Majorization for CRFs and Latent Likelihoods (Supplementary Material)

Tony Jebara
Department of Computer Science
Columbia University
jebara@cs.columbia.edu

Anna Choromanska
Department of Electrical Engineering
Columbia University
aec2163@columbia.edu

Abstract

This supplement presents additional details in support of the full article. These include the application of the majorization method to maximum entropy problems. It also contains proofs of the various theorems, in particular, a guarantee that the bound majorizes the partition function. In addition, a proof is provided guaranteeing convergence on (non-latent) maximum conditional likelihood problems. The supplement also contains supporting lemmas that show the bound remains applicable in constrained optimization problems. The supplement then proves the soundness of the junction tree implementation of the bound for graphical models with large n. It also proves the soundness of the low-rank implementation of the bound for problems with large d. Finally, the supplement contains additional experiments and figures to provide further empirical support for the majorization methodology.

Supplement for Section 2

Proof of Theorem 1 Rewrite the partition function as a sum over the integer index $j = 1, \ldots, n$ under the random ordering $\pi : \Omega \mapsto \{1, \ldots, n\}$. This defines $j = \pi(y)$ and associates h and $\mathbf{f}$ with $h_j = h(\pi^{-1}(j))$ and $\mathbf{f}_j = \mathbf{f}(\pi^{-1}(j))$. Next, write $Z(\boldsymbol{\theta}) = \sum_{j=1}^{n} \alpha_j \exp(\boldsymbol{\lambda}^\top \mathbf{f}_j)$ by introducing $\boldsymbol{\lambda} = \boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}$ and $\alpha_j = h_j \exp(\tilde{\boldsymbol{\theta}}^\top \mathbf{f}_j)$. Define the partition function over only the first i components as $Z_i(\boldsymbol{\theta}) = \sum_{j=1}^{i} \alpha_j \exp(\boldsymbol{\lambda}^\top \mathbf{f}_j)$. When $i = 0$, a trivial quadratic upper bound holds

$$Z_0(\boldsymbol{\theta}) \quad \leq \quad z_0 \exp\left(\tfrac{1}{2}\boldsymbol{\lambda}^\top \boldsymbol{\Sigma}_0 \boldsymbol{\lambda} + \boldsymbol{\lambda}^\top \boldsymbol{\mu}_0\right)$$

with the parameters $z_0 \to 0^+, \boldsymbol{\mu}_0 = \mathbf{0}$, and $\boldsymbol{\Sigma}_0 = z_0 \mathbf{I}$. Next, add one term to the current partition function $Z_1(\boldsymbol{\theta}) = Z_0(\boldsymbol{\theta}) + \alpha_1 \exp(\boldsymbol{\lambda}^\top \mathbf{f}_1)$. Apply the current bound $Z_0(\boldsymbol{\theta})$ to obtain

$$Z_1(\boldsymbol{\theta}) \;\leq\; z_0 \exp(\tfrac{1}{2}\boldsymbol{\lambda}^\top \boldsymbol{\Sigma}_0 \boldsymbol{\lambda} + \boldsymbol{\lambda}^\top \boldsymbol{\mu}_0) + \alpha_1 \exp(\boldsymbol{\lambda}^\top \mathbf{f}_1).$$

Consider the following change of variables

$$\begin{aligned} \mathbf{u} &= \boldsymbol{\Sigma}_0^{1/2} \boldsymbol{\lambda} - \boldsymbol{\Sigma}_0^{-1/2} (\mathbf{f}_1 - \boldsymbol{\mu}_0)) \\ \gamma &= \tfrac{\alpha_1}{z_0} \exp(\tfrac{1}{2}(\mathbf{f}_1 - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_0^{-1} (\mathbf{f}_1 - \boldsymbol{\mu}_0)) \end{aligned}$$

and rewrite the logarithm of the bound as

$$\log Z_1(\boldsymbol{\theta}) \quad \leq \quad \log z_0 - \tfrac{1}{2}(\mathbf{f}_1 - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_0^{-1} (\mathbf{f}_1 - \boldsymbol{\mu}_0) + \boldsymbol{\lambda}^\top \mathbf{f}_1 + \log\left(\exp(\tfrac{1}{2}\|\mathbf{u}\|^2) + \gamma\right).$$

Apply Lemma 1 (cf. [32] p. 100) to the last term to get

$$\begin{aligned} \log Z_1(\boldsymbol{\theta}) \leq \log z_0 - \tfrac{1}{2}(\mathbf{f}_1 - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_0^{-1} (\mathbf{f}_1 - \boldsymbol{\mu}_0) + \boldsymbol{\lambda}^\top \mathbf{f}_1 + \log\left(\exp\left(\tfrac{1}{2}\|\mathbf{v}\|^2\right) + \gamma\right) \\ + \frac{\mathbf{v}^\top (\mathbf{u} - \mathbf{v})}{1 + \gamma \exp(-\frac{1}{2}\|\mathbf{v}\|^2)} + \frac{1}{2}(\mathbf{u} - \mathbf{v})^\top \left(I + \Gamma \mathbf{v}\mathbf{v}^\top\right)(\mathbf{u} - \mathbf{v}) \end{aligned}$$

where $\Gamma = \frac{\tanh(\frac{1}{2}\log(\gamma\exp(-\frac{1}{2}\|\mathbf{v}\|^2)))}{2\log(\gamma\exp(-\frac{1}{2}\|\mathbf{v}\|^2))}$. The bound in [32] is tight when $\mathbf{u} = \mathbf{v}$. To achieve tightness when $\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}$ or, equivalently, $\boldsymbol{\lambda} = \mathbf{0}$, we choose $\mathbf{v} = \boldsymbol{\Sigma}_0^{-1/2}(\boldsymbol{\mu}_0 - \mathbf{f}_1)$ yielding

$$Z_1(\boldsymbol{\theta}) \quad \leq \quad z_1 \exp\left(\tfrac{1}{2}\boldsymbol{\lambda}^\top\boldsymbol{\Sigma}_1\boldsymbol{\lambda} + \boldsymbol{\lambda}^\top\boldsymbol{\mu}_1\right)$$

where we have

$$\begin{aligned} z_1 &= z_0 + \alpha_1 \\ \boldsymbol{\mu}_1 &= \frac{z_0}{z_0+\alpha_1}\boldsymbol{\mu}_0 + \frac{\alpha_1}{z_0+\alpha_1}\mathbf{f}_1 \\ \boldsymbol{\Sigma}_1 &= \boldsymbol{\Sigma}_0 + \frac{\tanh(\frac{1}{2}\log(\alpha_1/z_0))}{2\log(\alpha_1/z_0)}(\boldsymbol{\mu}_0 - \mathbf{f}_1)(\boldsymbol{\mu}_0 - \mathbf{f}_1)^\top. \end{aligned}$$

This rule updates the bound parameters $z_0, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0$ to incorporate an extra term in the sum over i in $Z(\boldsymbol{\theta})$. The process is iterated n times (replacing 1 with i and 0 with $i-1$) to produce an overall bound on all terms.

Lemma 1 *(See [32] p. 100)*
For all $\mathbf{u} \in \mathbb{R}^d$*, any* $\mathbf{v} \in \mathbb{R}^d$ *and any* $\gamma \geq 0$*, the bound* $\log\left(\exp\left(\frac{1}{2}\|\mathbf{u}\|^2\right) + \gamma\right) \leq$

$$\log\left(\exp\left(\tfrac{1}{2}\|\mathbf{v}\|^2\right) + \gamma\right) + \frac{\mathbf{v}^\top(\mathbf{u}-\mathbf{v})}{1+\gamma\exp(-\frac{1}{2}\|\mathbf{v}\|^2)} + \frac{1}{2}(\mathbf{u}-\mathbf{v})^\top\left(I + \Gamma\mathbf{v}\mathbf{v}^\top\right)(\mathbf{u}-\mathbf{v})$$

holds when the scalar term $\Gamma = \frac{\tanh(\frac{1}{2}\log(\gamma\exp(-\|\mathbf{v}\|^2/2)))}{2\log(\gamma\exp(-\|\mathbf{v}\|^2/2))}$*. Equality is achieved when* $\mathbf{u} = \mathbf{v}$.

Proof of Lemma 1 The proof is provided in [32].

Supplement for Section 3

Maximum entropy problem We show here that partition functions arise naturally in maximum entropy estimation or minimum relative entropy $\mathcal{RE}(p\|h) = \sum_y p(y)\log\frac{p(y)}{h(y)}$ estimation. Consider the following problem:

$$\min_p \mathcal{RE}(p\|h) \;\text{ s.t.} \sum_y p(y)\mathbf{f}(y) = \mathbf{0}, \sum_y p(y)\mathbf{g}(y) \geq \mathbf{0}.$$

Here, assume that $\mathbf{f} : \Omega \mapsto \mathbb{R}^d$ and $\mathbf{g} : \Omega \mapsto \mathbb{R}^{d'}$ are arbitrary (non-constant) vector-valued functions over the sample space. The solution distribution $p(y) = h(y)\exp\left(\boldsymbol{\theta}^\top\mathbf{f}(y) + \boldsymbol{\vartheta}^\top\mathbf{g}(y)\right)/Z(\boldsymbol{\theta},\boldsymbol{\vartheta})$ is recovered by the dual optimization

$$\boldsymbol{\theta},\boldsymbol{\vartheta} = \underset{\boldsymbol{\vartheta}\geq\mathbf{0},\boldsymbol{\theta}}{\arg\max} -\log\sum_y h(y)\exp\left(\boldsymbol{\theta}^\top\mathbf{f}(y) + \boldsymbol{\vartheta}^\top\mathbf{g}(y)\right)$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$ and $\boldsymbol{\vartheta} \in \mathbb{R}^{d'}$. These are obtained by minimizing $Z(\boldsymbol{\theta},\boldsymbol{\vartheta})$ or equivalently by maximizing its negative logarithm. Algorithm 1 permits variational maximization of the dual via the quadratic program

$$\min_{\boldsymbol{\vartheta}\geq\mathbf{0},\boldsymbol{\theta}} \tfrac{1}{2}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})^\top\boldsymbol{\Sigma}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}) + \boldsymbol{\beta}^\top\boldsymbol{\mu}$$

where $\boldsymbol{\beta}^\top = [\boldsymbol{\theta}^\top\boldsymbol{\vartheta}^\top]$. Note that any general convex hull of constraints $\boldsymbol{\beta} \in \boldsymbol{\Lambda} \subseteq \mathbb{R}^{d+d'}$ could be imposed without loss of generality.

Proof of Theorem 2 We begin by proving a lemma that will be useful later.

Lemma 2 *If* $\kappa\boldsymbol{\Psi} \succeq \boldsymbol{\Phi} \succ \mathbf{0}$ *for* $\boldsymbol{\Phi}, \boldsymbol{\Psi} \in \mathbb{R}^{d\times d}$*, then*

$$\begin{aligned} L(\boldsymbol{\theta}) &= -\tfrac{1}{2}(\boldsymbol{\theta}-\tilde{\boldsymbol{\theta}})^\top\boldsymbol{\Phi}(\boldsymbol{\theta}-\tilde{\boldsymbol{\theta}}) - (\boldsymbol{\theta}-\tilde{\boldsymbol{\theta}})^\top\boldsymbol{\mu} \\ U(\boldsymbol{\theta}) &= -\tfrac{1}{2}(\boldsymbol{\theta}-\tilde{\boldsymbol{\theta}})^\top\boldsymbol{\Psi}(\boldsymbol{\theta}-\tilde{\boldsymbol{\theta}}) - (\boldsymbol{\theta}-\tilde{\boldsymbol{\theta}})^\top\boldsymbol{\mu} \end{aligned}$$

satisfy $\sup_{\boldsymbol{\theta}\in\boldsymbol{\Lambda}} L(\boldsymbol{\theta}) \geq \frac{1}{\kappa}\sup_{\boldsymbol{\theta}\in\boldsymbol{\Lambda}} U(\boldsymbol{\theta})$ *for any convex* $\boldsymbol{\Lambda} \subseteq \mathbb{R}^d$*,* $\tilde{\boldsymbol{\theta}} \in \boldsymbol{\Lambda}$*,* $\boldsymbol{\mu} \in \mathbb{R}^d$ *and* $\kappa \in \mathbb{R}^+$.

Proof of Lemma 2 Define the primal problems of interest as $\mathbf{P}_L = \sup_{\boldsymbol{\theta}\in\boldsymbol{\Lambda}} L(\boldsymbol{\theta})$ and $\mathbf{P}_U = \sup_{\boldsymbol{\theta}\in\boldsymbol{\Lambda}} U(\boldsymbol{\theta})$. The constraints $\boldsymbol{\theta} \in \boldsymbol{\Lambda}$ can be summarized by a set of linear inequalities $\mathbf{A}\boldsymbol{\theta} \leq \mathbf{b}$ where $\mathbf{A} \in \mathbb{R}^{k\times d}$ and $\mathbf{b} \in \mathbb{R}^k$ for some (possibly infinite) $k \in \mathbb{Z}$. Apply the change of variables $\mathbf{z} = \boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}$. The constraint $\mathbf{A}(\mathbf{z}+\tilde{\boldsymbol{\theta}}) \leq \mathbf{b}$ simplifies into $\mathbf{A}\mathbf{z} \leq \tilde{\mathbf{b}}$ where $\tilde{\mathbf{b}} = \mathbf{b} - \mathbf{A}\tilde{\boldsymbol{\theta}}$. Since $\tilde{\boldsymbol{\theta}} \in \boldsymbol{\Lambda}$, it is easy to show that $\tilde{\mathbf{b}} \geq \mathbf{0}$. We obtain the equivalent primal problems $\mathbf{P}_L = \sup_{\mathbf{A}\mathbf{z}\leq\tilde{\mathbf{b}}} -\frac{1}{2}\mathbf{z}^\top\boldsymbol{\Phi}\mathbf{z} - \mathbf{z}^\top\boldsymbol{\mu}$ and $\mathbf{P}_U = \sup_{\mathbf{A}\mathbf{z}\leq\tilde{\mathbf{b}}} -\frac{1}{2}\mathbf{z}^\top\boldsymbol{\Psi}\mathbf{z} - \mathbf{z}^\top\boldsymbol{\mu}$. The corresponding dual problems are

$$\mathbf{D}_L = \inf_{\mathbf{y}\geq\mathbf{0}} \frac{\mathbf{y}^\top\mathbf{A}\boldsymbol{\Phi}^{-1}\mathbf{A}^\top\mathbf{y}}{2} + \mathbf{y}^\top\mathbf{A}\boldsymbol{\Phi}^{-1}\boldsymbol{\mu} + \mathbf{y}^\top\tilde{\mathbf{b}} + \frac{\boldsymbol{\mu}^\top\boldsymbol{\Phi}^{-1}\boldsymbol{\mu}}{2}$$
$$\mathbf{D}_U = \inf_{\mathbf{y}\geq\mathbf{0}} \frac{\mathbf{y}^\top\mathbf{A}\boldsymbol{\Psi}^{-1}\mathbf{A}^\top\mathbf{y}}{2} + \mathbf{y}^\top\mathbf{A}\boldsymbol{\Psi}^{-1}\boldsymbol{\mu} + \mathbf{y}^\top\tilde{\mathbf{b}} + \frac{\boldsymbol{\mu}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\mu}}{2}.$$

Due to strong duality, $\mathbf{P}_L = \mathbf{D}_L$ and $\mathbf{P}_U = \mathbf{D}_U$. Apply the inequalities $\boldsymbol{\Phi} \preceq \kappa\boldsymbol{\Psi}$ and $\mathbf{y}^\top\tilde{\mathbf{b}} > 0$ as

$$\begin{aligned}\mathbf{P}_L &\geq \sup_{\mathbf{A}\mathbf{z}\leq\tilde{\mathbf{b}}} -\frac{\kappa}{2}\mathbf{z}^\top\boldsymbol{\Psi}\mathbf{z} - \mathbf{z}^\top\boldsymbol{\mu} = \inf_{\mathbf{y}\geq\mathbf{0}} \frac{\mathbf{y}^\top\mathbf{A}\boldsymbol{\Psi}^{-1}\mathbf{A}^\top\mathbf{y}}{2\kappa} + \frac{\mathbf{y}^\top\mathbf{A}\boldsymbol{\Psi}^{-1}\boldsymbol{\mu}}{\kappa} + \mathbf{y}^\top\tilde{\mathbf{b}} + \frac{\boldsymbol{\mu}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\mu}}{2\kappa} \\ &\geq \frac{1}{\kappa}\mathbf{D}_U = \frac{1}{\kappa}\mathbf{P}_U.\end{aligned}$$

This proves that $\mathbf{P}_L \geq \frac{1}{\kappa}\mathbf{P}_U$.

We will use the above to prove Theorem 2. First, we will upper-bound (in the Loewner ordering sense) the matrices $\boldsymbol{\Sigma}_j$ in Algorithm 2. Since $\|\mathbf{f}_{x_j}(y)\|^2 \leq r$ for all $y \in \Omega_j$ and since $\boldsymbol{\mu}_j$ in Algorithm 1 is a convex combination of $\mathbf{f}_{x_j}(y)$, the outer-product terms in the update for $\boldsymbol{\Sigma}_j$ satisfy

$$(\mathbf{f}_{x_j}(y) - \boldsymbol{\mu})(\mathbf{f}_{x_j}(y) - \boldsymbol{\mu})^\top \quad \preceq \quad 4r^2\mathbf{I}.$$

Thus, $\boldsymbol{\Sigma}_j \preceq \mathcal{F}(\alpha_1, \ldots, \alpha_n)4r^2\mathbf{I}$ holds where

$$\mathcal{F}(\alpha_1, \ldots, \alpha_n) \quad = \quad \sum_{i=2}^{n} \frac{\tanh(\frac{1}{2}\log(\frac{\alpha_i}{\sum_{k=1}^{i-1}\alpha_k}))}{2\log(\frac{\alpha_i}{\sum_{k=1}^{i-1}\alpha_k})}$$

using the definition of $\alpha_1, \ldots, \alpha_n$ in the proof of Theorem 1. The formula for $\mathcal{F}$ starts at $i = 2$ since $z_0 \to 0^+$. Assume permutation π is sampled uniformly at random. The expected value of $\mathcal{F}$ is then

$$\mathrm{E}_\pi[\mathcal{F}(\alpha_1, \ldots, \alpha_n)] = \frac{1}{n!}\sum_{\pi}\sum_{i=2}^{n} \frac{\tanh(\frac{1}{2}\log(\frac{\alpha_{\pi(i)}}{\sum_{k=1}^{i-1}\alpha_{\pi(k)}}))}{2\log(\frac{\alpha_{\pi(i)}}{\sum_{k=1}^{i-1}\alpha_{\pi(k)}})}.$$

We claim that the expectation is maximized when all $\alpha_i = 1$ or any positive constant. Also, $\mathcal{F}$ is invariant under uniform scaling of its arguments. Write the expected value of $\mathcal{F}$ as E for short. Next, consider $\frac{\partial \mathrm{E}}{\partial \alpha_l}$ at the setting $\alpha_i = 1, \forall i$. Due to the expectation over π, we have $\frac{\partial \mathrm{E}}{\partial \alpha_l} = \frac{\partial \mathrm{E}}{\partial \alpha_o}$ for any l, o. Therefore, the gradient vector is constant when all $\alpha_i = 1$. Since $\mathcal{F}(\alpha_1, \ldots, \alpha_n)$ is invariant to scaling, the gradient vector must therefore be the all zeros vector. Thus, the point when all $\alpha_i = 1$ is an extremum or a saddle. Next, consider $\frac{\partial}{\partial \alpha_o}\frac{\partial \mathrm{E}}{\partial \alpha_l}$ for any l, o. At the setting $\alpha_i = 1$, $\frac{\partial^2 \mathrm{E}}{\partial \alpha_l^2} = -c(n)$ and, $\frac{\partial}{\partial \alpha_o}\frac{\partial \mathrm{E}}{\partial \alpha_l} = c(n)/(n-1)$ for some non-negative constant function $c(n)$. Thus, the $\alpha_i = 1$ extremum is locally concave and is a maximum. This establishes that $\mathrm{E}_\pi[\mathcal{F}(\alpha_1, \ldots, \alpha_n)] \leq \mathrm{E}_\pi[\mathcal{F}(1, \ldots, 1)]$ and yields the Loewner bound

$$\boldsymbol{\Sigma}_j \quad \preceq \quad \left(2r^2\sum_{i=1}^{n-1}\frac{\tanh(\log(i)/2)}{\log(i)}\right)\mathbf{I} \;=\; \omega\mathbf{I}.$$

Apply this bound to each $\boldsymbol{\Sigma}_j$ in the lower bound on $J(\boldsymbol{\theta})$ and also note a corresponding upper bound

$$J(\boldsymbol{\theta}) \geq J(\tilde{\boldsymbol{\theta}}) - \tfrac{t\omega+t\lambda}{2}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|^2 - \sum_j (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top(\boldsymbol{\mu}_j - \mathbf{f}_{x_j}(y_j))$$
$$J(\boldsymbol{\theta}) \leq J(\tilde{\boldsymbol{\theta}}) - \tfrac{t\lambda}{2}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|^2 - \sum_j (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top(\boldsymbol{\mu}_j - \mathbf{f}_{x_j}(y_j))$$

which follows from Jensen's inequality. Define the current $\tilde{\boldsymbol{\theta}}$ at time τ as $\boldsymbol{\theta}_\tau$ and denote by $L_\tau(\boldsymbol{\theta})$ the above lower bound and by $U_\tau(\boldsymbol{\theta})$ the above upper bound at time τ. Clearly, $L_\tau(\boldsymbol{\theta}) \leq J(\boldsymbol{\theta}) \leq U_\tau(\boldsymbol{\theta})$ with equality when $\boldsymbol{\theta} = \boldsymbol{\theta}_\tau$. Algorithm 2 maximizes $J(\boldsymbol{\theta})$ after initializing at $\boldsymbol{\theta}_0$ and performing an update by maximizing a lower bound based on $\boldsymbol{\Sigma}_j$. Since $L_\tau(\boldsymbol{\theta})$ replaces the definition of $\boldsymbol{\Sigma}_j$ with $\omega\mathbf{I} \succeq \boldsymbol{\Sigma}_j$, $L_\tau(\boldsymbol{\theta})$ is a looser bound than the one used by Algorithm 2. Thus, performing $\boldsymbol{\theta}_{\tau+1} = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Lambda}} L_\tau(\boldsymbol{\theta})$ makes less progress than a step of Algorithm 1. Consider computing the slower update at each iteration τ and returning $\boldsymbol{\theta}_{\tau+1} = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Lambda}} L_\tau(\boldsymbol{\theta})$. Setting $\boldsymbol{\Phi} = (t\omega + t\lambda)\mathbf{I}$, $\boldsymbol{\Psi} = t\lambda\mathbf{I}$ and $\kappa = \frac{\omega+\lambda}{\lambda}$ allows us to apply Lemma 2 as follows

$$\sup_{\boldsymbol{\theta}\in\boldsymbol{\Lambda}} L_\tau(\boldsymbol{\theta}) - L_\tau(\boldsymbol{\theta}_\tau) \quad = \quad \frac{1}{\kappa} \sup_{\boldsymbol{\theta}\in\boldsymbol{\Lambda}} U_\tau(\boldsymbol{\theta}) - U_\tau(\boldsymbol{\theta}_\tau).$$

Since $L_\tau(\boldsymbol{\theta}_\tau) = J(\boldsymbol{\theta}_\tau) = U_\tau(\boldsymbol{\theta}_\tau)$, $J(\boldsymbol{\theta}_{\tau+1}) \geq \sup_{\boldsymbol{\theta}\in\boldsymbol{\Lambda}} L_\tau(\boldsymbol{\theta})$ and $\sup_{\boldsymbol{\theta}\in\boldsymbol{\Lambda}} U_\tau(\boldsymbol{\theta}) \geq J(\boldsymbol{\theta}^*)$, we obtain

$$J(\boldsymbol{\theta}_{\tau+1}) - J(\boldsymbol{\theta}^*) \quad \geq \quad \left(1 - \frac{1}{\kappa}\right)\left(J(\boldsymbol{\theta}_\tau) - J(\boldsymbol{\theta}^*)\right).$$

Iterate the above inequality starting at $t = 0$ to obtain

$$J(\boldsymbol{\theta}_\tau) - J(\boldsymbol{\theta}^*) \quad \geq \quad \left(1 - \frac{1}{\kappa}\right)^\tau \left(J(\boldsymbol{\theta}_0) - J(\boldsymbol{\theta}^*)\right).$$

A solution within a multiplicative factor of ϵ implies that $\epsilon = \left(1 - \frac{1}{\kappa}\right)^\tau$ or $\log(1/\epsilon) = \tau \log \frac{\kappa}{\kappa-1}$. Inserting the definition for κ shows that the number of iterations τ is at most $\left\lceil \frac{\log(1/\epsilon)}{\log \frac{\kappa}{\kappa-1}} \right\rceil$ or $\left\lceil \frac{\log(1/\epsilon)}{\log(1+\lambda/\omega)} \right\rceil$. Inserting the definition for ω gives the bound.

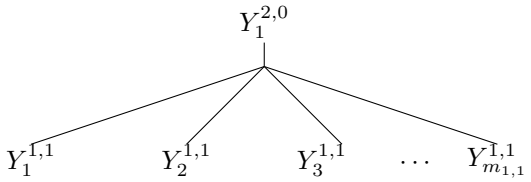

Figure 3: Junction tree of depth 2.

Algorithm 5 SmallJunctionTree

Input Parameters $\tilde{\boldsymbol{\theta}}$ and $h(u), f(u)\ \forall u \in Y_1^{2,0}$ and $z_i, \boldsymbol{\Sigma}_i, \boldsymbol{\mu}_i\ \forall i = 1, \ldots, m_{1,1}$

Initialize $z \to 0^+, \boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = z\mathbf{I}$

For each configuration $u \in Y_1^{2,0}$ {

$\alpha \quad = h(u)(\prod_{i=1}^{m_{1,1}} z_i \exp(-\tilde{\boldsymbol{\theta}}^\top \boldsymbol{\mu}_i)) \exp(\tilde{\boldsymbol{\theta}}^\top (f(u) + \sum_{i=1}^{m_{1,1}} \boldsymbol{\mu}_i)) = h(u) \exp(\tilde{\boldsymbol{\theta}}^\top f(u)) \prod_{i=1}^{m_{1,1}} z_i$

$\mathbf{l} \quad = f(u) + \sum_{i=1}^{m_{1,1}} \boldsymbol{\mu}_i - \boldsymbol{\mu}$

$\boldsymbol{\Sigma} += \sum_{i=1}^{m_{1,1}} \boldsymbol{\Sigma}_i + \frac{\tanh(\frac{1}{2}\log(\alpha/z))}{2\log(\alpha/z)} \mathbf{l}\mathbf{l}^\top$

$\boldsymbol{\mu} += \frac{\alpha}{z+\alpha}\mathbf{l}$

$z \ += \alpha \quad \}$

Output $z, \boldsymbol{\mu}, \boldsymbol{\Sigma}$

Supplement for Section 5

Proof of correctness for Algorithm 3 Consider a simple junction tree of depth 2 shown on Figure 3. The notation $Y_c^{a,b}$ refers to the c^{th} tree node located at tree level a (first level is considered as the one with tree leaves) whose parent is the b^{th} from the higher tree level (the root has no parent so $b = 0$). Let $\sum_{Y_{c_1}^{a_1,b_1}}$ refer to the sum over all configurations of variables in $Y_{c_1}^{a_1,b_1}$ and $\sum_{Y_{c_1}^{a_1,b_1} \setminus Y_{c_2}^{a_2,b_2}}$ refers to the sum over all configurations of variables that are in $Y_{c_1}^{a_1,b_1}$ but not in $Y_{c_2}^{a_2,b_2}$. Let $m_{a,b}$ denote the number of children of the b^{th} node located at tree level $a+1$. For short-hand, use $\psi(Y) = h(Y)\exp(\boldsymbol{\theta}^\top f(Y))$. The partition function can be expressed as:

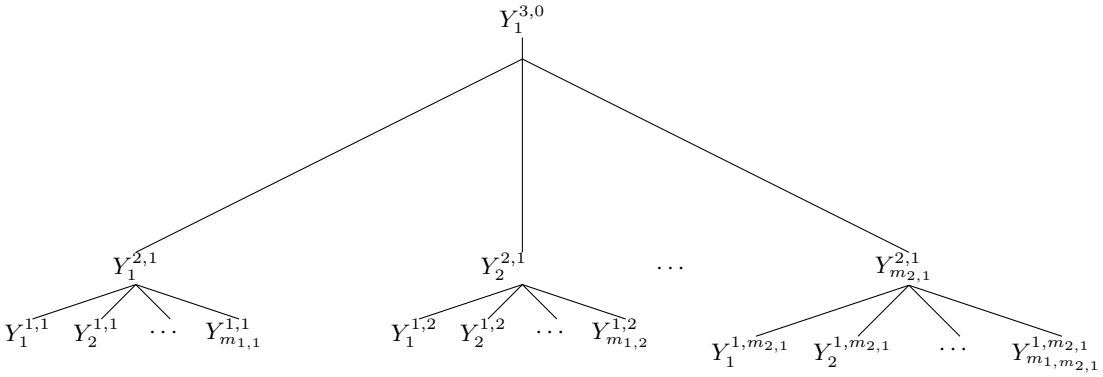

Figure 4: Junction tree of depth 3.

$$
\begin{aligned}
Z(\boldsymbol{\theta}) &= \sum_{u \in Y_1^{2,0}} \left[\psi(u) \prod_{i=1}^{m_{1,1}} \left(\sum_{v \in Y_i^{1,1} \setminus Y_1^{2,0}} \psi(v) \right) \right] \\
&\leq \sum_{u \in Y_1^{2,0}} \left[\psi(u) \prod_{i=1}^{m_{1,1}} z_i \exp\left(\frac{1}{2} \left(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}} \right)^\top \boldsymbol{\Sigma}_i (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top \boldsymbol{\mu}_i \right) \right] \\
&= \sum_{u \in Y_1^{2,0}} \left[h(u) \exp(\boldsymbol{\theta}^\top f(u)) \prod_{i=1}^{m_{1,1}} z_i \exp\left(\frac{1}{2} (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top \boldsymbol{\Sigma}_i (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top \boldsymbol{\mu}_i \right) \right]
\end{aligned}
$$

where the upper-bound is obtained by applying Theorem 1 to each of the terms $\sum_{v \in Y_i^{1,1} \setminus Y_1^{2,0}} \psi(v)$. By simply rearranging terms we get:

$$
\begin{aligned}
Z(\boldsymbol{\theta}) \leq \sum_{u \in Y_1^{2,0}} \Bigg[h(u) \left(\prod_{i=1}^{m_{1,1}} z_i \exp(-\tilde{\boldsymbol{\theta}}^\top \boldsymbol{\mu}_i) \right) \exp\left(\boldsymbol{\theta}^\top \left(f(u) + \sum_{i=1}^{m_{1,1}} \boldsymbol{\mu}_i \right) \right) \\
\exp\left(\frac{1}{2} (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top \left(\sum_{i=1}^{m_{1,1}} \boldsymbol{\Sigma}_i \right) (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}) \right) \Bigg].
\end{aligned}
$$

One can prove that this expression can be upper-bounded by $z \exp\left(\frac{1}{2} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \boldsymbol{\Sigma} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \boldsymbol{\mu} \right)$ where z, $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ can be computed using Algorithm 5 (a simplification of Algorithm 3). We will call this result Lemma A. The proof is similar to the proof of Theorem 1 so is not repeated here.

Consider enlarging the tree to a depth 3 as shown on Figure 4. The partition function is now

$$
Z(\boldsymbol{\theta}) = \sum_{u \in Y_1^{3,0}} \left[\psi(u) \prod_{i=1}^{m_{2,1}} \left(\sum_{v \in Y_i^{2,1} \setminus Y_1^{3,0}} \left(\psi(v) \prod_{j=1}^{m_{1,i}} \left(\sum_{w \in Y_j^{1,i} \setminus Y_i^{2,1}} \psi(w) \right) \right) \right) \right].
$$

By Lemma A we can upper bound each $\sum_{v \in Y_i^{2,1} \setminus Y_1^{3,0}} \left(\psi(v) \prod_{j=1}^{m_{1,i}} \left(\sum_{w \in Y_j^{1,i} \setminus Y_i^{2,1}} \psi(w) \right) \right)$ term by the expression $z_i \exp\left(\frac{1}{2} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \boldsymbol{\Sigma}_i (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \boldsymbol{\mu}_i \right)$. This yields

$$
Z(\boldsymbol{\theta}) \leq \sum_{u \in Y_1^{3,0}} \left[\psi(u) \prod_{i=1}^{m_{2,1}} z_i \exp\left(\frac{1}{2} (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top \boldsymbol{\Sigma}_i (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top \boldsymbol{\mu}_i \right) \right].
$$

This process can be viewed as collapsing the sub-trees $S_1^{2,1}$, $S_2^{2,1}$, ..., $S_{m_{2,1}}^{2,1}$ to super-nodes that are represented by bound parameters, z_i, $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\mu}_i, i = \{1, 2, \cdots, m_{2,1}\}$, where the sub-trees are

defined as:

$$\begin{aligned} S_1^{2,1} &= \{Y_1^{2,1}, Y_1^{1,1}, Y_2^{1,1}, Y_3^{1,1}, \ldots, Y_{m_{1,1}}^{1,1}\} \\ S_2^{2,1} &= \{Y_2^{2,1}, Y_1^{1,2}, Y_2^{1,2}, Y_3^{1,2}, \ldots, Y_{m_{1,2}}^{1,2}\} \\ &\vdots \\ S_{m_{2,1}}^{2,1} &= \{Y_{m_{2,1}}^{2,1}, Y_1^{1,m_{2,1}}, Y_2^{1,m_{2,1}}, Y_3^{1,m_{2,1}}, \ldots, Y_{m_{1,m_{2,1}}}^{1,m_{2,1}}\}. \end{aligned}$$

Notice that the obtained expression can be further upper bounded using again Lemma A (induction) yielding a bound of the form: $z \exp\left(\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \boldsymbol{\Sigma}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \boldsymbol{\mu}\right)$.

Finally, for a general tree, follow the same steps described above, starting from leaves and collapsing nodes to super-nodes, each represented by bound parameters. This procedure effectively yields Algorithm 3 for the junction tree under consideration.

Supplement for Section 6

Proof of correctness for Algorithm 4 We begin by proving a lemma that will be useful later.

Lemma 3 *For all* $\mathbf{x} \in \mathbb{R}^d$ *and for all* $\mathbf{l} \in \mathbb{R}^d$,

$$\sum_{i=1}^{d} \mathbf{x}(i)^2 \mathbf{l}(i)^2 \geq \left(\sum_{i=1}^{d} \mathbf{x}(i) \frac{\mathbf{l}(i)^2}{\sqrt{\sum_{j=1}^{d} \mathbf{l}(j)^2}} \right)^2 .$$

Proof of Lemma 3 By Jensen's inequality,

$$\sum_{i=1}^{d} \mathbf{x}(i)^2 \frac{\mathbf{l}(i)^2}{\sum_{j=1}^{d} \mathbf{l}(j)^2} \geq \left(\sum_{i=1}^{d} \frac{\mathbf{x}(i)\mathbf{l}(i)^2}{\sum_{j=1}^{d} \mathbf{l}(j)^2} \right)^2 \Longleftrightarrow \sum_{i=1}^{d} \mathbf{x}(i)^2 l(i)^2 \geq \left(\sum_{i=1}^{d} \frac{\mathbf{x}(i)\mathbf{l}(i)^2}{\sqrt{\sum_{j=1}^{d} \mathbf{l}(j)^2}} \right)^2 .$$

Now we prove the correctness of Algorithm 4. At the i^{th} iteration, the algorithm stores $\boldsymbol{\Sigma}_i$ using a low-rank representation $\mathbf{V}_i^\top \mathbf{S}_i \mathbf{V}_i + \mathbf{D}_i$ where $\mathbf{V}_i \in \mathbb{R}^{k \times d}$ is orthonormal, $\mathbf{S}_i \in \mathbb{R}^{k \times k}$ positive semi-definite and $\mathbf{D}_i \in \mathbb{R}^{d \times d}$ is non-negative diagonal. The diagonal terms $\mathbf{D}$ are initialized to $t\lambda\mathbf{I}$ where λ is the regularization term. To mimic Algorithm 1 we must increment the $\boldsymbol{\Sigma}$ matrix by a rank one update of the form $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_{i-1} + \mathbf{r}_i \mathbf{r}_i^\top$. By projecting $\mathbf{r}_i$ onto each eigenvector in $\mathbf{V}$, we can decompose it as $\mathbf{r}_i = \sum_{j=1}^{k} \mathbf{V}_{i-1}(j, \cdot)\mathbf{r}_i \mathbf{V}_{i-1}(j, \cdot)^\top + \mathbf{g} = \mathbf{V}_{i-1}^\top \mathbf{V}_{i-1} \mathbf{r}_i + \mathbf{g}$ where $\mathbf{g}$ is the remaining residue. Thus the update rule can be rewritten as:

$$\begin{aligned} \boldsymbol{\Sigma}_i &= \boldsymbol{\Sigma}_{i-1} + \mathbf{r}_i \mathbf{r}_i^\top = \mathbf{V}_{i-1}^\top \mathbf{S}_{i-1} \mathbf{V}_{i-1} + \mathbf{D}_{i-1} + (\mathbf{V}_{i-1}^\top \mathbf{V}_{i-1} \mathbf{r}_i + \mathbf{g})(\mathbf{V}_{i-1}^\top \mathbf{V}_{i-1} \mathbf{r}_i + \mathbf{g})^\top \\ &= \mathbf{V}_{i-1}^\top (\mathbf{S}_{i-1} + \mathbf{V}_{i-1} \mathbf{r}_i \mathbf{r}_i^\top \mathbf{V}_{i-1}^\top) \mathbf{V}_{i-1} + \mathbf{D}_{i-1} + \mathbf{g}\mathbf{g}^\top = \mathbf{V}_{i-1}^{'\top} \mathbf{S}_{i-1}^{'} \mathbf{V}_{i-1}^{'} + \mathbf{g}\mathbf{g}^\top + \mathbf{D}_{i-1} \end{aligned}$$

where we define $\mathbf{V}_{i-1}^{'} = \mathbf{Q}_{i-1} \mathbf{V}_{i-1}$ and defined $\mathbf{Q}_{i-1}$ in terms of the singular value decomposition, $\mathbf{Q}_{i-1}^\top \mathbf{S}_{i-1}^{'} \mathbf{Q}_{i-1} = \mathrm{svd}(\mathbf{S}_{i-1} + \mathbf{V}_{i-1} \mathbf{r}_i \mathbf{r}_i^\top \mathbf{V}_{i-1}^\top)$. Note that $\mathbf{S}_{i-1}^{'}$ is diagonal and nonnegative by construction. The current formula for $\boldsymbol{\Sigma}_i$ shows that we have a rank $(k+1)$ system (plus diagonal term) which needs to be converted back to a rank k system (plus diagonal term) which we denote by $\boldsymbol{\Sigma}_i^{'}$. We have two options as follows.
Case 1) Remove $\mathbf{g}$ from $\boldsymbol{\Sigma}_i$ to obtain

$$\boldsymbol{\Sigma}_i^{'} = \mathbf{V}_{i-1}^{'\top} \mathbf{S}_{i-1}^{'} \mathbf{V}_{i-1}^{'} + \mathbf{D}_{i-1} = \boldsymbol{\Sigma}_i - \mathbf{g}\mathbf{g}^\top = \boldsymbol{\Sigma}_i - c\mathbf{v}\mathbf{v}^\top$$

where $c = \|\mathbf{g}\|^2$ and $\mathbf{v} = \frac{1}{\|\mathbf{g}\|}\mathbf{g}$.
Case 2) Remove the m^{th} (smallest) eigenvalue in $\mathbf{S}_{i-1}^{'}$ and its corresponding eigenvector:

$$\boldsymbol{\Sigma}_i^{'} = \mathbf{V}_{i-1}^{'\top} \mathbf{S}_{i-1}^{'} \mathbf{V}_{i-1}^{'} + \mathbf{D}_{i-1} + \mathbf{g}\mathbf{g}^\top - \mathbf{S}^{'}(m,m) \mathbf{V}^{'}(m, \cdot)^\top \mathbf{V}^{'}(m, \cdot) = \boldsymbol{\Sigma}_i - c\mathbf{v}\mathbf{v}^\top$$

where $c = \mathbf{S}^{'}(m,m)$ and $\mathbf{v} = \mathbf{V}(m, \cdot)^{'}$.

Clearly, both cases can be written as an update of the form $\mathbf{\Sigma}_i^{'} = \mathbf{\Sigma}_i + c\mathbf{v}\mathbf{v}^\top$ where $c \geq 0$ and $\mathbf{v}^\top\mathbf{v} = 1$. We choose the case with smaller c value to minimize the change as we drop from a system of order $(k+1)$ to order k. Discarding the smallest singular value and its corresponding eigenvector would violate the bound. Instead, consider absorbing this term into the diagonal component to preserve the bound. Formally, we look for a diagonal matrix $\mathbf{F}$ such that $\mathbf{\Sigma}_i^{''} = \mathbf{\Sigma}_i^{'} + \mathbf{F}$ which also maintains $\mathbf{x}^\top\mathbf{\Sigma}_i^{''}\mathbf{x} \geq \mathbf{x}^\top\mathbf{\Sigma}_i\mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^d$. Thus, we want to satisfy:

$$\mathbf{x}^\top\mathbf{\Sigma}_i^{''}\mathbf{x} \geq \mathbf{x}^\top\mathbf{\Sigma}_i\mathbf{x} \iff \mathbf{x}^\top c\mathbf{v}\mathbf{v}^\top\mathbf{x} \leq \mathbf{x}^\top\mathbf{F}\mathbf{x} \iff c\left(\sum_{i=1}^{d}\mathbf{x}(i)\mathbf{v}(i)\right)^2 \leq \sum_{i=1}^{d}\mathbf{x}(i)^2\mathbf{F}(i)$$

where, for ease of notation, we take $\mathbf{F}(i) = \mathbf{F}(i,i)$.

Define $\mathbf{v}^{'} = \frac{1}{w}\mathbf{v}$ where $w = \mathbf{v}^\top\mathbf{1}$. Consider the case where $\mathbf{v} \geq \mathbf{0}$ though we will soon get rid of this assumption. We need an $\mathbf{F}$ such that $\sum_{i=1}^{d}\mathbf{x}(i)^2\mathbf{F}(i) \geq c\left(\sum_{i=1}^{d}\mathbf{x}(i)\mathbf{v}(i)\right)^2$. Equivalently, we need $\sum_{i=1}^{d}\mathbf{x}(i)^2\frac{\mathbf{F}(i)}{cw^2} \geq \left(\sum_{i=1}^{d}\mathbf{x}(i)\mathbf{v}(i)^{'}\right)^2$. Define $\mathbf{F}(i)^{'} = \frac{\mathbf{F}(i)}{cw^2}$ for all $i = 1, \ldots, d$. So, we need an $\mathbf{F}^{'}$ such that $\sum_{i=1}^{d}\mathbf{x}(i)^2\mathbf{F}(i)^{'} \geq \left(\sum_{i=1}^{d}\mathbf{x}(i)\mathbf{v}(i)^{'}\right)^2$. Using Lemma 3 it is easy to show that we may choose $\mathbf{F}^{'}(i) = \mathbf{v}(i)^{'}$. Thus, we obtain $\mathbf{F}(i) = cw^2\mathbf{F}(i)^{'} = cw\mathbf{v}(i)$. Therefore, for all $\mathbf{x} \in \mathbb{R}^d$, all $\mathbf{v} \geq \mathbf{0}$, and for $\mathbf{F}(i) = c\mathbf{v}(i)\sum_{j=1}^{d}\mathbf{v}(j)$ we have

$$\sum_{i=1}^{d}\mathbf{x}(i)^2\mathbf{F}(i) \geq c\left(\sum_{i=1}^{d}\mathbf{x}(i)\mathbf{v}(i)\right)^2. \tag{3}$$

To generalize the inequality to hold for all vectors $\mathbf{v} \in \mathbb{R}^d$ with potentially negative entries, it is sufficient to set $\mathbf{F}(i) = c|\mathbf{v}(i)|\sum_{j=1}^{d}|\mathbf{v}(j)|$. To verify this, consider flipping the sign of any $\mathbf{v}(i)$. The left side of the Inequality 3 does not change. For the right side of this inequality, flipping the sign of $\mathbf{v}(i)$ is equivalent to flipping the sign of $\mathbf{x}(i)$ and not changing the sign of $\mathbf{v}(i)$. However, in this case the inequality holds as shown before (it holds for any $\mathbf{x} \in \mathbb{R}^d$). Thus for all $\mathbf{x}, \mathbf{v} \in \mathbb{R}^d$ and for $\mathbf{F}(i) = c|\mathbf{v}(i)|\sum_{j=1}^{d}|\mathbf{v}(j)|$, Inequality 3 holds.

Supplement for Section 7

Small scale experiments In additional small-scale experiments, we compared Algorithm 2 with steepest descent (SD), conjugate gradient (CG), BFGS and Newton-Raphson. Small-scale problems may be interesting in real-time learning settings, for example, when a website has to learn from a user's uploaded labeled data in a split second to perform real-time retrieval. We considered logistic regression on five UCI data sets where missing values were handled via mean-imputation. A range of regularization settings $\lambda \in \{10^0, 10^2, 10^4\}$ was explored and all algorithms were initialized from the same ten random start-points. Table 3 shows the average number of seconds each algorithm needed to achieve the same solution that BFGS converged to (all algorithms achieve the same solution due to concavity). The bound is the fastest algorithm as indicated in bold.

$data\|\lambda$	$a\|10^0$	$a\|10^2$	$a\|10^4$	$b\|10^0$	$b\|10^2$	$b\|10^4$	$c\|10^0$	$c\|10^2$	$c\|10^4$	$d\|10^0$	$d\|10^2$	$d\|10^4$	$e\|10^0$	$e\|10^2$	$e\|10^4$
BFGS	1.90	0.89	2.45	3.14	2.00	1.60	4.09	1.03	1.90	5.62	2.88	3.28	2.63	2.01	1.49
SD	1.74	0.92	1.60	2.18	6.17	5.83	1.92	0.64	0.56	12.04	1.27	1.94	2.68	2.49	1.54
CG	0.78	0.83	0.85	0.70	0.67	0.83	0.65	0.64	0.72	1.36	1.21	1.23	0.48	0.55	0.43
Newton	0.31	0.25	0.22	0.43	0.37	0.35	0.39	0.34	0.32	0.92	0.63	0.60	0.35	0.26	0.20
Bound	**0.01**	**0.01**	**0.01**	**0.07**	**0.04**	**0.04**	**0.07**	**0.02**	**0.02**	**0.16**	**0.09**	**0.07**	**0.03**	**0.03**	**0.03**

Table 3: Convergence time in seconds under various regularization levels for a) Bupa ($t = 345, \mathsf{dim} = 7$), b) Wine ($t = 178, \mathsf{dim} = 14$), c) Heart ($t = 187, \mathsf{dim} = 23$), d) Ion ($t = 351, \mathsf{dim} = 34$), and e) Hepatitis ($t = 155, \mathsf{dim} = 20$) data sets.

Influence of rank k on bound performance in large scale experiments We also examined the influence of k on bound performance and compared it with LBFGS, SD and CG. Several choices

of k were explored. Table 4 shows results for the SRBCT data-set. In general, the bound performs best but slows down for superfluously large values of k. Steepest descent and conjugate gradient are slow yet obviously do not vary with k. Note that each iteration takes less time with smaller k for the bound. However, we are reporting overall runtime which is also a function of the number of iterations. Therefore, total runtime (a function of both) may not always decrease/increase with k.

k	1	2	4	8	16	32	64
LBFGS	1.37	1.32	1.39	1.35	1.46	1.40	1.54
SD	8.80	8.80	8.80	8.80	8.80	8.80	8.80
CG	4.39	4.39	4.39	4.39	4.39	4.39	4.39
Bound	**0.56**	**0.56**	0.67	0.96	1.34	2.11	4.57

Table 4: Convergence time in seconds as a function of k.

Additional latent-likelihood results For completeness, Figure 5 depicts two additional data-sets to complement Figure 2. Similarly, Table 5 shows all experimental settings explored in order to provide the summary Table 2 in the main article.

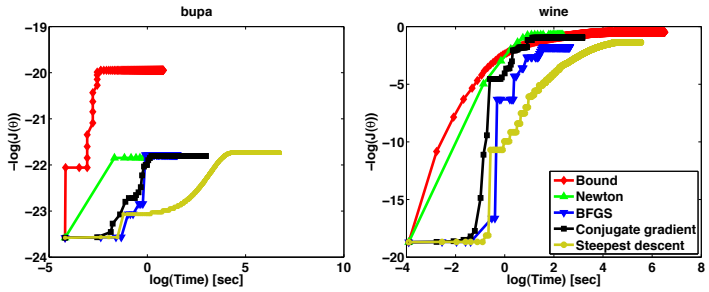

Figure 5: Convergence of test latent log-likelihood for bupa and wine data-sets.

Data-set	ion				bupa				hepatitis			
Algorithm	m = 1	m = 2	**m = 3**	m = 4	m = 1	**m = 2**	m = 3	m = 4	m = 1	**m = 2**	m = 3	m = 4
BFGS	-4.96	-5.55	-5.88	-5.79	-22.07	-21.78	-21.92	-21.87	-4.42	-5.28	-4.95	-4.93
SD	-11.80	-9.92	-5.56	-8.59	-21.76	-21.74	-21.73	-21.83	-4.93	-5.14	-5.01	-5.20
CG	-5.47	-5.81	-5.57	-5.22	-21.81	-21.81	-21.81	-21.81	-4.84	-4.84	-4.84	-4.84
Newton	-5.95	-5.95	-5.95	-5.95	-21.85	-21.85	-21.85	-21.85	-5.50	-5.50	-5.50	-4.50
Bound	-6.08	-4.84	**-4.18**	-5.17	-21.85	**-19.95**	-20.01	-19.97	-5.47	**-4.40**	-4.75	-4.92

Data-set	wine				SRBCT			
Algorithm	m = 1	m = 2	**m = 3**	m = 4	m = 1	m = 2	m = 3	**m = 4**
BFGS	-0.90	-0.91	-1.79	-1.35	-5.99	-6.17	-6.09	-6.06
SD	-1.61	-1.60	-1.37	-1.63	-5.61	-5.62	-5.62	-5.61
CG	-0.51	-0.78	-0.95	-0.51	-5.62	-5.49	-5.36	-5.76
Newton	-0.71	-0.71	-0.71	-0.71	-5.54	-5.54	-5.54	-5.54
Bound	-0.51	-0.51	**-0.48**	-0.51	-5.31	-5.31	-4.90	**-0.11**

Table 5: Test latent log-likelihood at convergence for different values of $m \in \{1, 2, 3, 4\}$ on ion, bupa, hepatitis, wine and SRBCT data-sets.