The Expectation-Propagation Algorithm: a tutorial

Simon Barthelmé (Gipsa-lab, CNRS)

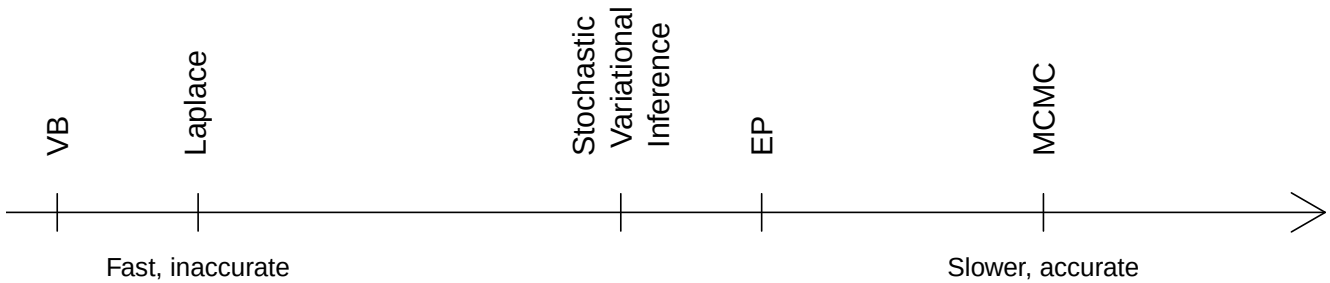
March 2, 2016

Outline

- Today: Basics of EP, application to GLMs and latent Gaussian models
- Tomorrow: Recent developments

Tradeoffs in variational inference

- Variational Inference methods are on an axis that goes from "fast and inaccurate" to "slower but more accurate"



Expectation Propagation

- EP was introduced by Tom Minka (2001).
- EP is known empirically to be very accurate in many cases
  - Gaussian processes
  - Logistic regression
- EP is very easy to parallelise (Barthelmé, Chopin, Cottet, 2015. Cseke & Heskes, 2011)
- EP is fast when implemented properly

Objective

We have a posterior distribution $\pi(\boldsymbol{\theta})$, we wish to approximate it with a Gaussian q such that

$$\underset{q \in \mathcal{Q}}{\operatorname{argmin}} KL\left(\pi || q\right)$$

$$\mathrm{KL}(\pi||q) = \int \pi(\boldsymbol{\theta}) \log \frac{\pi(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \, d\boldsymbol{\theta}$$

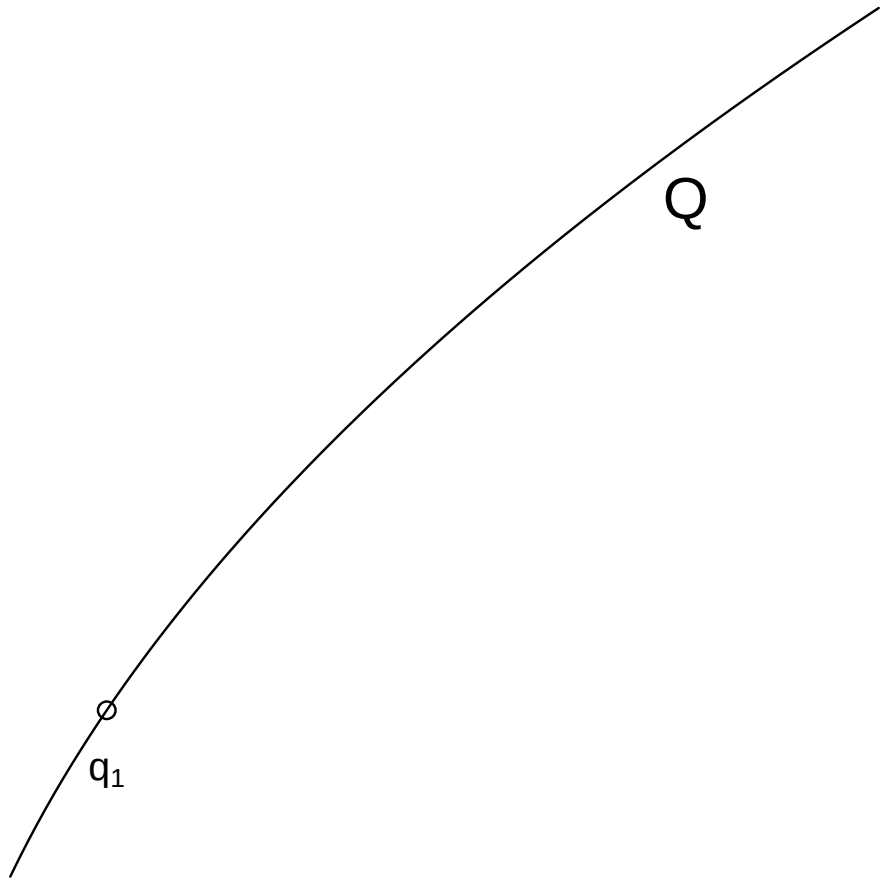
Properties of the KL objective

The solution of:

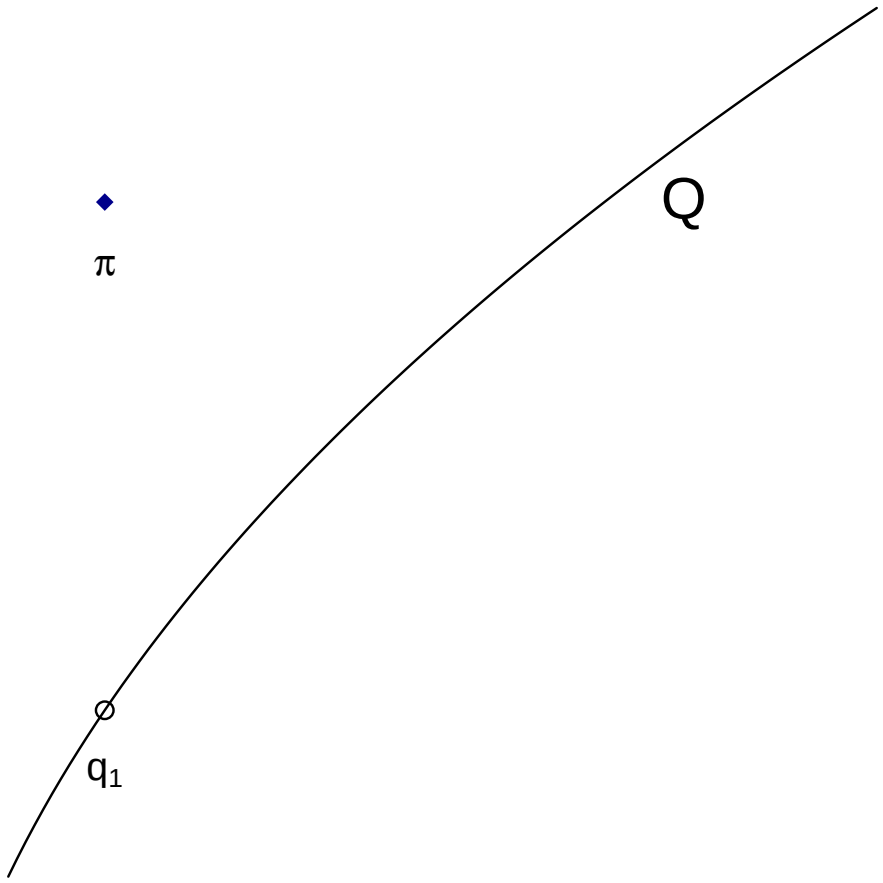
$$\underset{q \in \mathcal{Q}}{\operatorname{argmin}} KL\left(\pi || q\right)$$

has a "closed form" of sorts. It is the Gaussian $q^\star$ with mean $E(\pi)$ and variance $Var(\pi)$.
Obviously we have no hope of optimising the objective *exactly*.
In EP we will replace it with simpler, local problems we can actually solve.

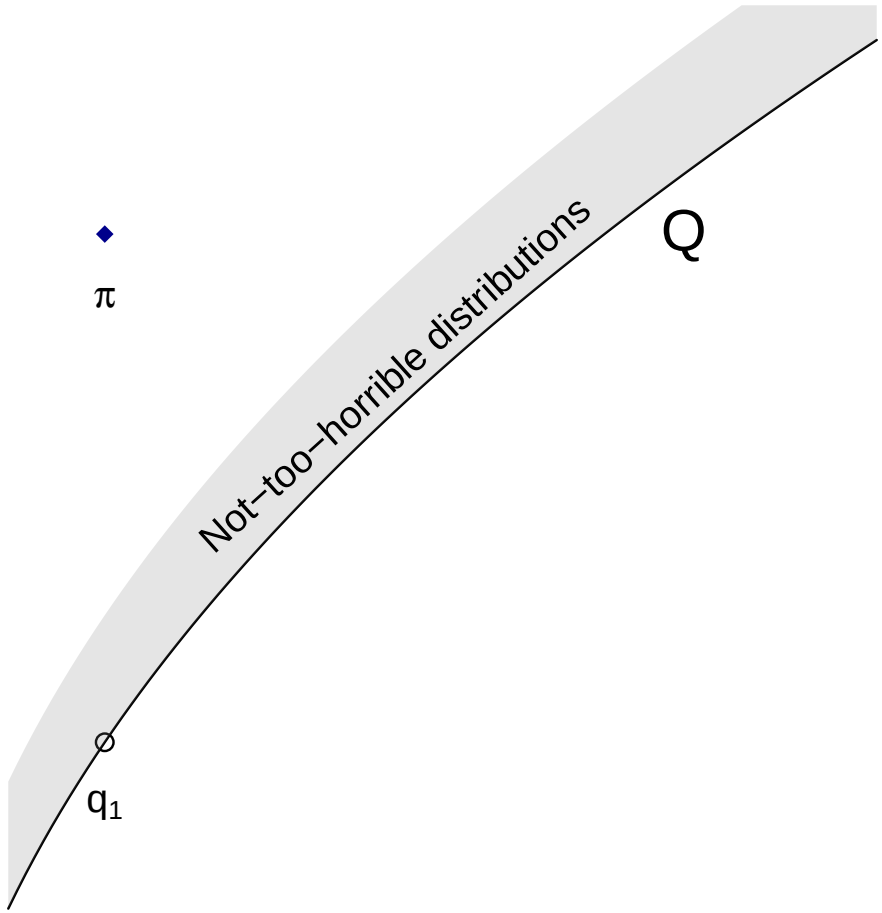
EP: the big picture



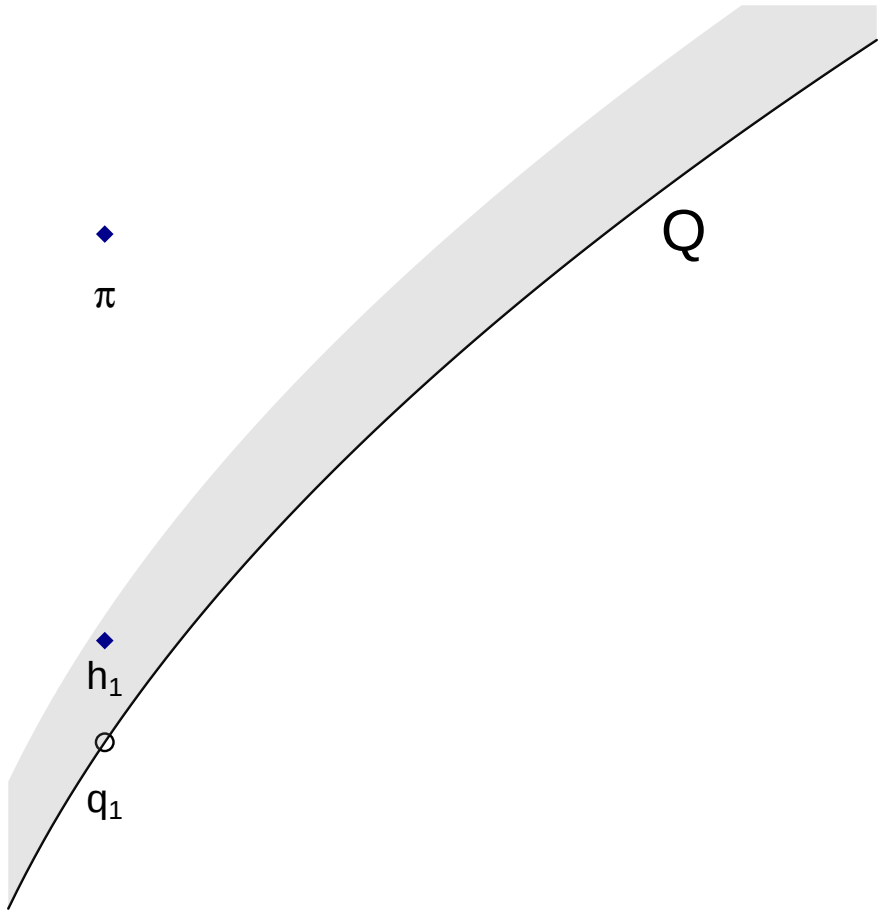
EP: the big picture



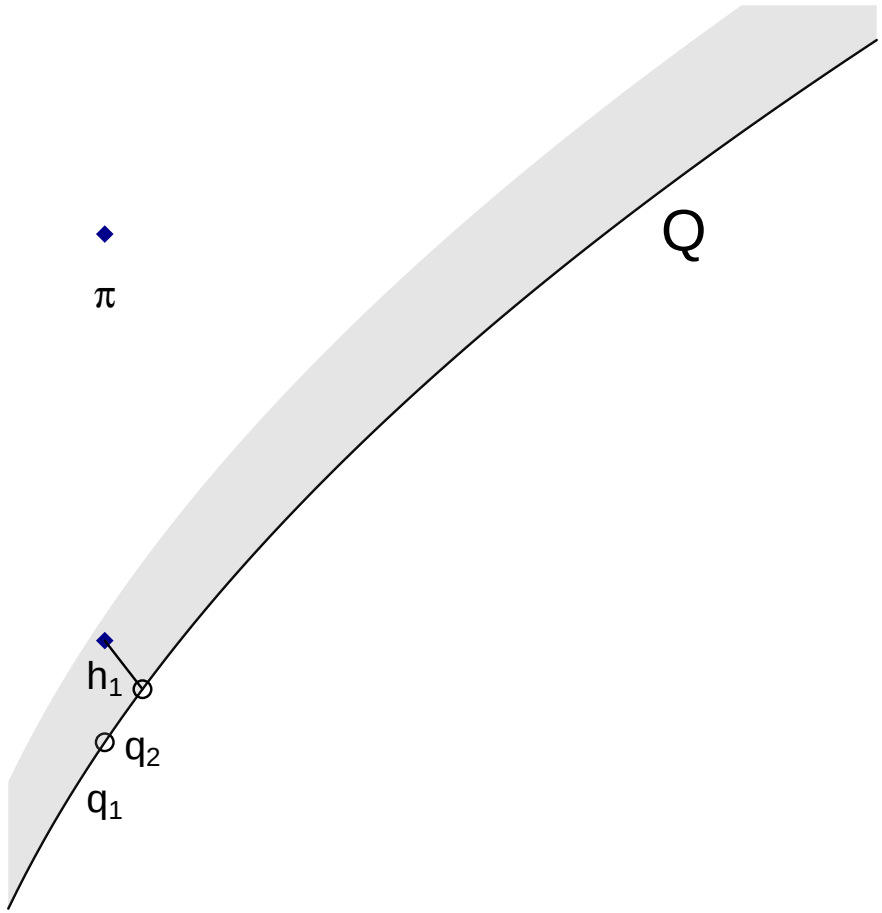
EP: the big picture



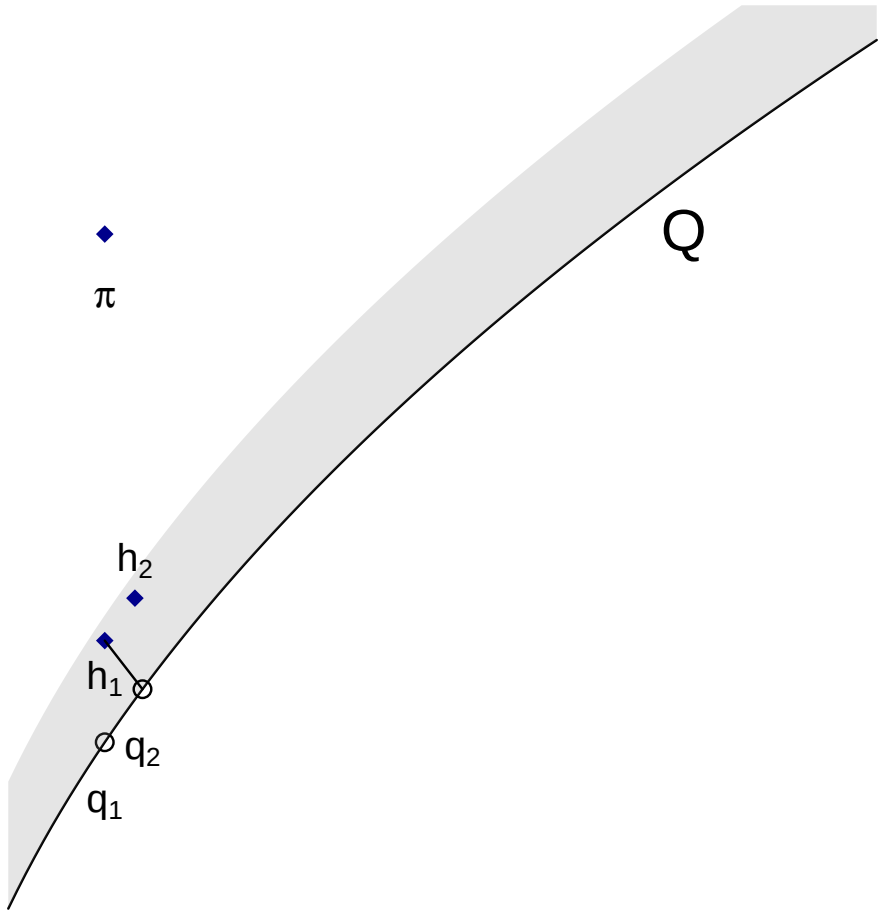
EP: the big picture



EP: the big picture



EP: the big picture



How EP works (I): write the posterior as a product

Consider a posterior distribution with independent datapoints:

$$\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y}) \propto p\left(\boldsymbol{\theta}\right) \prod_{i=1}^{n} l_i\left(\boldsymbol{\theta}\right)$$

It can be written as a product of factors:

$$\pi(\boldsymbol{\theta}) \propto \prod_{i=0}^{n} l_i(\boldsymbol{\theta})$$

How EP works (II): take a product of Gaussians

We will approximate the posterior:

$$\pi(\boldsymbol{\theta}) \propto \prod_{i=0}^{n} l_i(\boldsymbol{\theta})$$

with a product of a Gaussian factors:

$$q(\boldsymbol{\theta}) \propto \prod_{i=0}^{n} q_i(\boldsymbol{\theta})$$

How EP works (II): take a product of Gaussians

Each Gaussian factor equals:

$$q_i(\boldsymbol{\theta}) = \exp(-\frac{1}{2}\boldsymbol{\theta}^t \mathbf{A}_i \boldsymbol{\theta} + \mathbf{r}_i^t \boldsymbol{\theta})$$

And so the approximation is a Gaussian too:

$$q(\boldsymbol{\theta}) = \prod_{i=0}^{n} q_i(\boldsymbol{\theta}) = \exp(-\frac{1}{2}\boldsymbol{\theta}^t \sum_i \{\mathbf{A}_i\}\boldsymbol{\theta} + \sum_i \{\mathbf{r}_i^t\}\boldsymbol{\theta})$$

How EP works (III): hybridise the true and approximate distribution

You can form a *hybrid* between the true and the approximate distribution by replacing one of the approximate factors with one of the true factors:

1. Take out the approximate factor

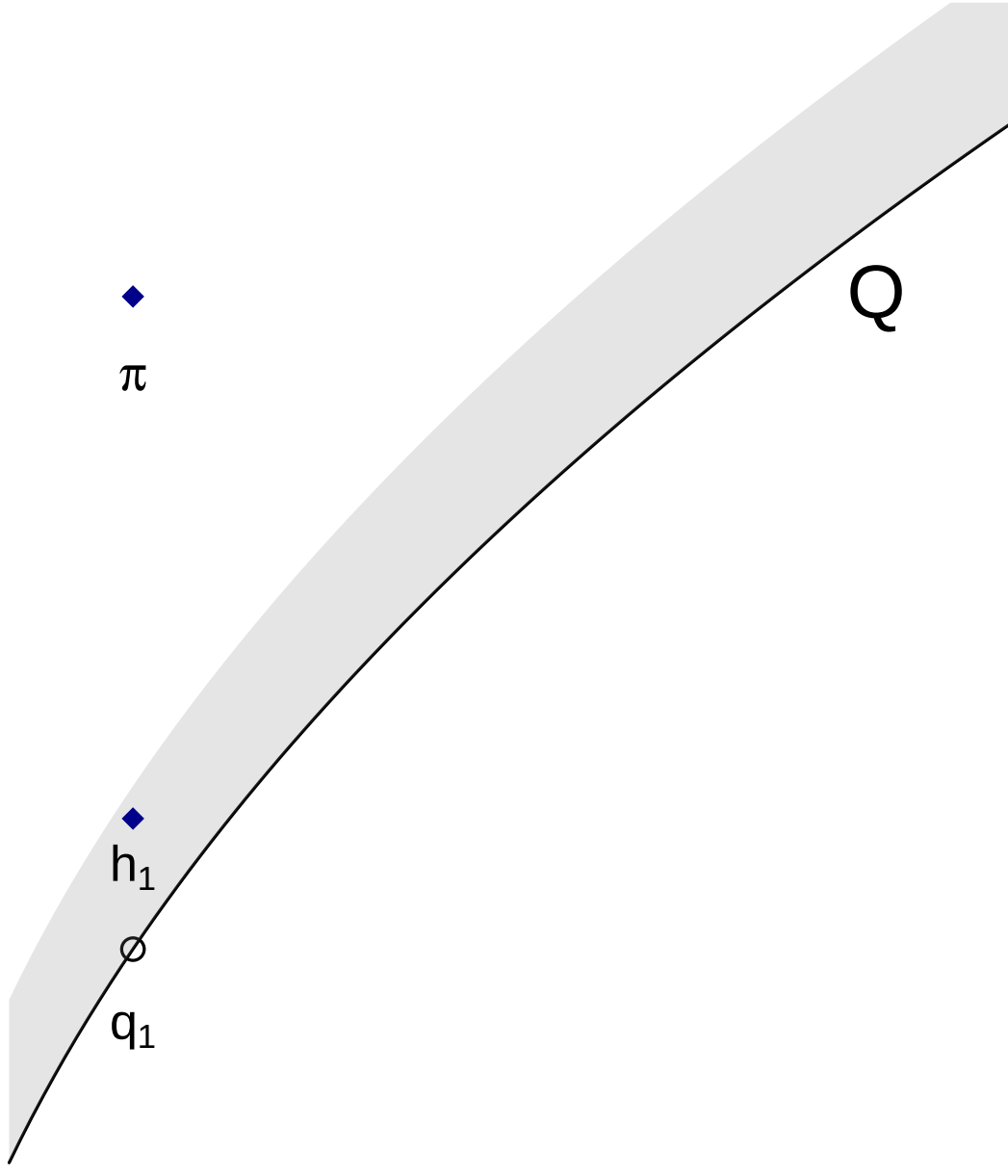
$$q_{-i}(\boldsymbol{\theta}) = \prod_{i \neq j}^{n} q_i(\boldsymbol{\theta})$$

2. Insert the true factor

$$h_i(\boldsymbol{\theta}) = l_i(\boldsymbol{\theta}) q_{-i}(\boldsymbol{\theta})$$

How EP works (III): hybridise the true and approximate distribution

Hopefully the hybrid is in some sense closer to the true distribution



How EP works (III): project the hybrid

That's just equivalent to computing the moments:

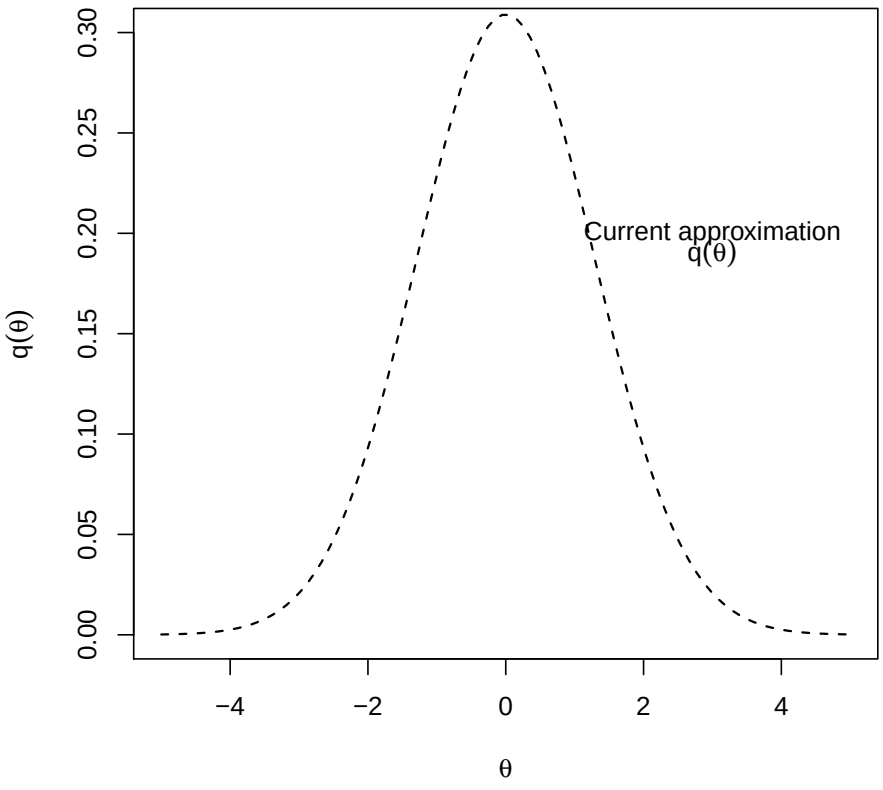
$$z = \int h_i(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

$$E(\boldsymbol{\theta}) = z^{-1} \int \boldsymbol{\theta} h_i(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

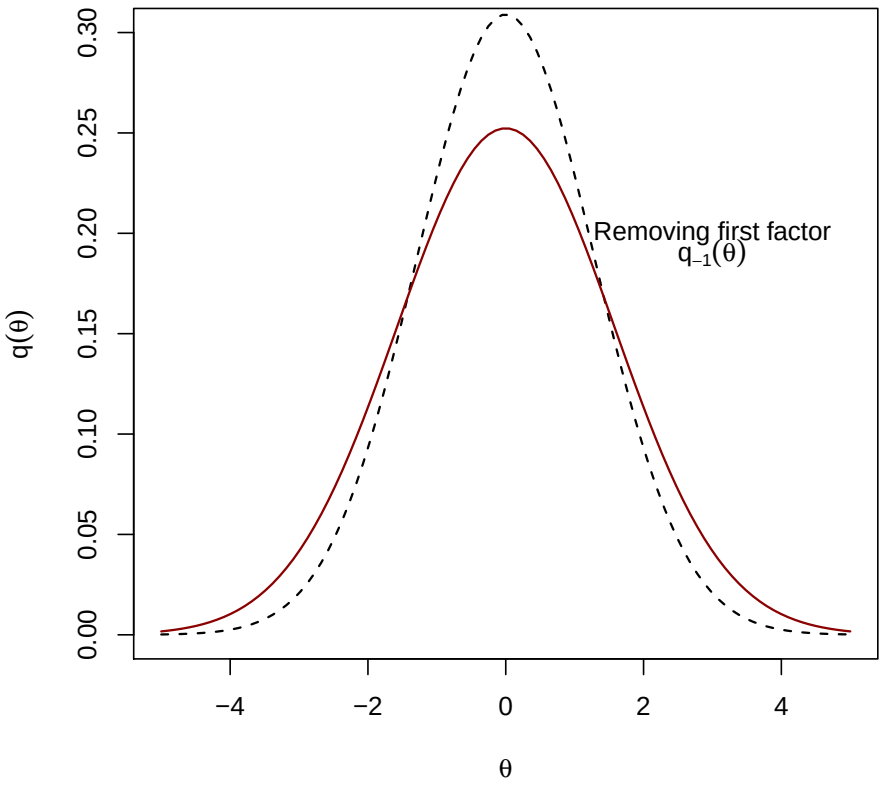
$$\Sigma = z^{-1} \int (\boldsymbol{\theta} - E(\boldsymbol{\theta}))(\boldsymbol{\theta} - E(\boldsymbol{\theta}))^t h_i(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

Our new global approximation q' is a Gaussian with mean and covariance as above.

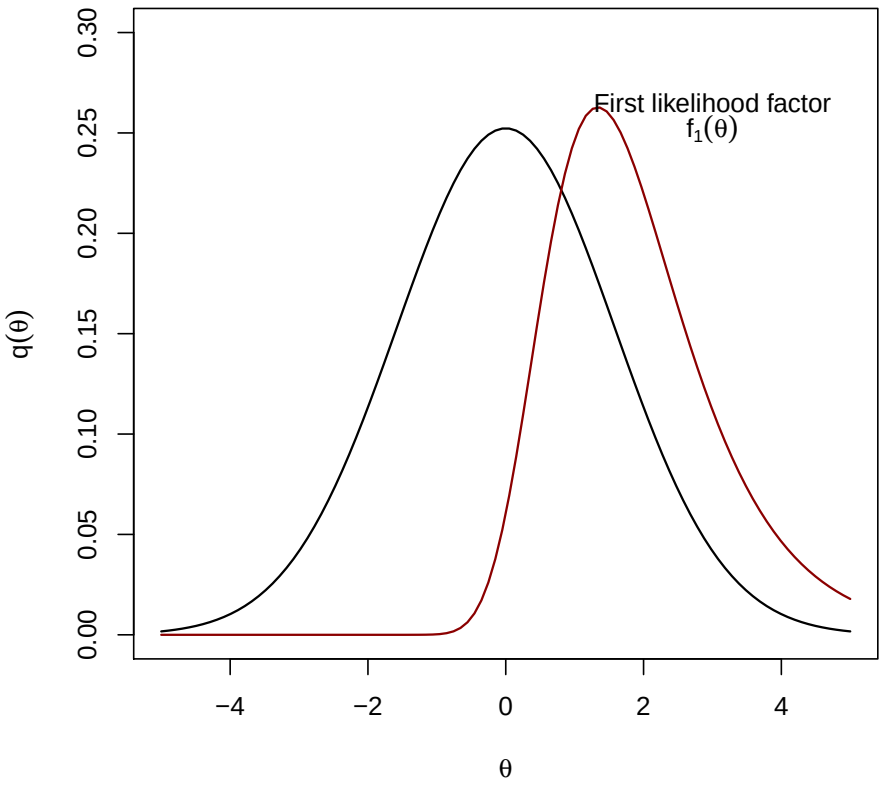
An illustration



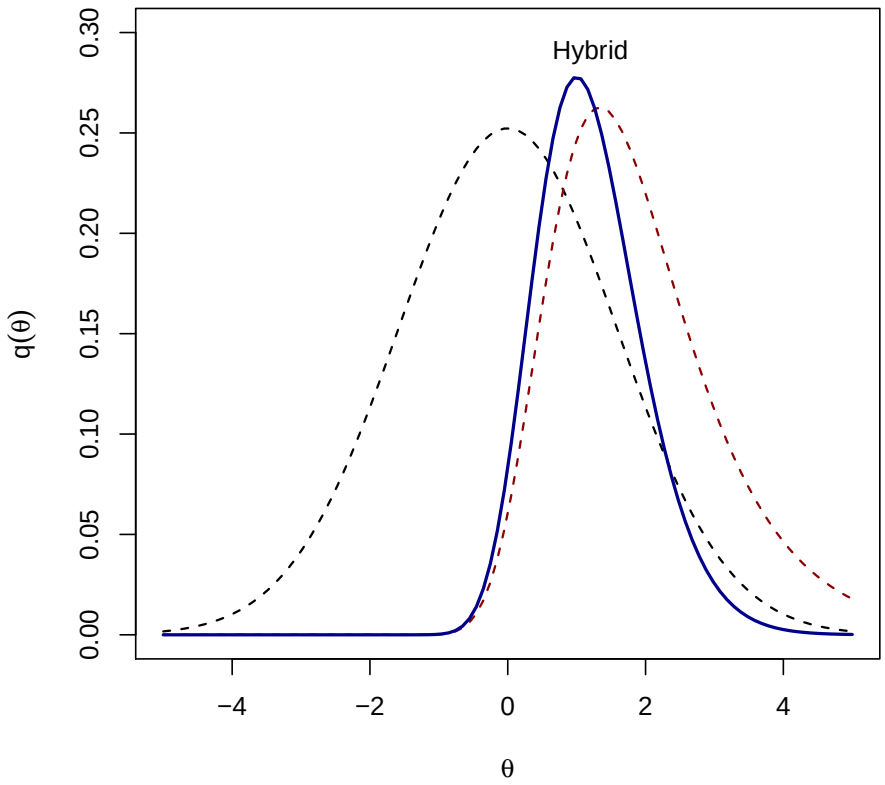
An illustration



An illustration



An illustration



How EP works (IV): update the approximate factor

- The last step is to update q_i, the Gaussian approximation of the factor we've just updated.
- Find the Gaussian q_i such that $q_i q_{-i}$ has the same moments as the hybrid.
- It's a simple linear operation in the natural parameters (more on that later)

EP for logistic regression

- So far we've stayed at a very abstract level
- Let's work through a concrete case: logistic regression
- Data $\mathbf{y} \in \{-1, 1\}^n$, covariates $\mathbf{X}_{n \times p}$, model:

$$p\left(y_i = 1 | \mathbf{x}_i, \boldsymbol{\theta}\right) = \phi\left(\mathbf{x}_i^t \boldsymbol{\theta}\right)$$

- ϕ is the logistic function.

Choosing a factorisation

- Assume Gaussian prior $p(\boldsymbol{\theta})$ (can be relaxed), our target distribution is:

$$\pi(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}) \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \boldsymbol{\theta}) = \prod_{i=0}^{n} l_i(\boldsymbol{\theta})$$

- Here $l_0(\boldsymbol{\theta})$ is the prior and each site corresponds to a data-point. This is the most traditional factorisation.

The hard bit: computing moments of the hybrid

- The hybrid is a product of a Gaussian times a single likelihood site, i.e.:

$$h_i(\boldsymbol{\theta}) \propto \phi\left(y_i \mathbf{x}_i^t \boldsymbol{\theta}\right) \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- We need the normalisation constant, mean and covariance of h_i
- That's a non-Gaussian distribution in $\mathbb{R}^p$, it's non-tractable! Are we back to using MCMC?
- Actually no: the linear subspace property comes to the rescue

The linear subspace property: logistic case

Start with the normalisation constant. We need to compute:

$$z = \int_{\mathbb{R}^p} \phi\left(y_i \mathbf{x}_i^t \boldsymbol{\theta}\right) \mathcal{N}\left(\boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \mathrm{d}\boldsymbol{\theta}$$

Express it as an expectation under the cavity prior:

$$z = E\left(\phi\left(y_i \mathbf{x}_i^t \boldsymbol{\theta}\right)\right), \boldsymbol{\theta} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

We have:

$$z = E\left(\phi\left(y_i \mathbf{x}_i^t \boldsymbol{\theta}\right)\right) = E\left(\phi\left(\mathbf{b}^t \boldsymbol{\theta}\right)\right) = E_u\left(\phi\left(u\right)\right)$$

where u is a one-dimensional Gaussian variable! We can just use quadrature.

The linear subspace property: mean and covariance

- We are able to compute z using a one-dimensional integral.
- The same goes for the mean and covariance but it's harder to see (and the formulas are more complicated)
- At the end of the day, all you need is the mean and variance of a one-dimensional marginal
- Proof:
  - Stein's lemma
  - Characteristic functions (more general)

The linear subspace property in general

If sites can be expressed as

$$l_i(\boldsymbol{\theta}) = g_i(\mathbf{B}_i\boldsymbol{\theta})$$

such that $\mathbf{B}_i\boldsymbol{\theta}$ has dimension $k < p$, then the hybrid moments can be computed from a *marginal* hybrid distribution of dimension k. In logistic regression (and GLMs) $k = 1$, meaning that all the moments can be computed using simple quadrature methods! The linear subspace property is central to the success of many EP methods.

Actual implementation

For those who can read R fluently: the expensive step is

```
compute.moments.logit <- function(y,m,v)
{
sd <- sqrt(v)
f <- function(x) dnorm(x,m,sd)*plogis(y*x)
z <- integrate(f,-Inf,Inf)$val
m.new <- integrate(function(x) f(x)*x,-Inf,Inf)$val/z
v.new <- integrate(function(x) f(x)*(x-m.new)^2,-Inf,Inf
list(m.new=m.new,var.new=v.new)
}
```

EP in exponential families

- There's a very elegant way of writing down EP iterations using exponential family notation, due to Matthias Seeger
- It's worth investing a few minutes setting it up
- Lets you generalise EP to other exponential families (not just Gaussians)

Gaussians as exponential families

- Rewrite

$$
\begin{aligned}
q(\boldsymbol{\theta}) &\propto \exp(-\frac{1}{2}\boldsymbol{\theta}^t \mathbf{A}\boldsymbol{\theta} + \mathbf{r}^t\boldsymbol{\theta}) \\
&= \exp\left(\sum \mathbf{A}_{ij}\left(\frac{-\theta_i\theta_j}{2}\right) + \sum \theta_i \mathbf{r}_i\right) \\
&= \exp\left(\mathbf{s}\left(\boldsymbol{\theta}\right)^t \boldsymbol{\lambda}\right)
\end{aligned}
$$

Natural and moment parameters

$$q(\boldsymbol{\theta}) \propto \exp(-\frac{1}{2}\boldsymbol{\theta}^t \mathbf{A}\boldsymbol{\theta} + \mathbf{r}^t\boldsymbol{\theta}) = \exp\left(\mathbf{s}\left(\boldsymbol{\theta}\right)^t \boldsymbol{\lambda}\right)$$

- $\boldsymbol{\lambda} = \{\mathbf{A}, \mathbf{r}\}$ (precision and shift) are the "natural" parameters of the Gaussian
- $\boldsymbol{\eta} = E_q(\mathbf{s}(\boldsymbol{\theta}))$ (mean and covariance) are the "moment" parameters of the Gaussian
- $\boldsymbol{\lambda} = \nu\left(\boldsymbol{\eta}\right)$: one-to-one transformation from natural to moment parameters

Benefits of rewriting: additivity

- Multiply sites => natural parameters add

$$\prod q_i(\theta) = \prod \exp\left(s(\theta)^t \lambda_i\right) = \exp\left(s(\theta)^t \sum \lambda_i\right)$$

- Take out a site => subtract

$$\frac{q(\theta)}{q_i(\theta)} = \exp\left(s(\theta)^t (\lambda - \lambda_i)\right)$$

EP in one slide

1. Initialise site parameters $\boldsymbol{\lambda}_1 \ldots \boldsymbol{\lambda}_n$. Global parameter: $\boldsymbol{\lambda} = \sum \boldsymbol{\lambda}_i$.
2. While not converged, loop over i:
   - 2.1 Form cavity: $\boldsymbol{\lambda}_{-i} = \boldsymbol{\lambda} - \boldsymbol{\lambda}_i$, hybrid $h_i(\boldsymbol{\theta}) \propto l_i(\boldsymbol{\theta}) \exp\left(s(\boldsymbol{\theta})^t \boldsymbol{\lambda}_{-i}\right)$
   - 2.2 Compute moments: $\boldsymbol{\eta}_i = E_{h_i}(s(\boldsymbol{\theta}))$, transform back to natural parameters $\boldsymbol{\lambda}_i = \nu(\boldsymbol{\eta}_i) - \boldsymbol{\lambda}_{-i}$
   - 2.3 Update global approximation: $\boldsymbol{\lambda} = \boldsymbol{\lambda}_{-i} + \boldsymbol{\lambda}_i$

Some remarks

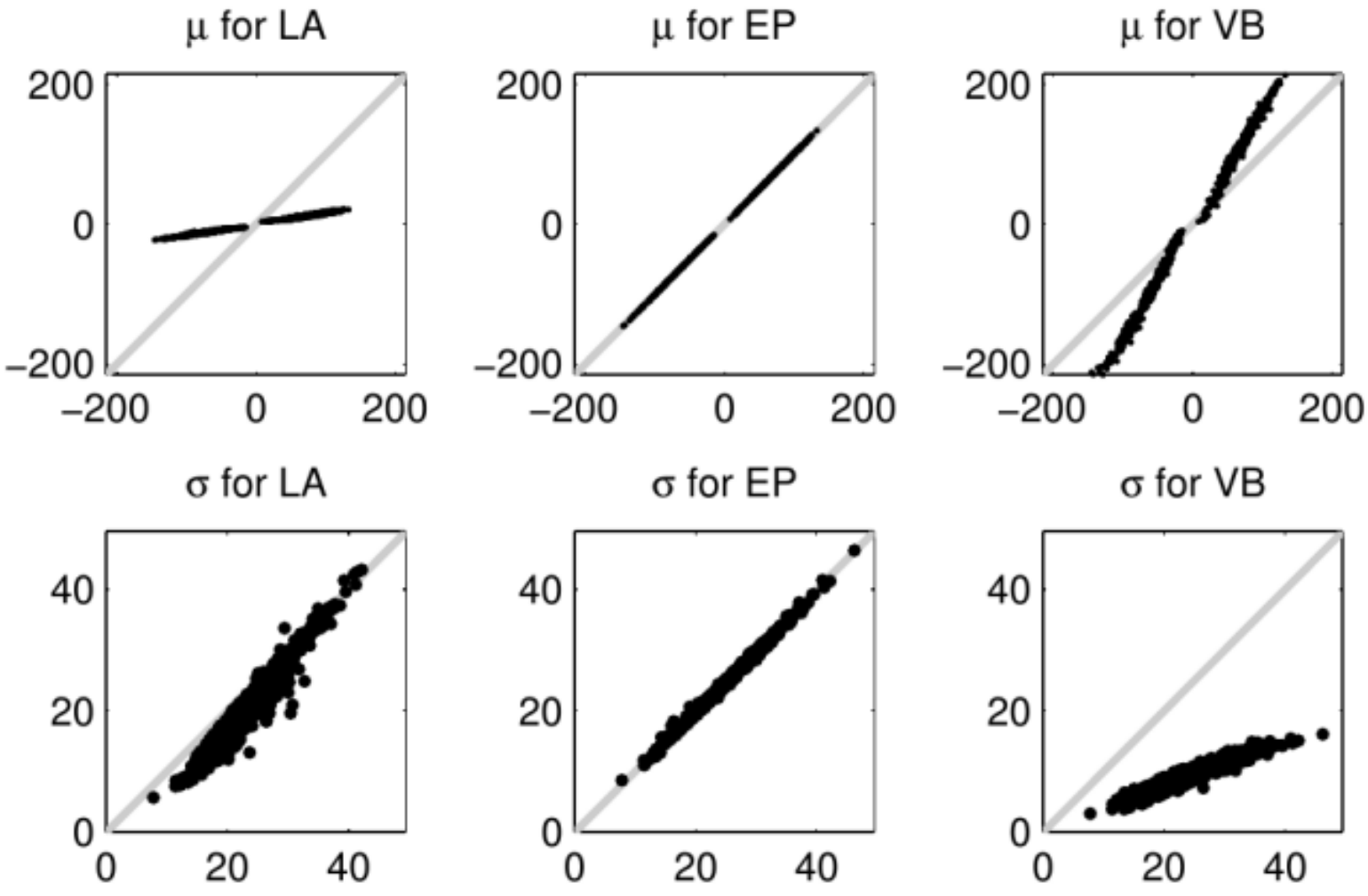
- Global approximation $q(\boldsymbol{\theta}) \propto \exp\left(s\left(\boldsymbol{\theta}\right)^t \boldsymbol{\lambda}\right)$, with$\boldsymbol{\lambda} = \sum \boldsymbol{\lambda}_i$
- For Gaussian family: $\boldsymbol{\lambda} = \{\mathbf{A}, \mathbf{r}\}$ (precision and shift)
  - Each site (each bit of the likelihood) contributes a little bit of precision and a little bit of shift to the whole approximation
  - Cavity: remove that contribution

Computational cost

- Two potentially expensive operations:
  1. Computing moments of the hybrid $\boldsymbol{\eta}_i = E_{h_i}\left(s\left(\boldsymbol{\theta}\right)\right)$ ($\mathcal{O}\left(n\right)$ per complete pass over the data)
  2. Transforming natural parameters to moment parameters involves matrix inverse ($\mathcal{O}\left(np^3\right)$ per complete pass in the general case)
- In typical problems Sequential EP stabilises in 3-4 passes regardless of n so $\mathcal{O}\left(n\right)$ scaling!

EP success stories

Gaussian process classification - essentially a big non-parametric probit model



Nickish & Rasmussen (2008)

EP success stories

- EP does very well in latent Gaussian models: GLMs with Gaussian priors (Cseke & Heskes 2011)
- Logistic regression with various priors (Ridway & Chopin 2015): non-Gaussian priors are approximated as just another set of sites
- Sparse models (Seeger, 2008; Jylänki et al. 2014).
- Many more - incomplete list at: http://research.microsoft.com/en-us/um/people/minka/papers/ep/roadmap.html
- Can also be used to speed up MCMC (Fillipone & Girolami 2015)

Potential difficulties with EP

- Only one known large-scale application of EP in industry: Microsoft's True Skill player rating system (designed by Tom Minka).
- Why don't people use it more?
  - It's hard to prove anything much about EP
  - Also: EP can work extremely well, but implementation requires care, especially in complex models.

Stability in EP: stable linear algebra

- EP is a fixed point algorithm: you iterate the updates until the parameters stabilise
- Occasionnally, especially on models that aren't log-concave, EP doesn't stabilise, it diverges
- Sometime, covariance matrices accumulate noise until one of the eigenvalues goes to 0
- Fix #1: use stable linear algebra operations (Cholesky decompositions, not explicit inverses). See Seeger (2008).

Stability in EP: slowing down

- Fix #2: If EP diverges try slowing down the iterations: instead of going to the full update $\boldsymbol{\lambda}'$, use

$$\boldsymbol{\lambda}_{t+1} = \alpha \boldsymbol{\lambda}_t + (1 - \alpha)\, \boldsymbol{\lambda}'$$

- See Dehaene & Barthelmé (2015) for why this helps
- Interesting open question: optimal rate α, optimal parameterisation

Stability in EP: Power EP

- Fix #3: Power-EP (Minka, 2004). Power EP is a form of likelihood tempering.
  - instead of having n full-strength sites, split them artificially
  - New factorisation:

$$p(\boldsymbol{\theta}) \propto \prod_k \prod_n l_i^{1/k}(\boldsymbol{\theta})$$

- Power-EP behaves better on "hard" models.
- Equivalent to minimising $KL(q||p)$ in the $k \to \infty$ limit

Implementations

- GPstuff toolbox (Vanhatalo et al. 2015, Matlab), latent Gaussian models with various likelihoods
- gpml and glm-ie (H. Nickish, Matlab) for latent Gaussian models and sparse GLMs.
- EPGLM (R, James Ridgway): logistic and probit models
- ABC-EP (Barthelmé & Chopin, Matlab): more on that tomorrow

Learning more about EP

- There are different perspectives on EP, each interesting:
  - EP as improved Assumed Density Filtering: Minka (2001)
  - Links to statistical physics: lecture notes by M. Opper (2015).
  - Links to other forms of variational inference: Wainwright & Jordan (); Minka (2005)
  - Exponential families: Seeger (2008)