4F13: Machine Learning

Lecture 10: Variational Approximations

Zoubin Ghahramani
zoubin@eng.cam.ac.uk

Department of Engineering
University of Cambridge

Michaelmas, 2006

http://learning.eng.cam.ac.uk/zoubin/ml06/

Motivation

Many statistical inference problems result in intractable computations...

- Bayesian posterior over model parameters:

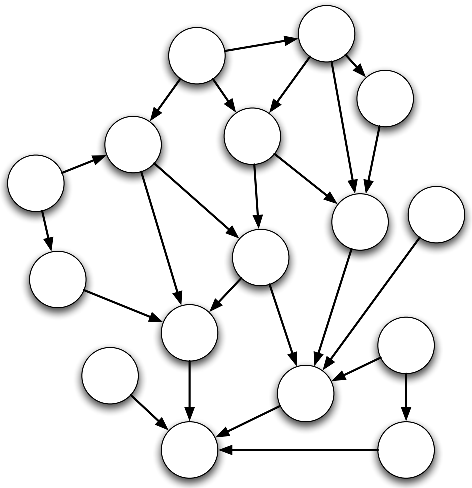
$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})}$$

- Computing posterior over hidden variables (e.g. for E step of EM):

$$P(H|V,\theta) = \frac{P(V|H,\theta)P(H|\theta)}{P(V|\theta)}$$

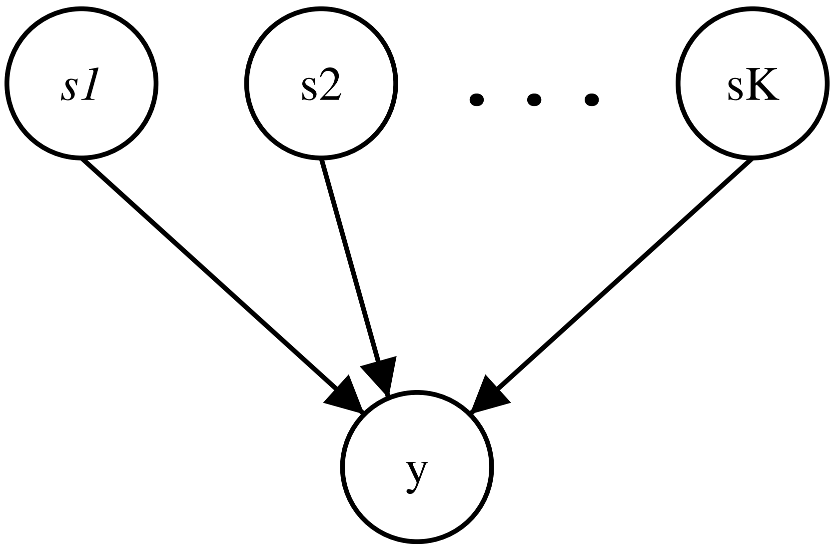
- Computing marginals in a multiply-connected graphical models:

$$P(x_i|x_j = e) = \sum_{\mathbf{x}\backslash\{x_i,x_j\}} P(\mathbf{x}|x_j = e)$$



Solutions: Markov chain Monte Carlo, variational approximations

Example: Binary latent factor model



Model with K binary latent variables, $s_i \in \{0, 1\}$,
organised into a vector $\mathbf{s} = (s_1, \ldots, s_K)$
real-valued observation vector $\mathbf{y}$
parameters $\boldsymbol{\theta} = \{\{\boldsymbol{\mu}_i, \pi_i\}_{i=1}^K, \sigma^2\}$

$\mathbf{s} \sim$ Bernoulli
$\mathbf{y}|\mathbf{s} \sim$ Gaussian

$$p(\mathbf{s}|\boldsymbol{\pi}) = p(s_1, \ldots, s_K|\boldsymbol{\pi}) = \prod_{i=1}^K p(s_i|\pi_i) = \prod_{i=1}^K \pi_i^{s_i}(1-\pi_i)^{(1-s_i)}$$

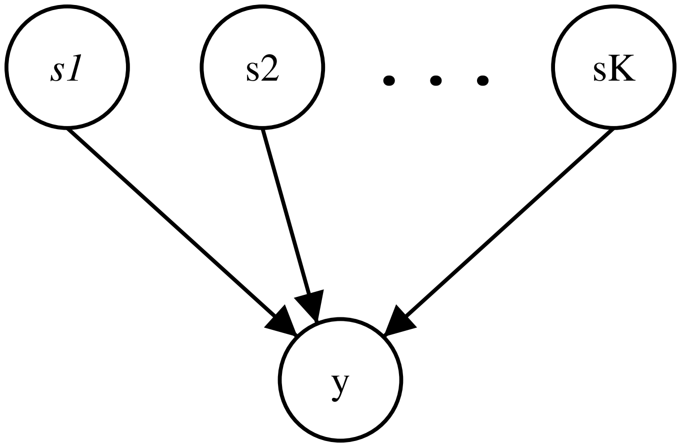
$$p(\mathbf{y}|s_1, \ldots, s_K, \boldsymbol{\mu}, \sigma^2) = \mathcal{N}\left(\sum_{i=1}^K s_i \boldsymbol{\mu}_i, \sigma^2 I\right)$$

EM optimizes lower bound on likelihood: $\mathcal{F}(q, \boldsymbol{\theta}) = \langle \log p(\mathbf{s}, \mathbf{y}|\boldsymbol{\theta})\rangle_{q(\mathbf{s})} - \langle \log q(\mathbf{s})\rangle_{q(\mathbf{s})}$

where $\langle\rangle_q$ is expectation under q: $\langle f(\mathbf{s})\rangle_q \stackrel{\text{def}}{=} \sum_{\mathbf{s}} f(\mathbf{s}) q(\mathbf{s})$

Exact E step: $q(\mathbf{s}) = p(\mathbf{s}|\mathbf{y}, \boldsymbol{\theta})$ is a distribution over 2^K states: **intractable** for large K

Example: Binary latent factor model



Model with K binary latent variables, $s_i \in \{0, 1\}$,
organised into a vector $\mathbf{s} = (s_1, \ldots, s_K)$
real-valued observation vector $\mathbf{y}$
parameters $\boldsymbol{\theta} = \{\{\boldsymbol{\mu}_i, \pi_i\}_{i=1}^{K}, \sigma^2\}$

$\mathbf{s} \sim$ Bernoulli
$\mathbf{y}|\mathbf{s} \sim$ Gaussian

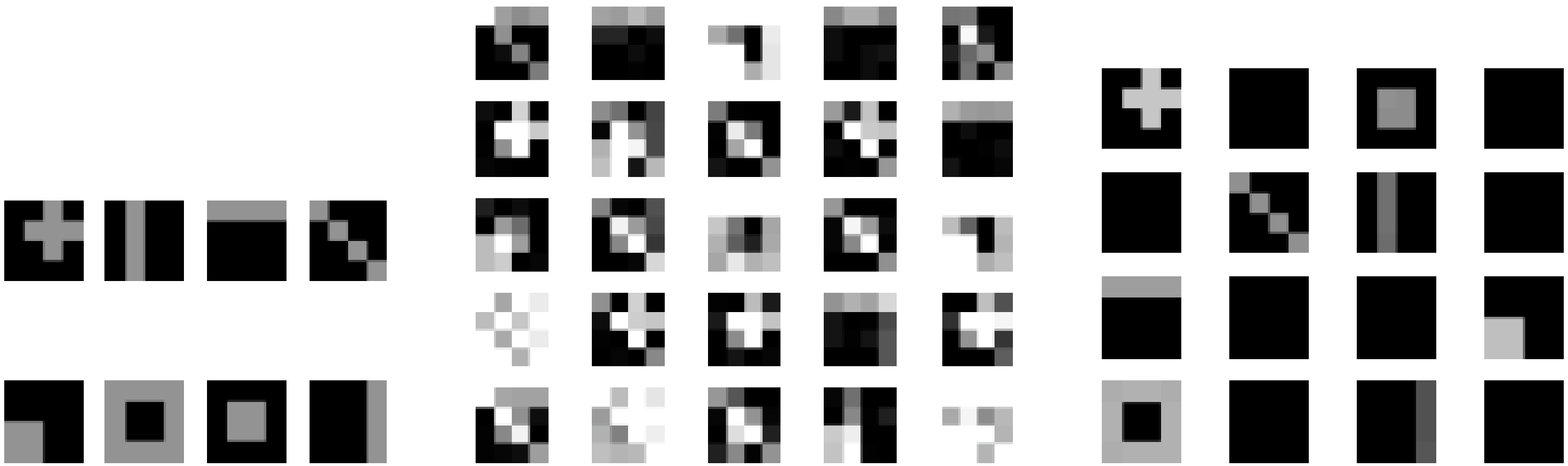


Figure 2: **Left panel**: Original source images used to generate data. **Middle panel**: Observed images resulting from mixture of sources. **Right panel**: Recovered sources

from Lu et al (2004)

Review: The EM algorithm

Given a set of observed (visible) variables V, a set of unobserved (hidden / latent / missing) variables H, and model parameters θ, optimize the log likelihood:

$$\mathcal{L}(\theta) = \log p(V|\theta) = \log \int p(H, V|\theta) dH,$$

Using Jensen's inequality, for any distribution of hidden variables $q(H)$ we have:

$$\mathcal{L}(\theta) = \log \int q(H) \frac{p(H, V|\theta)}{q(H)} \, dH \geq \int q(H) \log \frac{p(H, V|\theta)}{q(H)} \, dH = \mathcal{F}(q, \theta),$$

defining the $\mathcal{F}(q, \theta)$ functional, which is a lower bound on the log likelihood.

In the EM algorithm, we alternately optimize $\mathcal{F}(q, \theta)$ wrt q and θ, and we can prove that this will never decrease $\mathcal{L}$.

The E and M steps of EM

The lower bound on the log likelihood:

$$\mathcal{F}(q,\theta) = \int q(H) \log \frac{p(H,V|\theta)}{q(H)} dH = \int q(H) \log p(H,V|\theta) dH + \mathcal{H}(q),$$

where $\mathcal{H}(q) = -\int q(H) \log q(H) dH$ is the entropy of q. We iteratively alternate:

E step: maximize $\mathcal{F}(q,\theta)$ wrt the distribution over hidden variables given the parameters:

$$q^{[k]}(H) := \underset{q(H)}{\operatorname{argmax}} \ \mathcal{F}\big(q(H), \theta^{[k-1]}\big) = p(H|V, \theta^{[k-1]}).$$

M step: maximize $\mathcal{F}(q,\theta)$ wrt the parameters given the hidden distribution:

$$\theta^{[k]} := \underset{\theta}{\operatorname{argmax}} \ \mathcal{F}\big(q^{[k]}(H), \theta\big) = \underset{\theta}{\operatorname{argmax}} \int q^{[k]}(H) \log p(H,V|\theta) dH,$$

which is equivalent to optimizing the expected complete-data log likelihood $\log p(H,V|\theta)$, since the entropy of $q(H)$ does not depend on θ.

Variational Approximations to the EM algorithm

Often $p(H|V,\theta)$ is computationally intractable, so an exact E step is out of the question.

Assume some simpler form for $q(H)$, e.g. $q \in \mathcal{Q}$, the set of fully-factorized distributions over the hidden variables: $q(H) = \prod_i q(H_i)$

E step (approximate): maximize $\mathcal{F}(q,\theta)$ wrt the distribution over hidden variables given the parameters:

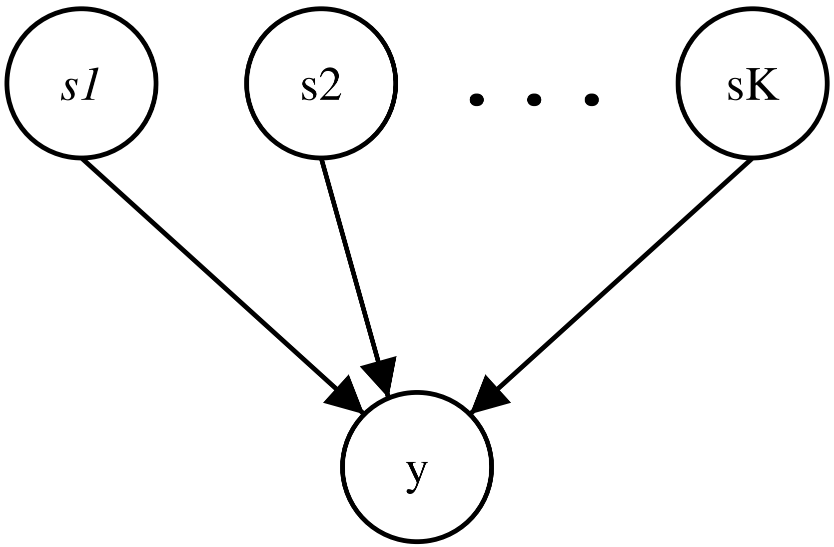
$$q^{[k]}(H) := \underset{q(H)\in\mathcal{Q}}{\operatorname{argmax}} \; \mathcal{F}\big(q(H), \theta^{[k-1]}\big).$$

M step : maximize $\mathcal{F}(q,\theta)$ wrt the parameters given the hidden distribution:

$$\theta^{[k]} := \underset{\theta}{\operatorname{argmax}} \; \mathcal{F}\big(q^{[k]}(H), \theta\big) = \underset{\theta}{\operatorname{argmax}} \; \int q^{[k]}(H) \log p(H, V|\theta) dH,$$

This maximizes a lower bound on the log likelihood.
Using the fully-factorized form of q is sometimes called a **mean-field approximation**.

Binary latent factor model



Model with K binary latent variables, $s_i \in \{0, 1\}$,
organised into a vector $\mathbf{s} = (s_1, \dots, s_K)$
real-valued observation vector $\mathbf{y}$
parameters $\boldsymbol{\theta} = \{\{\boldsymbol{\mu}_i, \pi_i\}_{i=1}^{K}, \sigma^2\}$

$\mathbf{s} \sim$ Bernoulli
$\mathbf{y}|\mathbf{s} \sim$ Gaussian

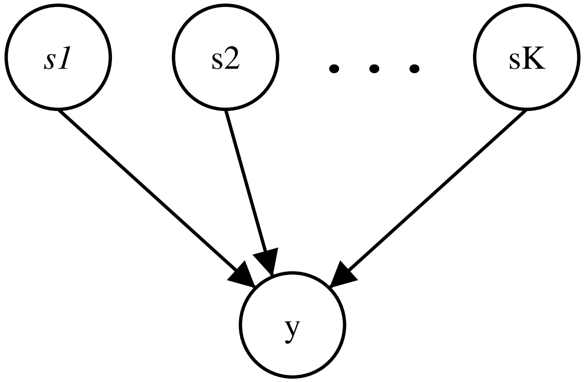
$$p(\mathbf{s}|\boldsymbol{\pi}) = p(s_1, \dots, s_K|\boldsymbol{\pi}) = \prod_{i=1}^{K} p(s_i|\pi_i) = \prod_{i=1}^{K} \pi_i^{s_i}(1-\pi_i)^{(1-s_i)}$$

$$p(\mathbf{y}|s_1, \dots, s_K, \boldsymbol{\mu}, \sigma^2) = \mathcal{N}\left(\sum_{i=1}^{K} s_i \boldsymbol{\mu}_i, \sigma^2 I\right)$$

EM optimizes lower bound on likelihood: $\mathcal{F}(q, \boldsymbol{\theta}) = \langle \log p(\mathbf{s}, \mathbf{y}|\boldsymbol{\theta}) \rangle_{q(\mathbf{s})} - \langle \log q(\mathbf{s}) \rangle_{q(\mathbf{s})}$
where $\langle \rangle_q$ is expectation under q: $\langle f(\mathbf{s}) \rangle_q \overset{\text{def}}{=} \sum_{\mathbf{s}} f(\mathbf{s}) q(\mathbf{s})$

Exact E step: $q(\mathbf{s}) = p(\mathbf{s}|\mathbf{y}, \boldsymbol{\theta})$ is a distribution over 2^K states: **intractable** for large K

Example: Binary latent factors model (cont)



$$\mathcal{F}(q, \boldsymbol{\theta}) = \langle \log p(\mathbf{s}, \mathbf{y}|\boldsymbol{\theta}) \rangle_{q(\mathbf{s})} - \langle \log q(\mathbf{s}) \rangle_{q(\mathbf{s})}$$

$$\begin{aligned}
\log \quad & p(\mathbf{s}, \mathbf{y}|\boldsymbol{\theta}) + c \\
= \quad & \textstyle\sum_{i=1}^{K} s_i \log \pi_i \quad + (1 - s_i) \log(1 - \pi_i) - D \log \sigma - \frac{1}{2\sigma^2} (\mathbf{y} - \sum_i s_i \boldsymbol{\mu}_i)^\top (\mathbf{y} - \sum_i s_i \boldsymbol{\mu}_i) \\
= \quad & \textstyle\sum_{i=1}^{K} s_i \log \pi_i \quad + (1 - s_i) \log(1 - \pi_i) - D \log \sigma \\
& \qquad - \frac{1}{2\sigma^2} \left(\mathbf{y}^\top \mathbf{y} - 2 \sum_i s_i \boldsymbol{\mu}_i^\top \mathbf{y} + \sum_i \sum_j s_i s_j \boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j \right)
\end{aligned}$$

we therefore need $\langle s_i \rangle$ and $\langle s_i s_j \rangle$ to compute $\mathcal{F}$.

These are the expected *sufficient statistics* of the hidden variables.

Example: Binary latent factors model (cont)

Variational approximation:

$$q(\mathbf{s}) = \prod_i q_i(s_i) = \prod_{i=1}^{K} \lambda_i^{s_i}(1-\lambda_i)^{(1-s_i)}$$

where λ_i is a parameter of the variational approximation modelling the posterior mean of s_i (compare to π_i which models the *prior* mean of s_i).

Under this approximation we know $\langle s_i \rangle = \lambda_i$ and $\langle s_i s_j \rangle = \lambda_i \lambda_j + \delta_{ij}(\lambda_i - \lambda_i^2)$.

$$\begin{aligned}\mathcal{F}(\boldsymbol{\lambda}, \boldsymbol{\theta}) = &\sum_i \lambda_i \log \frac{\pi_i}{\lambda_i} + (1-\lambda_i)\log\frac{(1-\pi_i)}{(1-\lambda_i)} \\ &- D\log\sigma - \frac{1}{2\sigma^2}(\mathbf{y} - \sum_i \lambda_i \boldsymbol{\mu}_i)^\top(\mathbf{y} - \sum_i \lambda_i \boldsymbol{\mu}_i) \\ &- \frac{1}{2\sigma^2}\sum_i (\lambda_i - \lambda_i^2){\boldsymbol{\mu}_i}^\top \boldsymbol{\mu}_i - \frac{D}{2}\log(2\pi)\end{aligned}$$

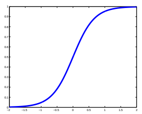
Fixed point equations for the binary latent factors model

Taking derivatives w.r.t. λ_i:

$$\frac{\partial \mathcal{F}}{\partial \lambda_i} = \log \frac{\pi_i}{1-\pi_i} - \log \frac{\lambda_i}{1-\lambda_i} + \frac{1}{\sigma^2}(\mathbf{y} - \sum_{j \neq i} \lambda_j \boldsymbol{\mu}_j)^\top \boldsymbol{\mu}_i - \frac{1}{2\sigma^2} \boldsymbol{\mu}_i{}^\top \boldsymbol{\mu}_i$$

Setting to zero we get fixed point equations:

$$\lambda_i = f\left(\log \frac{\pi_i}{1-\pi_i} + \frac{1}{\sigma^2}(\mathbf{y} - \sum_{j \neq i} \lambda_j \boldsymbol{\mu}_j)^\top \boldsymbol{\mu}_i - \frac{1}{2\sigma^2} \boldsymbol{\mu}_i{}^\top \boldsymbol{\mu}_i\right)$$

where $f(x) = 1/(1+\exp(-x))$ is the logistic (sigmoid) function. 

Learning algorithm:

E step: run fixed point equations until convergence of $\boldsymbol{\lambda}$ *for each data point*.
M step: re-estimate $\boldsymbol{\theta}$ given $\boldsymbol{\lambda}$s.

KL divergence

Note that

E step maximize $\mathcal{F}(q, \theta)$ wrt the distribution over hidden variables, given the parameters:

$$q^{[k]}(H) := \underset{q(H) \in \mathcal{Q}}{\operatorname{argmax}} \; \mathcal{F}\big(q(H), \theta^{[k-1]}\big).$$

is equivalent to:

E step minimize $\mathcal{KL}(q \| p(H|V, \theta))$ wrt the distribution over hidden variables, given the parameters:

$$q^{[k]}(H) := \underset{q(H) \in \mathcal{Q}}{\operatorname{argmin}} \int q(H) \log \frac{q(H)}{p(H|V, \theta^{[k-1]})} dH$$

So, in each E step, the algorithm is trying to find the best approximation to p in $\mathcal{Q}$.

This is related to ideas in *information geometry*.

Variational Approximations to Bayesian Learning

$$\begin{aligned}\log p(V) &= \log \int\int p(V,H|\boldsymbol{\theta})p(\boldsymbol{\theta})\ dH\ d\boldsymbol{\theta} \\ &\geq \int\int q(H,\boldsymbol{\theta}) \log \frac{p(V,H,\boldsymbol{\theta})}{q(H,\boldsymbol{\theta})}\ dH\ d\boldsymbol{\theta}\end{aligned}$$

Constrain $q \in \mathcal{Q}$ s.t. $q(H,\boldsymbol{\theta}) = q(H)q(\boldsymbol{\theta})$.

This results in the **variational Bayesian EM algorithm**.

More about this later (when we study model selection).

Variational Approximations and Graphical Models I

Let $q(H) = \prod_i q_i(H_i)$.

Variational approximation maximises $\mathcal{F}$:

$$\mathcal{F}(q) = \int q(H) \log p(H,V) dH - \int q(H) \log q(H) dH$$

Focusing on one term, q_j, we can write this as:

$$\mathcal{F}(q_j) = \int q_j(H_j) \left\langle \log p(H,V) \right\rangle_{\sim q_j(H_j)} dH_j + \int q_j(H_j) \log q_j(H_j) dH_j + \text{const}$$

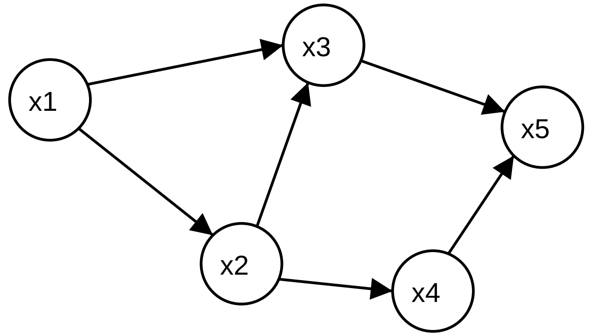
Where $\langle \cdot \rangle_{\sim q_j(H_j)}$ denotes averaging w.r.t. $q_i(H_i)$ for all $i \neq j$

Optimum occurs when:

$$q_j^*(H_j) = \frac{1}{Z} \exp \left\langle \log p(H,V) \right\rangle_{\sim q_j(H_j)}$$

Variational Approximations and Graphical Models II

Optimum occurs when:



$$q_j^*(H_j) = \frac{1}{Z} \exp \langle \log p(H, V) \rangle_{\sim q_j(H_j)}$$

Assume graphical model: $p(H, V) = \prod_i p(X_i | \mathsf{pa}_i)$

$$\begin{aligned} \log q_j^*(H_j) &= \left\langle \sum_i \log p(X_i | \mathsf{pa}_i) \right\rangle_{\sim q_j(H_j)} + \text{const} \\ &= \left\langle \log p(H_j | \mathsf{pa}_j) \right\rangle_{\sim q_j(H_j)} + \sum_{k \in \mathsf{ch}_j} \left\langle \log p(X_k | \mathsf{pa}_k) \right\rangle_{\sim q_j(H_j)} + \text{const} \end{aligned}$$

This defines messages that get passed between nodes in the graph. Each node receives messages from its Markov boundary: parents, children and parents of children.

Variational Message Passing (Winn and Bishop, 2004)

Expectation Propagation (EP)

Data (iid) $\mathcal{D} = \{\mathbf{x}^{(1)} \ldots, \mathbf{x}^{(N)}\}$, model $p(\mathbf{x}|\boldsymbol{\theta})$, with parameter prior $p(\boldsymbol{\theta})$.

The parameter posterior is:
$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{1}{p(\mathcal{D})} p(\boldsymbol{\theta}) \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta})$$

We can write this as product of factors over $\boldsymbol{\theta}$:
$$p(\boldsymbol{\theta}) \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) = \prod_{i=0}^{N} f_i(\boldsymbol{\theta})$$

where $f_0(\boldsymbol{\theta}) \stackrel{\text{def}}{=} p(\boldsymbol{\theta})$ and $f_i(\boldsymbol{\theta}) \stackrel{\text{def}}{=} p(\mathbf{x}^{(i)}|\boldsymbol{\theta})$ and we will ignore the constants.

We wish to approximate this by a product of *simpler* terms:
$$q(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \prod_{i=0}^{N} \tilde{f}_i(\boldsymbol{\theta})$$

$$\min_{q(\boldsymbol{\theta})} \mathrm{KL}\left(\prod_{i=0}^{N} f_i(\boldsymbol{\theta}) \middle\| \prod_{i=0}^{N} \tilde{f}_i(\boldsymbol{\theta}) \right) \qquad \text{(intractable)}$$

$$\min_{\tilde{f}_i(\boldsymbol{\theta})} \mathrm{KL}\left(f_i(\boldsymbol{\theta}) \| \tilde{f}_i(\boldsymbol{\theta}) \right) \qquad \text{(simple, non-iterative, inaccurate)}$$

$$\min_{\tilde{f}_i(\boldsymbol{\theta})} \mathrm{KL}\left(f_i(\boldsymbol{\theta}) \prod_{j \neq i} \tilde{f}_j(\boldsymbol{\theta}) \middle\| \tilde{f}_i(\boldsymbol{\theta}) \prod_{j \neq i} \tilde{f}_j(\boldsymbol{\theta}) \right) \qquad \text{(simple, iterative, accurate)} \leftarrow \text{EP}$$

Expectation Propagation II

Input $f_0(\boldsymbol{\theta}) \dots f_N(\boldsymbol{\theta})$
Initialize $\tilde{f}_0(\boldsymbol{\theta}) = f_0(\boldsymbol{\theta})$, $\tilde{f}_i(\boldsymbol{\theta}) = 1$ for $i > 0$, $q(\boldsymbol{\theta}) = \prod_i \tilde{f}_i(\boldsymbol{\theta})$
repeat
  for $i = 0 \dots N$ **do**
    Deletion: $q_{\setminus i}(\boldsymbol{\theta}) \leftarrow \dfrac{q(\boldsymbol{\theta})}{\tilde{f}_i(\boldsymbol{\theta})} = \prod_{j \neq i} \tilde{f}_j(\boldsymbol{\theta})$
    Projection: $\tilde{f}_i^{\text{new}}(\boldsymbol{\theta}) \leftarrow \arg\min_{f(\boldsymbol{\theta})} \mathrm{KL}(f_i(\boldsymbol{\theta}) q_{\setminus i}(\boldsymbol{\theta}) \| f(\boldsymbol{\theta}) q_{\setminus i}(\boldsymbol{\theta}))$
    Inclusion: $q(\boldsymbol{\theta}) \leftarrow \tilde{f}_i^{\text{new}}(\boldsymbol{\theta})\, q_{\setminus i}(\boldsymbol{\theta})$
  end for
until convergence

The EP algorithm. Some variations are possible: here we assumed that f_0 is in the exponential family, and we updated sequentially over i. The names for the steps (deletion, projection, inclusion) are not the same as in (Minka, 2001)

- Tries to minimize the opposite KL to variational methods
- $\tilde{f}_i(\boldsymbol{\theta})$ in exponential family $\rightarrow$ projection step is moment matching
- No convergence guarantee (although convergent forms can be developed)

Readings

- MacKay, D. (2003) Information Theory, Inference, and Learning Algorithms. Chapter 33.
- Bishop, C. (2006) Pattern Recognition and Machine Learning.
- Winn, J. and Bishop, C. (2005) Variational Message Passing. *J. Machine Learning Research*. `http://johnwinn.org/Publications/papers/VMP2005.pdf`
- Minka, T. (2004) Roadmap to EP: `http://research.microsoft.com/~minka/papers/ep/roadmap.html`
- Ghahramani, Z. (1995) Factorial learning and the EM algorithm. In Adv Neur Info Proc Syst 7. `http://learning.eng.cam.ac.uk/zoubin/zoubin/factorial.abstract.html`
- Jordan, M.I., Ghahramani, Z., Jaakkola, T.S. and Saul, L.K. (1999) An Introduction to Variational Methods for Graphical Models. Machine Learning 37:183-233. Available at: `http://learning.eng.cam.ac.uk/zoubin/papers/varintro.pdf`

Appendix: The binary latent factors model for an i.i.d. data set

Assume a data set $\mathcal{D} = \{\mathbf{y}^{(1)} \ldots, \mathbf{y}^{(N)}\}$ of N points. Parameters $\boldsymbol{\theta} = \{\{\boldsymbol{\mu}_i, \pi_i\}_{i=1}^K, \sigma^2\}$

Use a factorised distribution:

$$q(\mathbf{s}) = \prod_{n=1}^{N} q_n(\mathbf{s}^{(n)}) = \prod_{n=1}^{N} \prod_{i=1}^{K} q_n(s_i^{(n)}) = \prod_n \prod_i (\lambda_i^{(n)})^{s_i^{(n)}} (1 - \lambda_i^{(n)})^{(1-s_i^{(n)})}$$

$$\begin{aligned}
p(\mathcal{D}|\boldsymbol{\theta}) &= \prod_{n=1}^{N} p(\mathbf{y}^{(n)}|\boldsymbol{\theta}) \\
p(\mathbf{y}^{(n)}|\boldsymbol{\theta}) &= \sum_{\mathbf{s}} p(\mathbf{y}^{(n)}|\mathbf{s}, \boldsymbol{\mu}, \sigma) p(\mathbf{s}|\boldsymbol{\pi}) \\
\mathcal{F}(q(\mathbf{s}), \boldsymbol{\theta}) &= \sum_n \mathcal{F}_n(q_n(\mathbf{s}^{(n)}), \boldsymbol{\theta}) \leq \log p(\mathcal{D}|\boldsymbol{\theta}) \\
\mathcal{F}_n(q_n(\mathbf{s}^{(n)}), \boldsymbol{\theta}) &= \Big\langle \log p(\mathbf{s}^{(n)}, \mathbf{y}^{(n)}|\boldsymbol{\theta}) \Big\rangle_{q_n(\mathbf{s}^{(n)})} - \Big\langle \log q_n(\mathbf{s}^{(n)}) \Big\rangle_{q_n(\mathbf{s}^{(n)})}
\end{aligned}$$

We need to optimise w.r.t. the distribution over latent variables for *each data point*, so

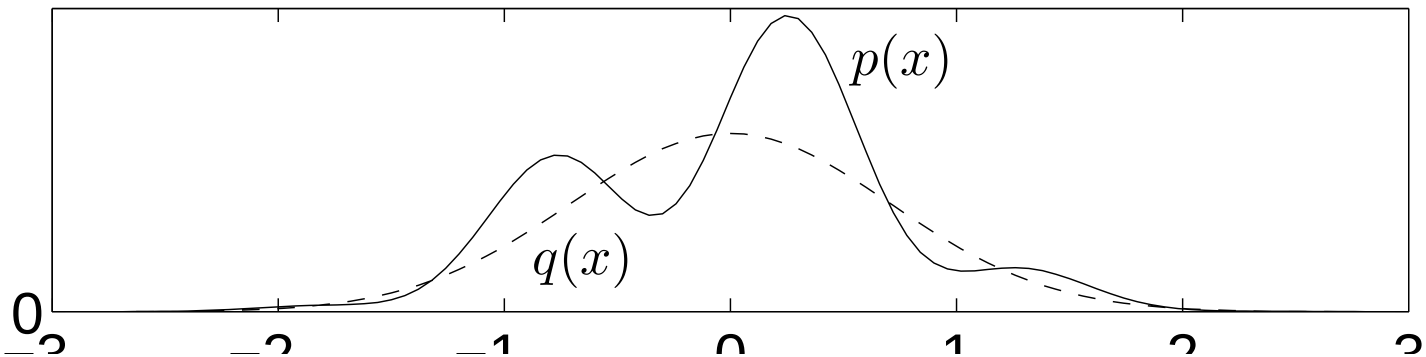
E step: optimize $q_n(\mathbf{s}^{(n)})$ (i.e. $\boldsymbol{\lambda}^{(n)}$) for each n.
M step: re-estimate $\boldsymbol{\theta}$ given $q_n(\mathbf{s}^{(n)})$'s.

Appendix: How tight is the lower bound?

It is hard to compute a nontrivial general upper bound.

To determine how tight the bound is, one can approximate the true likelihood by a variety of other methods.

One approach is to use the variational approximation as as a proposal distribution for **importance sampling**.



But this will generally not work well. See exercise 33.6 in David MacKay's textbook.