

Chalmers Machine Learning Summer School
Approximate message passing and biomedicine

Part 1: Expectation Propagation

Tom Heskes

Machine Learning Group, Institute for Computing and Information Sciences
Radboud University Nijmegen, The Netherlands

April 15, 2015

Outline

Bayesian Machine Learning

- Probabilistic modeling

- Approximate inference

Expectation Propagation

- Bit of history

- Factor graphs

- Iterative procedure

EP for Gaussian process classification

- Locality property

- Step by step

Conclusion

Outline

Bayesian Machine Learning

- Probabilistic modeling

- Approximate inference

Expectation Propagation

- Bit of history

- Factor graphs

- Iterative procedure

EP for Gaussian process classification

- Locality property

- Step by step

Conclusion

Bayesian Machine Learning

- ▶ Enumerate all ‘reasonable’ models θ and assign a **prior belief** $p(\theta)$.
- ▶ Upon observing the data \mathcal{D} , compute the **likelihood** $p(\mathcal{D}|\theta)$.
- ▶ Compute the **posterior probability** over models using Bayes’ rule:

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta) .$$

Problem: the posterior distribution is often **intractable**.



Approximate Inference

- ▶ Stochastic sampling methods
 - ▶ Markov Chain Monte Carlo sampling
 - ▶ Gibbs sampling
 - ▶ Particle filtering
- ▶ Deterministic methods
 - ▶ Variational ('mean-field') approaches
 - ▶ Loopy belief propagation
 - ▶ **Expectation propagation**

Outline

Bayesian Machine Learning

- Probabilistic modeling

- Approximate inference

Expectation Propagation

- Bit of history

- Factor graphs

- Iterative procedure

EP for Gaussian process classification

- Locality property

- Step by step

Conclusion

Expectation Propagation

- ▶ Message passing algorithm, invented by Thomas Minka (PhD thesis, 2001).
- ▶ Its generalization, power EP, contains a large class of deterministic algorithms for approximate inference.
- ▶ Arguably the best approximate inference results, if it converges. . .
- ▶ Implemented in Microsoft's [infer.net](#).



VMP

EP, BP

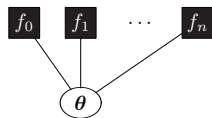
Factors

- ▶ Many probabilistic models factorize, i.e., can be written in the form

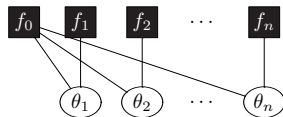
$$p(\mathcal{D}, \boldsymbol{\theta}) = \prod_i f_i(\boldsymbol{\theta}).$$

- ▶ For example, with independently, identically distributed data, there is one factor $f_n(\boldsymbol{\theta}) = p(\mathbf{x}_n|\boldsymbol{\theta})$ for each data point \mathbf{x}_n along with a factor $f_0(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$ for the prior.
- ▶ This also applies to Gaussian process regression and classification: $\boldsymbol{\theta}$ is drawn from a Gaussian process prior and each of the factors further simplifies into $f_n(\boldsymbol{\theta}) = p(\mathbf{x}_n|\theta_n)$.

Factor graph



General case



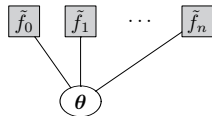
Gaussian processes

Approximation

- ▶ Approximate the posterior by an exponential distribution:

$$p(\mathcal{D}, \boldsymbol{\theta}) = \prod_i f_i(\boldsymbol{\theta}) \approx \frac{1}{Z} \prod_i \tilde{f}_i(\boldsymbol{\theta}) = \tilde{p}(\boldsymbol{\theta}).$$

- ▶ I.e., approximate each term $f_i(\boldsymbol{\theta})$ by an “exponential” term approximation $\tilde{f}_i(\boldsymbol{\theta})$.
- ▶ Terms in exponential form, often the prior, do not have to be approximated.



Exponential form:

$$\tilde{f}(\boldsymbol{\theta}) = h(\boldsymbol{\theta})g(\boldsymbol{\eta}) \exp \left[\boldsymbol{\eta}^T \mathbf{u}(\boldsymbol{\theta}) \right],$$

natural parameters $\boldsymbol{\eta}$ and sufficient statistics $\mathbf{u}(\boldsymbol{\theta})$.

Iterative Updating

- ▶ **Take out** term approximation i :

$$\tilde{p}_{\setminus i}(\boldsymbol{\theta}) \propto \prod_{j \neq i} \tilde{f}_j(\boldsymbol{\theta}).$$

- ▶ **Put back** in term i :

$$\tilde{p}^{(i)}(\boldsymbol{\theta}) \propto f_i(\boldsymbol{\theta}) \prod_{j \neq i} \tilde{f}_j(\boldsymbol{\theta}).$$

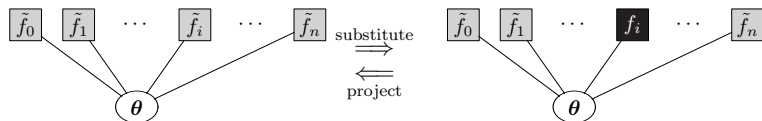
- ▶ **Match moments**, i.e., find the approximate distribution of exponential form such that

$$\int d\boldsymbol{\theta} \mathbf{u}(\boldsymbol{\theta}) \tilde{p}^{\text{new}}(\boldsymbol{\theta}) = \int d\boldsymbol{\theta} \mathbf{u}(\boldsymbol{\theta}) \tilde{p}^{(i)}(\boldsymbol{\theta}).$$

- ▶ **Bookkeeping**: set the new term approximation such that

$$\tilde{p}^{\text{new}}(\boldsymbol{\theta}) \propto \tilde{f}_i^{\text{new}}(\boldsymbol{\theta}) \prod_{j \neq i} \tilde{f}_j(\boldsymbol{\theta}).$$

Going Back and Forth



- ▶ Project: minimize the **KL-divergence**

$$\text{KL}(\tilde{p}^{(i)}, \tilde{p}) = \int d\theta \tilde{p}^{(i)}(\theta) \log \left[\frac{\tilde{p}^{(i)}(\theta)}{\tilde{p}(\theta)} \right].$$

- ▶ Equivalent to moment matching when \tilde{p} is in the **exponential family**.

Outline

Bayesian Machine Learning

- Probabilistic modeling

- Approximate inference

Expectation Propagation

- Bit of history

- Factor graphs

- Iterative procedure

EP for Gaussian process classification

- Locality property

- Step by step

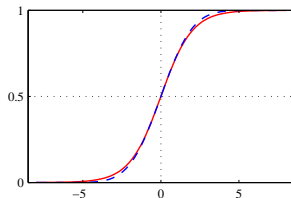
Conclusion

Gaussian Processes for Classification

- ▶ $f_0(\boldsymbol{\theta})$ is a Gaussian prior. No need to approximate.
- ▶ $f_i(\boldsymbol{\theta}) = f_i(\theta_i)$, some nonlinear sigmoidal function of θ_i .
- ▶ Each term approximation is a one-dimensional Gaussian form,

$$\tilde{f}_i(\theta_i) = \exp \left[h_i \theta_i - \frac{1}{2} K_i \theta_i^2 \right],$$

not necessarily normalizable: K_i may be negative.



$$f_i(\theta_i) = \sigma(y_i \theta_i)$$

Locality Property (1)

- ▶ Consider updating the term approximation $\tilde{f}_i(\theta_i)$. After replacing the old term approximation by the term we have

$$\tilde{p}^{(i)}(\boldsymbol{\theta}) \propto \tilde{p}_{\setminus i}(\boldsymbol{\theta}) f_i(\theta_i) \propto \tilde{p}_{\setminus i}(\boldsymbol{\theta}_{\setminus i} | \theta_i) \tilde{p}_{\setminus i}(\theta_i) f_i(\theta_i).$$

- ▶ We have to find the new approximation $\tilde{p}^{\text{new}}(\boldsymbol{\theta})$ closest in KL-divergence to $\tilde{p}^{(i)}(\boldsymbol{\theta})$:

$$\begin{aligned} \text{KL}(\tilde{p}^{(i)}, \tilde{p}^{\text{new}}) &= \int d\boldsymbol{\theta} \tilde{p}^{(i)}(\boldsymbol{\theta}) \log \left[\frac{\tilde{p}^{(i)}(\boldsymbol{\theta})}{\tilde{p}^{\text{new}}(\boldsymbol{\theta})} \right] \\ &= \int d\theta_i \tilde{p}^{(i)}(\theta_i) \log \left[\frac{\tilde{p}^{(i)}(\theta_i)}{\tilde{p}^{\text{new}}(\theta_i)} \right] \\ &\quad + \int d\theta_i \tilde{p}^{(i)}(\theta_i) \int d\boldsymbol{\theta}_{\setminus i} \tilde{p}^{(i)}(\boldsymbol{\theta}_{\setminus i} | \theta_i) \log \left[\frac{\tilde{p}^{(i)}(\boldsymbol{\theta}_{\setminus i} | \theta_i)}{\tilde{p}^{\text{new}}(\boldsymbol{\theta}_{\setminus i} | \theta_i)} \right]. \end{aligned}$$

Locality Property (2)

From previous slide:

$$\begin{aligned} \text{KL}(\tilde{p}^{(i)}, \tilde{p}^{\text{new}}) &= \int d\theta_i \tilde{p}^{(i)}(\theta_i) \log \left[\frac{\tilde{p}^{(i)}(\theta_i)}{\tilde{p}^{\text{new}}(\theta_i)} \right] \\ &+ \int d\theta_i \tilde{p}^{(i)}(\theta_i) \int d\boldsymbol{\theta}_{\setminus i} \tilde{p}^{(i)}(\boldsymbol{\theta}_{\setminus i}|\theta_i) \log \left[\frac{\tilde{p}^{(i)}(\boldsymbol{\theta}_{\setminus i}|\theta_i)}{\tilde{p}^{\text{new}}(\boldsymbol{\theta}_{\setminus i}|\theta_i)} \right]. \end{aligned}$$

Consequences:

1. At the optimum $\tilde{p}^{\text{new}}(\boldsymbol{\theta}_{\setminus i}|\theta_i) = \tilde{p}^{(i)}(\boldsymbol{\theta}_{\setminus i}|\theta_i)$, which means that only \tilde{K}_{ii} and \tilde{h}_i can change.
2. We only need to match moments for the marginal $\tilde{p}(\theta_i)$.

Take Out

- ▶ Easy in terms of canonical parameters,

$$\mathbf{K}_{\setminus i} = \mathbf{K} - \tilde{K}_i \mathbf{1}_i \mathbf{1}_i^T \quad \text{and} \quad \mathbf{h}_{\setminus i} = \mathbf{h} - \tilde{h}_i \mathbf{1}_i,$$

with $\mathbf{1}_i$ a vector with a 1 at element i and the rest 0.

- ▶ We need the corresponding moment form with (only) $C_{ii}^{\setminus i}$ and $m_i^{\setminus i}$.
- ▶ Efficiently with [Sherman-Morrison](#) formula (see next slide):

$$C_{ii}^{\setminus i} = C_{ii} + \frac{C_{ii} C_{ii} \tilde{K}_i}{1 - C_{ii} \tilde{K}_i} \left[1/C_{ii} - \tilde{K}_i \right]^{-1}.$$

- ▶ The new mean follows from

$$m_i^{\setminus i} = m_i + C_{ii}^{\setminus i} \left[-\tilde{h}_i + \tilde{K}_{ii} m_i \right].$$

- ▶ Computational complexity is order 1 per term, i.e., order N in total per iteration of EP.

Sherman-Morrison Formula

- ▶ Efficient way to recompute the inverse after adding a lower-dimensional part:

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}},$$

- ▶ The result on the previous slide follows by setting $\mathbf{u} = -\tilde{K}_i\mathbf{1}_i$ and $\mathbf{v} = \mathbf{1}_i$.
- ▶ **Woodbury formula**: generalization to matrices in terms of vectors.
- ▶ **Matrix determinant lemma** does something similar for (log) determinants.

See the [Matrix Cookbook](#) (or Wikipedia...).

Match Moments

- ▶ We have to compute

$$\int d\theta_i \mathcal{N}(\theta_i; m_i^{\setminus i}, C_{ii}^{\setminus i}) f_i(\theta_i) \{1, \theta_i, \theta_i^2\}.$$

- ▶ One-dimensional integrals that (sometimes) can be computed analytically, and otherwise approximated with Gauss-Hermite quadrature, i.e., from

$$\sum_k w_k f_i(m_i^{\setminus i} + \sqrt{C_{ii}^{\setminus i}} x_k) \{1, x_k, x_k^2\}.$$

- ▶ This then yields the new m_i^{new} and C_{ii}^{new} .
- ▶ Computational complexity is order W , the number of quadrature points per term, i.e., order NW per EP iteration.

Bookkeeping

- ▶ Now that we have the new moments, we have to find new term approximations that give exactly those **same moments**.
- ▶ It can be shown that the updates are simply as if we are in a one-dimensional situation:

$$\tilde{K}_{ii}^{\text{new}} = \tilde{K}_{ii} + [1/C_{ii}^{\text{new}} - 1/C_{ii}] ,$$

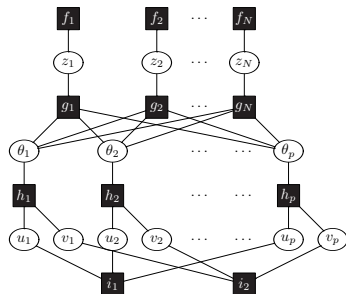
and similarly

$$\tilde{h}_i^{\text{new}} = \tilde{h}_i + [m_i^{\text{new}}/C_{ii}^{\text{new}} - m_i/C_{ii}] .$$

- ▶ Keep track of **C** and **m** using **Sherman-Morrison**, but now applied to the whole matrix.
- ▶ Computational complexity is order N^2 per term, i.e., N^3 per EP iteration.

Sequential vs. parallel

- ▶ Initial formulation of expectation propagation: **sequentially** update terms and keep track of approximated posterior.
- ▶ Viewed as a mapping from old to new term approximations, we may as well do this **in parallel**.
- ▶ Advantages: much, much faster for **sparse precision matrices**; numerically more stable.
- ▶ Disadvantage: convergence might be a bit slower.



Outline

Bayesian Machine Learning

- Probabilistic modeling

- Approximate inference

Expectation Propagation

- Bit of history

- Factor graphs

- Iterative procedure

EP for Gaussian process classification

- Locality property

- Step by step

Conclusion

Other Issues

- ▶ By keeping track of normalizations, we can also approximate the **model evidence** and use that for optimizing hyperparameters.
- ▶ **Power EP**: take out the term proxy/put back the term to power α . Standard EP/loopy belief propagation: $\alpha = 1$. Variational message passing: $\alpha = 0$. α just below 1 happens to be more stable than $\alpha = 1$.
- ▶ Convergence is a (big) issue: in particular there is no guarantee on **normalizability**.
- ▶ Many, many more applications: mixture models, nonlinear Kalman filters, Dirichlet models, Plackett-Luce, ...
- ▶ Consistently more accurate than **Laplace approximations**; ongoing efforts to speed it up.