_n^m\right) u_v^{m\top}
\end{aligned} \tag{36}$$

Similarly, $G = \lambda^{-1} \sum_{me} \left(y_e^m - \mu_e^m\right) v_e^\top$. Recalling the definition of the Frobenius norm and rearranging some summations, we get

$$\frac{\lambda}{2} \|F\|_F^2 = \frac{\lambda}{2\lambda^2} \left\| \sum_{mn} \left(y_n^m - \mu_n^m\right) u_n^{m\top} \right\|_F^2 = \frac{1}{2\lambda} \sum_{mn} \sum_{m'n'} \left(y_n^m - \mu_n^m\right)^\top \left(y_{n'}^{m'} - \mu_{n'}^{m'}\right) u_{n'}^{m'\top} u_n^m \tag{37}$$

On the other hand, the quadratic terms are

$$\begin{aligned}
\frac{1}{\lambda} \sum_{mn} \left(y_n^m - \mu_n^m\right)^\top \sum_{m'n'} \left(y_{n'}^{m'} - \mu_{n'}^{m'}\right) u_{n'}^{m'\top} u_n^m &= \frac{1}{\lambda} \sum_{mn} \sum_{m'n'} \left(y_n^m - \mu_n^m\right)^\top \left(y_{n'}^{m'} - \mu_{n'}^{m'}\right) u_{n'}^{m'\top} u_n^m \\
&= 2\lambda \|F\|_F^2
\end{aligned} \tag{38}$$

e.g. the same Frobenius norm of outer products, by comparison with (37).

We have eliminated the matrix F (and similarly, G), which reveals that our objective is a quadratic form in the Gram matrices $UU^\top$ and $VV^\top$, where U is obtained by vertically stacking the u_n^m and V obtained by vertically stacking the v_e^m, so that the entry $\left(UU^\top\right)_{nm,n'm'} = u_n^{m\top} u_{n'}^{m'}$ and $\left(VV^\top\right)_{em,e'm'} = v_e^{m'\top} v_{e'}^{m'}$. Let Y_N, T_N be the matrices obtained whose (mn, ℓ)'th entry is given by $u_n^m(\ell), \mu_n^m(\ell)$ and V_E, T_E be the matrices whose $(me, \ell\ell')$'th entries are given by $v_n^m(\ell, \ell'), \mu_n^m(\ell, \ell')$. The objective is quadratic in $(Y_E - T_E)$ and $(Y_N - T_N)$, so we can flip them to simplify some signs later. Write $W = T - Y$. Then the objective is

$$\min_{T_N, T_E \in T^{\mathcal{M}}} = \frac{1}{2\lambda} \left\langle W_N W_N^\top, UU^\top \right\rangle + \frac{1}{2\lambda} \left\langle W_E W_E^\top, VV^\top \right\rangle - H_\rho\left(T_N, T_E\right)$$

$$\begin{aligned} & \frac{1}{2\lambda}\operatorname{tr}\left(W_N^\top U U^\top W_N\right) + \frac{1}{2\lambda}\operatorname{tr}\left(W_E^\top V V^\top W_E\right) - H_\rho\left(T_N, T_E\right) \\ = \quad & \frac{1}{2\lambda}\left\|U^\top W_N\right\|_F^2 + \frac{1}{2\lambda}\left\|V^\top W_E\right\|_F^2 - H_\rho\left(T_N, T_E\right) \end{aligned} \tag{39}$$

with a matricized form of the entropy as

$$H_\rho(T_N, T_E) \quad = \quad -\left\langle 1 - R_N, T_N \circ \log T_N \right\rangle - \left\langle R_E, T_E \circ \log T_E \right\rangle$$

where $\circ$ denotes elementwise multiplication, and R_N, R_E are matrices of reweighting parameters conforming to T_N, T_E. That is, $(R_N)_{nm,:} = \sum_{n' \in \mathrm{Ne}(n)} \rho_{nn'}$ while $(R_E)_{ne,:} = \rho_{nn'}$. From this form, the gradients are evidently

$$\begin{aligned} \frac{\partial H_\rho}{\partial T_N} \quad &= \quad -\left(1 - R_N\right) \circ \left(1 + \log T_N\right) \\ \frac{\partial H_\rho}{\partial T_N} \quad &= \quad -R_E \circ \left(1 + \log T_E\right) \end{aligned}$$

So the gradients of our objective are

$$\frac{\partial h}{\partial T_N} \quad = \quad \lambda^{-1} U U^\top \left(T_N - Y_N\right) + \left(1 - R_N\right) \circ \left(1 + \log T_N\right) \tag{40}$$

$$\frac{\partial h}{\partial T_E} \quad = \quad \lambda^{-1} V V^\top \left(T_E - Y_E\right) + R_E \circ \left(1 + \log T_E\right) \tag{41}$$

E.2.1 Linesearch

The quadratic terms of $h(\mu + \eta d)$, as a function of η, are

$$\begin{aligned} & \frac{1}{2\lambda}\left(\left\|U^\top\left(W_N + \eta D_N\right)\right\|_F^2 + \left\|V^\top\left(W_E + \eta D_E\right)\right\|_F^2\right) \\ = \quad & \frac{1}{2\lambda}\left(\left\|U^\top W_N + \eta U^\top D_N\right\|_F^2 + \left\|V^\top W_E + \eta V^\top D_E\right\|_F^2\right) \\ = \quad & \frac{1}{2\lambda}\left(\left\|U^\top W_N\right\|_F^2 + \left\|V^\top W_E\right\|_F^2 + \eta\left(\left\langle U^\top W_N, U^\top D_N\right\rangle + \left\langle V^\top W_E, V^\top D_E\right\rangle\right) + \eta^2\left(\left\|U^\top D_N\right\|_F^2 + \left\|V^\top D_E\right\|_F^2\right)\right) \end{aligned}$$

The inner products $\langle \cdot, \cdot \rangle$ are matrix inner products of $C \times L$ and $D \times L^2$ matrices. For the Bethe/TRW entropy, we will just have to treat it as a black box.

E.2.2 Block-Coordinate Updates

For block-coordinate Frank-Wolfe, we can update the gradient and perform linesearch without computing the full inner products $U^\top W_N$ and $V^\top W_E$. Let η denote a step size, $D^m = (D_N^m, D_E^m)$ denote the step direction for sample m, and U^m, V^m be the submatrices containing only those rows for sample m. More precisely, $D^m = (D_N^m, D_E^m)$ where D_N^m has the same number of rows as U and D_E^m has the same number of rows as V, but only those rows for sample m are nonzero. Suppose we move to the point $T + \eta D^m = (T_N + \eta D_N^m, T_E + \eta D_E^m)$. Then our new inner products are

$$\begin{aligned} U^\top\left(T_N + \eta D_N^{[m]} - Y_N\right) \quad &= \quad U^\top W_N + \eta U^{m\top} D^m \\ V^\top\left(\mu_E + \eta d^{[m]} - y_E\right) \quad &= \quad V^\top W_E + \eta V^{m\top} d^m \end{aligned}$$

Thus we need only add the update terms $\eta U^{m\top} D_N^m$ and $\eta V^{m\top} D_E^m$. The algorithm depends on the value of T only through these inner products.

F Experiments for FW-based Inference

In Fig. 11, we compare Frank-Wolfe (FW) for bipartite perfect matching to the perturb-and-MAP algorithm of Li (2013), using code obtained from the authors. In (a) and (b) we plot the ℓ_∞ distance of the approximate

l_∞ error	.05	.01	0.005	0.001
rand-20	3.37	1.56	1.12	0 .25
rand-50	18.0	11.76	5.18	0.91
rand-75	23.87	23.87	9.8	1.5
rand-100	19.24	19.24	10.6	1.75
lda-20	1.6	.61	.34	.10

Table 1: Frank-Wolfe speedup over BP for various error tolerances.

marginals from exact marginals computed with brute force v.s. the number of calls to the maximum-matching solver. We use a bipartite graph with 10 nodes on each side, i.i.d. edge weights distributed unif$[0, 1]$, and inverse temperatures of 10 (a) and 0.25 (b). In (c), we run each algorithm for a very large number of MAP calls for a range of temperatures in order to identify the affect of temperature on the algorithms' errors, and results are aggregated over 10 random $n = 10$ graphs. Overall, we find that the Bethe approximation provided by FW is substantially more accurate than perturb-and-MAP and that FW converges more quickly. However, (c) suggests that changes in temperature affect the algorithms' approximation accuracies differently.

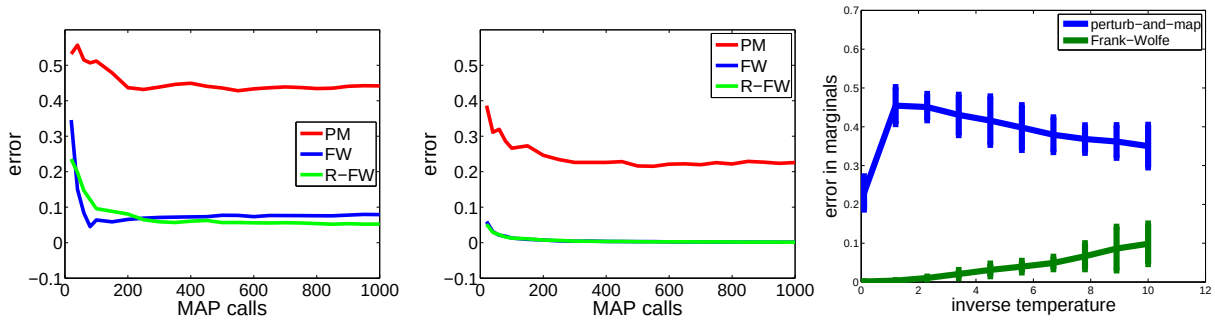


Figure 11: Frank-Wolfe v.s. Perturb-and-MAP

Our proposed FW algorithm for bipartite perfect matching and the belief propagation algorithm of Huang & Jebara (2009) minimize the same objective over the same polytope, so we focus on the speed-accuracy trade-offs of the algorithms. In Table 1, 'rand-n' refers to random complete bipartite graphs with n nodes on each side and i.i.d. edge weights from unif$[0, 1]$. The 'lda-20' experiment aligns topics from different runs of a Gibbs sampler for an LDA topic model with 20 topics. All results are averages over 10 graphs. We run the algorithms with a range of termination tolerances in order to obtain various speed-accuracy points. Then, for a range of l_∞ distances to the true Bethe marginals, we compute the time necessary to achieve the specified error. The table presents the ratio of computation time for BP to FW (ratio > 1 means FW is faster). As expected from an algorithm that is no faster than $O(\frac{1}{t})$, we find that it gives good accuracy quickly, but is slow to converge to within very tight error tolerances. Such a speed-accuracy trade off affects other first order methods such as SGD, but can still be advantageous in many cases, including large-scale applications or when optimizing beyond the statistical error of the problem is pointless.

Supplementary References

Gurvits, L. Unleashing the power of Schrijver's permanental inequality with the help of the Bethe approximation. *arXiv:1106.2844*.

Meltzer, T. et al. Convergent message passing algorithms: a unifying view. In *UAI*, 2009.

Ryser, H. J. *Combinatorial mathematics*. Carus mathematical monographs. MAA, 1963.