Approximate Inference for large Ising models with random coupling Matrices

Manfred Opper (TUB), Burak Cakmak (Aalborg), Ole Winther (DTU)

December 17, 2015

The problem

- Approximate inference using Expectation Propagation (EP) usually gives excellent results for Gaussian latent variable models.
- EP can become inefficient if model is very large.
- Assumptions on 'randomness' of the problem leads to simplification of EP fixed points (TAP equations).
- New: Algorithms that converge to these fixed points.
- More details in arXiv:1509.01229 [cond-mat.dis-nn]

Example: Compressed sensing

$\mathbf{Y} = \mathbf{AX} + \boldsymbol{\epsilon}$ with $K \times N$ matrix $\mathbf{A}$ and Gaussian noise $\boldsymbol{\epsilon}$.

Sparsity (spike & slab) prior $p_0(\mathbf{x}) = \prod_{k=1}^{N} \left((1-\rho)\delta(x_k) + \frac{\rho}{\sqrt{2\pi\sigma^2}} e^{-\frac{x_k^2}{2\sigma^2}} \right)$

1. Message passing algorithm (Donoho, Maleki, Montanari, 2009)
2. Analysis by statistical mechanics, phase diagrams, achieving of thresholds (Krzakala, Mézard, Sausset, Sun, Zdeborová, 2012)
3. Rigorous analysis for $\mathbf{A}$ with random i.i.d. matrix elements (Bayati, Montanari, 2011, Bayati, Lelarge, Montanari 2015).
4. Approximate inference for other random matrix ensembles (Cakmak, Winther, Fleury, 2014)

Simplest model: Ising

$\mathbf{S} = (S_1, \ldots, S_N) \in \{\pm 1\}^N$

$$P(\mathbf{S}) = \frac{1}{Z} \exp\left[\sum_{i<j}^{N} J_{ij} S_i S_j + \sum_{i}^{N} h_i S_i\right]$$

Try to compute marginals $m_i \doteq < S_i >$.

Gaussian Expectation - Propagation for Ising models

Write Ising model as $p(\mathbf{S}) \propto \exp\left[\sum_{i<j}^{N} J_{ij} S_i S_j\right] \prod_k f_k(S_k)$ where $f_k(S) = e^{h_k S}\left\{\delta(S-1) + \delta(S+1)\right\}$

Gaussian Expectation - Propagation for Ising models

Write Ising model as $p(\mathbf{S}) \propto \exp\left[\sum_{i<j}^{N} J_{ij} S_i S_j\right] \prod_k f_k(S_k)$ where $f_k(S) = e^{h_k S}\left\{\delta(S-1) + \delta(S+1)\right\}$ Approximate by Gaussian $q(\mathbf{S}) \propto \exp\left[\sum_{i<j}^{N} J_{ij} S_i S_j\right] \prod_{k=1}^{N} g_k(S_k)$ where $g_k(S) = e^{\gamma_k S - \frac{1}{2}\lambda_k S^2}$.

Gaussian Expectation - Propagation for Ising models

Write Ising model as $p(\mathbf{S}) \propto \exp\left[\sum_{i<j}^{N} J_{ij} S_i S_j\right] \prod_k f_k(S_k)$ where $f_k(S) = e^{h_k S}\left\{\delta(S-1) + \delta(S+1)\right\}$ Approximate by Gaussian $q(\mathbf{S}) \propto \exp\left[\sum_{i<j}^{N} J_{ij} S_i S_j\right] \prod_{k=1}^{N} g_k(S_k)$ where $g_k(S) = e^{\gamma_k S - \frac{1}{2}\lambda_k S^2}$.

Repeat until convergence: Choose i

1. remove term g_i

$$q_{\setminus i}(\mathbf{x}) \propto q(\mathbf{S})/g_i(S_i)$$

Gaussian Expectation - Propagation for Ising models

Write Ising model as $p(\mathbf{S}) \propto \exp\left[\sum_{i<j}^{N} J_{ij} S_i S_j\right] \prod_k f_k(S_k)$ where $f_k(S) = e^{h_k S}\left\{\delta(S-1)+\delta(S+1)\right\}$ Approximate by Gaussian $q(\mathbf{S}) \propto \exp\left[\sum_{i<j}^{N} J_{ij} S_i S_j\right] \prod_{k=1}^{N} g_k(S_k)$ where $g_k(S) = e^{\gamma_k S - \frac{1}{2}\lambda_k S^2}$.

Repeat until convergence: Choose i

1. remove term g_i

$$q_{\setminus i}(\mathbf{x}) \propto q(\mathbf{S})/g_i(S_i)$$

2. Update: (tilted distribution)

$$\tilde{q}_i(\mathbf{S}) \propto f_i(S_i) q_{\setminus i}(\mathbf{S})$$

3. Project:

$$q^{\text{new}}(\mathbf{S}) = \text{Project}(\tilde{q}_i, q)$$

Gaussian Expectation - Propagation for Ising models

Write Ising model as $p(\mathbf{S}) \propto \exp\left[\sum_{i<j}^{N} J_{ij} S_i S_j\right] \prod_k f_k(S_k)$ where $f_k(S) = e^{h_k S}\{\delta(S-1)+\delta(S+1)\}$ Approximate by Gaussian $q(\mathbf{S}) \propto \exp\left[\sum_{i<j}^{N} J_{ij} S_i S_j\right] \prod_{k=1}^{N} g_k(S_k)$ where $g_k(S) = e^{\gamma_k S - \frac{1}{2}\lambda_k S^2}$.

Repeat until convergence: Choose i

1. remove term g_i

$$q_{\setminus i}(\mathbf{x}) \propto q(\mathbf{S})/g_i(S_i)$$

2. Update: (tilted distribution)

$$\tilde{q}_i(\mathbf{S}) \propto f_i(S_i) q_{\setminus i}(\mathbf{S})$$

3. Project:

$$q^{\text{new}}(\mathbf{S}) = \text{Project}(\tilde{q}_i, q)$$

4. Refine term:

$$g_i^{\text{new}}(S_i) \propto \frac{q^{\text{new}}(\mathbf{S})}{q_{\setminus i}(\mathbf{S})}$$

Random matrix ensembles

Assume that coupling matrix is random and has the representation

$$\mathbf{J} = \mathbf{O}^\dagger \mathbf{\Lambda} \mathbf{O}$$

where $\mathbf{O}$ is random orthogonal, independent of diagonal matrix $\mathbf{\Lambda}$.

Random matrix ensembles

Assume that coupling matrix is random and has the representation

$$\mathbf{J} = \mathbf{O}^{\dagger}\mathbf{\Lambda}\mathbf{O}$$

where $\mathbf{O}$ is random orthogonal, independent of diagonal matrix $\mathbf{\Lambda}$. The distribution of $\mathbf{J}$ is determined by the generating function

$$\mathrm{G}(\mathbf{Q}) \triangleq \lim_{N\to\infty} \frac{1}{N} \log \left\langle e^{\frac{N}{2}\mathrm{tr}(\mathbf{QJ})} \right\rangle_{\mathbf{J}}$$

We also define the R–transform

$$\mathrm{R}(x) = 2\frac{d\mathrm{G}(x)}{dx} = \sum_{n=1}^{\infty} c_n x^{n-1}$$

and its inverse

$$\mathrm{R}^{-1}(x) = \sum_{n=1}^{\infty} a_n x^n$$

Random matrix ensembles and their R–transforms

$$\mathrm{R}(x) = \beta^2 x \quad \text{for} \quad J_{ij} \sim \mathcal{N}(0, \beta^2/N)$$

$$\mathrm{R}(x) = \frac{\beta^2 \alpha x}{1 + \beta \alpha x} \quad \text{for} \quad -\mathbf{J} = \text{ central Wishart}$$

$$\mathrm{R}(x) = \frac{-1 + \sqrt{1 + 4\beta^2 x^2}}{2x} \quad \text{for} \quad \mathbf{J} = \beta \mathbf{O}^\dagger \boldsymbol{\Lambda} \mathbf{O} \text{ with diagonal } \boldsymbol{\Lambda} = \pm 1$$

TAP equations

- For large invariant random matrices, one can show that EP fixed points converge to those of the TAP equations (Opper and Winther PRE 2001, Opper, Cakmak, Winther 2015)

$$\mathbf{m} = \tanh(\boldsymbol{\psi})$$
$$\boldsymbol{\psi} = \mathbf{h} + \mathbf{J}\mathbf{m} - \mathrm{R}(1-q)\mathbf{m}$$

where $q \triangleq \frac{1}{N}\mathbf{m}^{\dagger}\mathbf{m}$ (Parisi and Potters 1995).

- No matrix inversions !
- How can we solve these equations efficiently?

Analyse algorithms

- Candidate algorithm could be of the form

$$\mathbf{m}(t) = \tanh\left(\{\gamma(\tau), \mathbf{m}(\tau)\}_{\tau=0}^{t-1}\right)$$
$$\gamma(t) = \mathbf{h} + \mathbf{J}\mathbf{m}(t)$$

- Average case analysis: use generating functional $\langle Z(\{\mathbf{l}(t)\})\rangle_{\mathbf{J}}$ where

$$Z(\{\mathbf{l}(t)\}) = \int \prod_{t=0}^{T-1} \Big\{ \mathrm{d}\mathbf{m}(t)\mathrm{d}\gamma(t)\, \delta(\mathbf{m}(t) - \tanh\left(\{\gamma(\tau), \mathbf{m}(\tau)\}_{\tau=0}^{t-1}\right)) \\ \delta(\gamma(t) - \mathbf{h} - \mathbf{J}\mathbf{m}(t)) e^{i\gamma(t)^{\dagger}\mathbf{l}(t)} \Big\} .$$

Performing the average over the random matrix

With some effort we can compute the averaged generating functional for $N \to \infty$

$$\langle Z(\{\mathbf{l}(t)\})\rangle_{\mathbf{J}} \simeq \prod_{n=1}^{N} \int \mathrm{d}\mathcal{N}(\{\phi_n(t)\}; 0, \mathcal{C}_\phi)$$

$$\prod_{t=0}^{T-1} \Big\{ \mathrm{d}m_n(t)\mathrm{d}\gamma_n(t)\ \delta(m_n(t) - \tanh\{m_n(\tau), \gamma_n(\tau)\}_{\tau=0}^{t-1})$$

$$\delta\left(\gamma_n(t) - h_n - \sum_{s<t} \hat{\mathcal{G}}(t,s) m_n(s) - \phi_n(t)\right) e^{i\gamma_n(t) l_n(t)} \Big\}$$

with $\mathcal{N}(\cdot; \mu, \Sigma)$ denoting the multivariate normal distribution with mean μ and covariance Σ.

Effective stochastic dynamics with memory

$$\mathbf{m}(t) = \tanh\left(\{\boldsymbol{\gamma}(\tau), \mathbf{m}(\tau)\}_{\tau=0}^{t-1}\right)$$
$$\boldsymbol{\gamma}(t) = \mathbf{h} + \sum_{\tau=0}^{t-1} \hat{\mathcal{G}}(t,s)\mathbf{m}(\tau) + \phi(t) \ .$$

with $\phi(t)$ independent discrete time Gaussian process and

Effective stochastic dynamics with memory

$$\mathbf{m}(t) = \tanh\left(\{\boldsymbol{\gamma}(\tau), \mathbf{m}(\tau)\}_{\tau=0}^{t-1}\right)$$

$$\boldsymbol{\gamma}(t) = \mathbf{h} + \sum_{\tau=0}^{t-1} \hat{\mathcal{G}}(t,s)\mathbf{m}(\tau) + \boldsymbol{\phi}(t)\ .$$

with $\phi(t)$ independent discrete time Gaussian process and

$$\hat{\mathcal{G}} = \mathrm{R}(\mathcal{G}) \qquad \mathcal{C}_\phi = \sum_{n=1}^{\infty} c_n \sum_{k=0}^{n-2} \mathcal{G}^k \mathcal{C}(\mathcal{G}^\dagger)^{n-2-k}$$

$$\mathcal{G}(t,\tau) = \frac{1}{N}\sum_{i=1}^{N}\left\langle \frac{\partial m_i(t)}{\partial \phi_i(\tau)} \right\rangle_{\phi_i} \qquad \mathcal{C}(t,\tau) = \frac{1}{N}\sum_{i=1}^{N} \langle m_i(t) m_i(\tau) \rangle_{\phi_i}\ .$$

Effective stochastic dynamics with memory

$$\mathbf{m}(t) = \tanh\left(\{\boldsymbol{\gamma}(\tau), \mathbf{m}(\tau)\}_{\tau=0}^{t-1}\right)$$
$$\boldsymbol{\gamma}(t) = \mathbf{h} + \sum_{\tau=0}^{t-1} \hat{\mathcal{G}}(t,s)\mathbf{m}(\tau) + \boldsymbol{\phi}(t) \ .$$

with $\phi(t)$ independent discrete time Gaussian process and

$$\hat{\mathcal{G}} = \mathrm{R}(\mathcal{G}) \qquad \mathcal{C}_\phi = \sum_{n=1}^{\infty} c_n \sum_{k=0}^{n-2} \mathcal{G}^k \mathcal{C}(\mathcal{G}^\dagger)^{n-2-k}$$

$$\mathcal{G}(t,\tau) = \frac{1}{N}\sum_{i=1}^{N}\left\langle \frac{\partial m_i(t)}{\partial \phi_i(\tau)} \right\rangle_{\phi_i} \qquad \mathcal{C}(t,\tau) = \frac{1}{N}\sum_{i=1}^{N} \langle m_i(t) m_i(\tau)\rangle_{\phi_i} \ .$$

Memory could be bad for convergence !

Single step memory algorithm

Idea: Try to **kill the memory** terms, i.e. require

$$\hat{\mathcal{G}}(t,\tau) = 0, \forall \tau \neq t-1 \qquad \hat{\mathcal{G}}(t,t-1) = \frac{1-q(t)}{1-q(t-1)}\mathrm{R}(1-q(t-1))$$

This leads to

$$\mathbf{m}(t+1) = \tanh(\boldsymbol{\psi}(t))$$

$$\boldsymbol{\psi}(t) = Q(t)\sum_{\tau=0}^{t} a_{t+1-\tau}\mathbf{u}(\tau)$$

$$\mathbf{u}(t) = \frac{\mathbf{h} + \mathbf{J}\mathbf{m}(t) - \hat{\mathcal{G}}(t,t-1)\mathbf{m}(t-1)}{Q(t-1)(1-q(t))}$$

where we define

$$Q(t) = \prod_{\tau=0}^{t}\mathrm{R}(1-q(\tau)) \text{ and the coefficients } a_k \text{ via } \mathrm{R}^{-1}(x) = \sum_{n=1}^{\infty} a_n x^n \; .$$

with $Q(-1) = 1$. The algorithm initialises with $\mathbf{m}(t) = \mathbf{0}$ for $t \in \{-1, 0\}$.

Examples

- $\mathbf{J}$ Gaussian i.i.d. (Sherrington Kirkpatrick model)

$$\mathbf{m}(t+1) = \tanh(\mathbf{h} + \mathbf{J}\mathbf{m}(t) - \beta^2(1-q(t))\mathbf{m}(t-1))$$

  agrees with Bolthausen's (2014) result

- $-\mathbf{J} \sim$ Wishart

$$\mathbf{m}(t+1) = \tanh(\mathbf{z}(t) + A(t)\mathbf{m}(t))$$

$$\mathbf{z}(t) = \frac{1}{\beta}A(t)[\mathbf{h} + (\mathbf{J} - \beta\mathbf{I})\mathbf{m}(t)] + \alpha(1-q(t))A(t)\mathbf{z}(t-1)$$

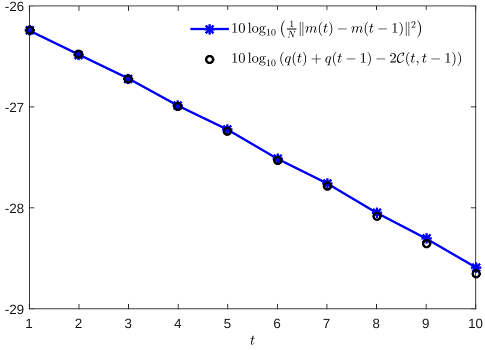
  where

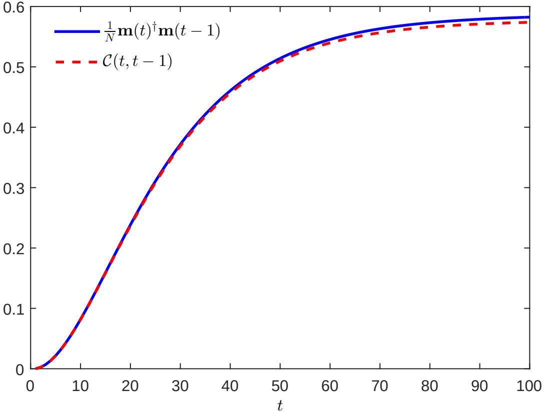
$$A(t) \triangleq \frac{\mathrm{R}(1-q(t))}{\beta\alpha(1-q(t))} = \frac{\beta}{1+\beta\alpha(1-q(t))}.$$

  Coincides with AMP algorithm introduced by Kabashima (2003) in the context of the CDMA and by Donoho, Maleki, Montanari (2009) for compressed sensing.

Random orthogonal ensemble: Analytical results vs Simulation

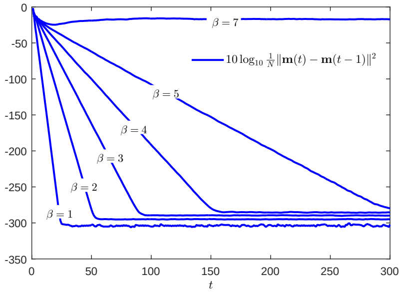
$\mathrm{R}^{-1}(x) = x/(\beta^2 - x^2)$ and $a_n = \frac{1}{\beta^{n+1}}$ for n odd and $a_n = 0$ else.
$N = 2^{14}$, $\beta = 20$ and $h_i = 1$, single realisation of $\mathbf{J}$.



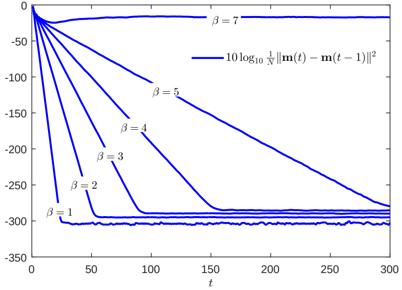


$\frac{1}{N}\mathbf{m}(t)^{\dagger}\mathbf{m}(t-1)$
$\mathcal{C}(t,t-1)$
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90
100
t

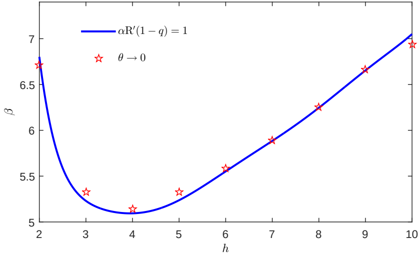
Convergence for $N = 2^{14}$, $h_i = 2$ (Simulations)



Convergence for $N = 2^{14}$, $h_i = 2$ (Simulations)

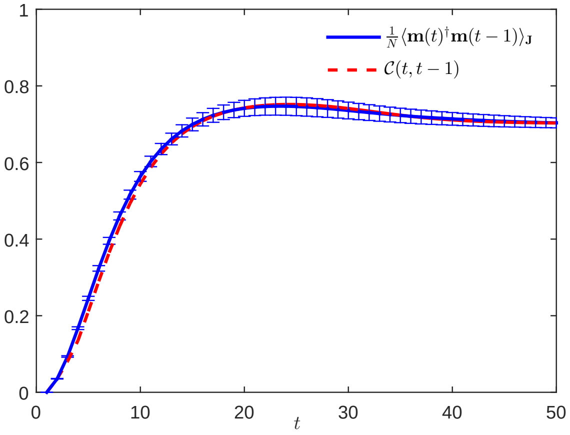


Stability of fixed point (Almeida–Thouless line)



Analytical results vs Simulation

Random orthogonal ensemble, region of instability: $N = 2^{12}$, $\beta = 10$ and $h_i = 2$, 5×10^3 Realisations of $\mathbf{J}$



Outlook

- Try to make things rigorous (random matrix theory).
- Extend to other Gaussian latent variable models.
- Estimate R – transform from real data.
- Develop a new algorithm for (full) EP fixed–points using idea of memory cancellation.