Expectation propagation for infinite mixtures

Tom Minka and Zoubin Ghahramani

NIPS'03 Workshop on

Nonparametric Bayesian Methods
and Infinite Models

The problem

- Want compact summary of posterior on mixture parameters given data: $p(\theta \mid Data)$
- Define θ_i to be parameters of component which generated x_i
- Approximate the posterior for θ_i

$$
\begin{tikzcd}
\theta_1 \arrow[r] \arrow[rr, bend left] \arrow[d] & \theta_2 \arrow[r] \arrow[d] & \theta_3 \arrow[d] \\
x_1 & x_2 & x_3
\end{tikzcd}
$$

Infinite mixture model

- Dirichlet process prior:

$$p(\theta_i \mid \theta_{<i}) = \frac{\alpha}{i-1+\alpha}\, p(\theta_i) + \frac{1}{i-1+\alpha} \sum_{j<i} \delta(\theta_i - \theta_j)$$

α is the “innovation” parameter $\qquad p(\theta_i) \sim N(m_0, V_0)$

- Gaussian components with known variance:

$$p(x_i \mid \theta_i) \sim N(\theta_i, \Sigma)$$

$$\begin{tikzcd}
\theta_1 \arrow[r] \arrow[d] \arrow[rr, bend left] & \theta_2 \arrow[r] \arrow[d] & \theta_3 \arrow[d] \\
x_1 & x_2 & x_3
\end{tikzcd}$$

Expectation Propagation

- Approximate a function by a simpler one:

$$p(\mathbf{x}) = \prod_a f_a(\mathbf{x}) \longrightarrow q(\mathbf{x}) = \prod_a \tilde{f}_a(\mathbf{x})$$

- Where each $\tilde{f}_a(\mathbf{x})$ lives in tractable family
- Iterate the fixed-point equations:

$$\tilde{f}_a(\mathbf{x}) = \arg\min D(f_a(\mathbf{x}) q^{\backslash a}(\mathbf{x}) \,\|\, \tilde{f}_a(\mathbf{x}) q^{\backslash a}(\mathbf{x}))$$

where $$q^{\backslash a}(\mathbf{x}) = \prod_{b \neq a} \tilde{f}_b(\mathbf{x})$$

- Want to approximate

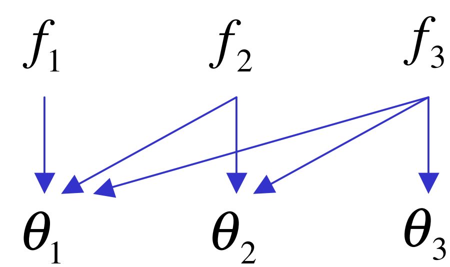
$$\prod_i p(x_i \mid \theta_i)\, p(\theta_i \mid \theta_{<i}) \approx \prod_i q(\theta_i)$$

$$q(\theta_i) \sim N(m_i, V_i)$$

- Likelihood terms are already Gaussian
- Prior terms are approximated by factorized Gaussians: $$p(\theta_i \mid \theta_{<i}) = f_i(\theta) \approx \tilde{f}_i(\theta) = \prod_{j \le i} \tilde{f}_{ij}(\theta_j)$$

- $\tilde{f}_{ij}$ are "messages"

$$\tilde{f}_{ij}(\theta_j) \sim N(m_{ij}, V_{ij})$$



EP algorithm

- Deletion: $q^{\setminus i}(\theta) = \dfrac{q(\theta)}{\tilde{f}_i(\theta)}$
- Inclusion: change $q(\theta)$ to match moments of
  $p(\theta_i \mid \theta_{<i})q^{\setminus i}(\theta)$
- Update:
  $$\tilde{f}_i(\theta) = \frac{q(\theta)}{q^{\setminus i}(\theta)} = \prod_{j \le i} \frac{q(\theta_j)}{q^{\setminus i}(\theta_j)}$$

Moment matching

$$m_i = \sum_{j \le i} r_{ji} E[\theta_i \mid \theta_i = \theta_j]$$

r_{ji} is probability that i picks j

$$r_{ji} \propto \frac{1}{i-1+\alpha} N(m_i^{\setminus i} - m_j^{\setminus i}, V_i^{\setminus i} + V_j^{\setminus i})$$

$$r_{ii} \propto \frac{\alpha}{i-1+\alpha} N(m_i^{\setminus i} - m_0, V_i^{\setminus i} + V_0) \qquad \text{(innovation)}$$

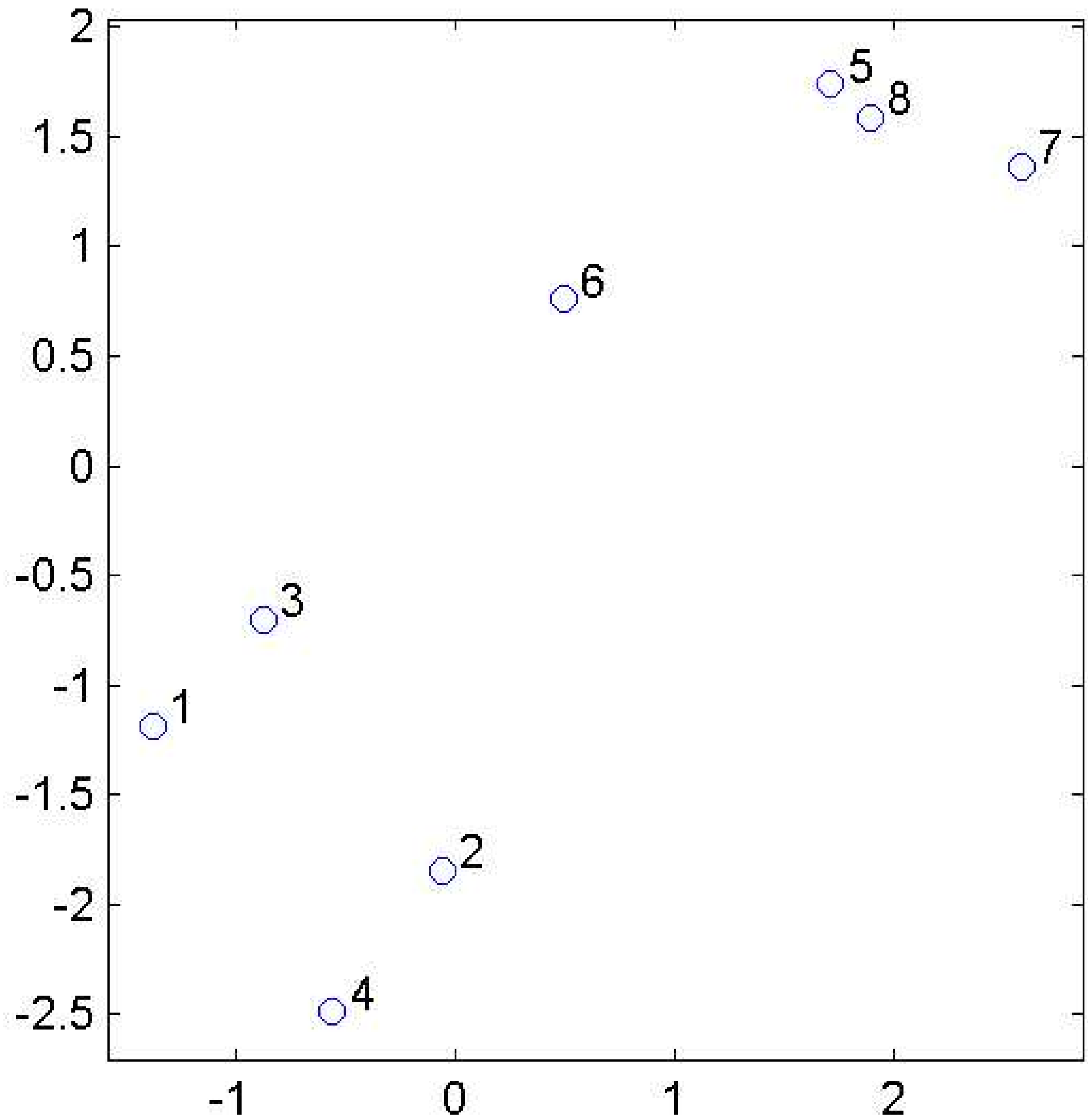
$$m_j = (1 - r_{ji}) m_j^{\setminus i} + r_{ji} E[\theta_j \mid \theta_i = \theta_j]$$

Usage

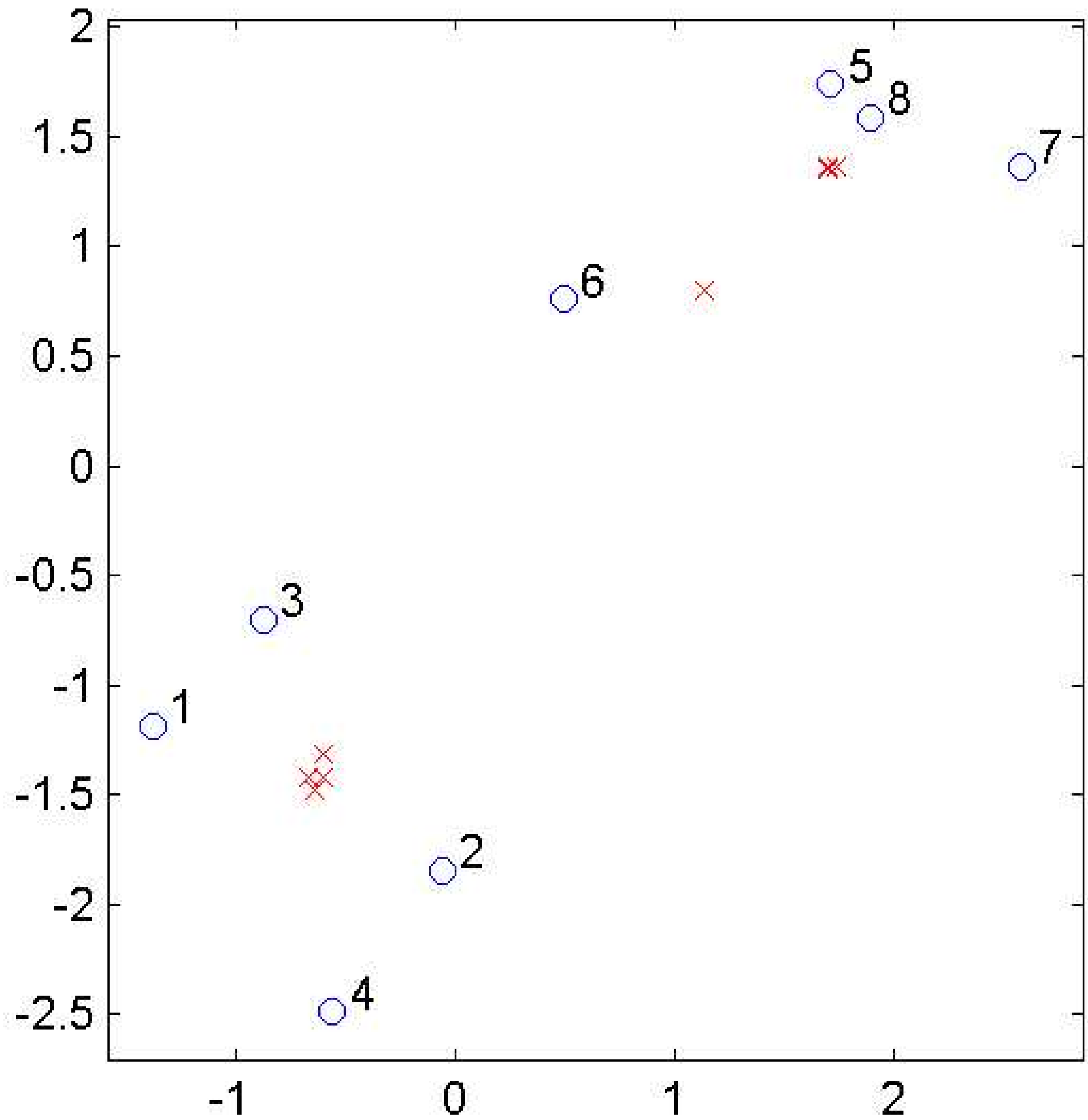
- Input is hyperparameters and data:

  $$(\alpha, \Sigma, m_0, V_0, x_1, \ldots, x_n)$$

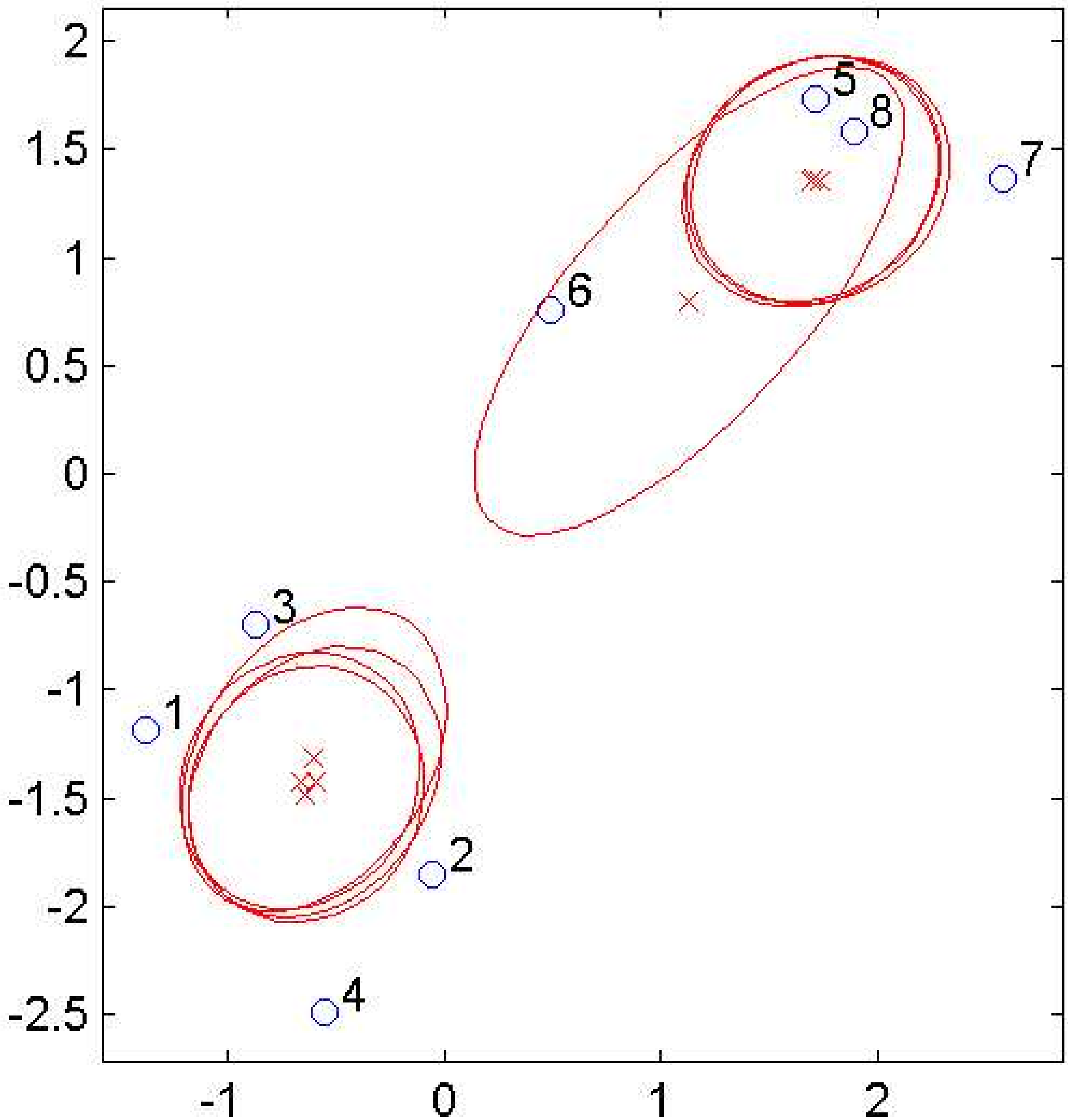
- Output is Gaussian posteriors and soft assignments: (m_i, V_i, r_{ji})
- Expected number of components: $\sum_i r_{ii}$
  - Prior expected number: $\alpha(\psi(\alpha + n) - \psi(\alpha))$
  - Set these equal to update α
  - Can be interleaved with EP iterations



2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-1
0
1
2
1
2
3
4
5
6
7
8



2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-1
0
1
2
1
2
3
4
5
6
7
8

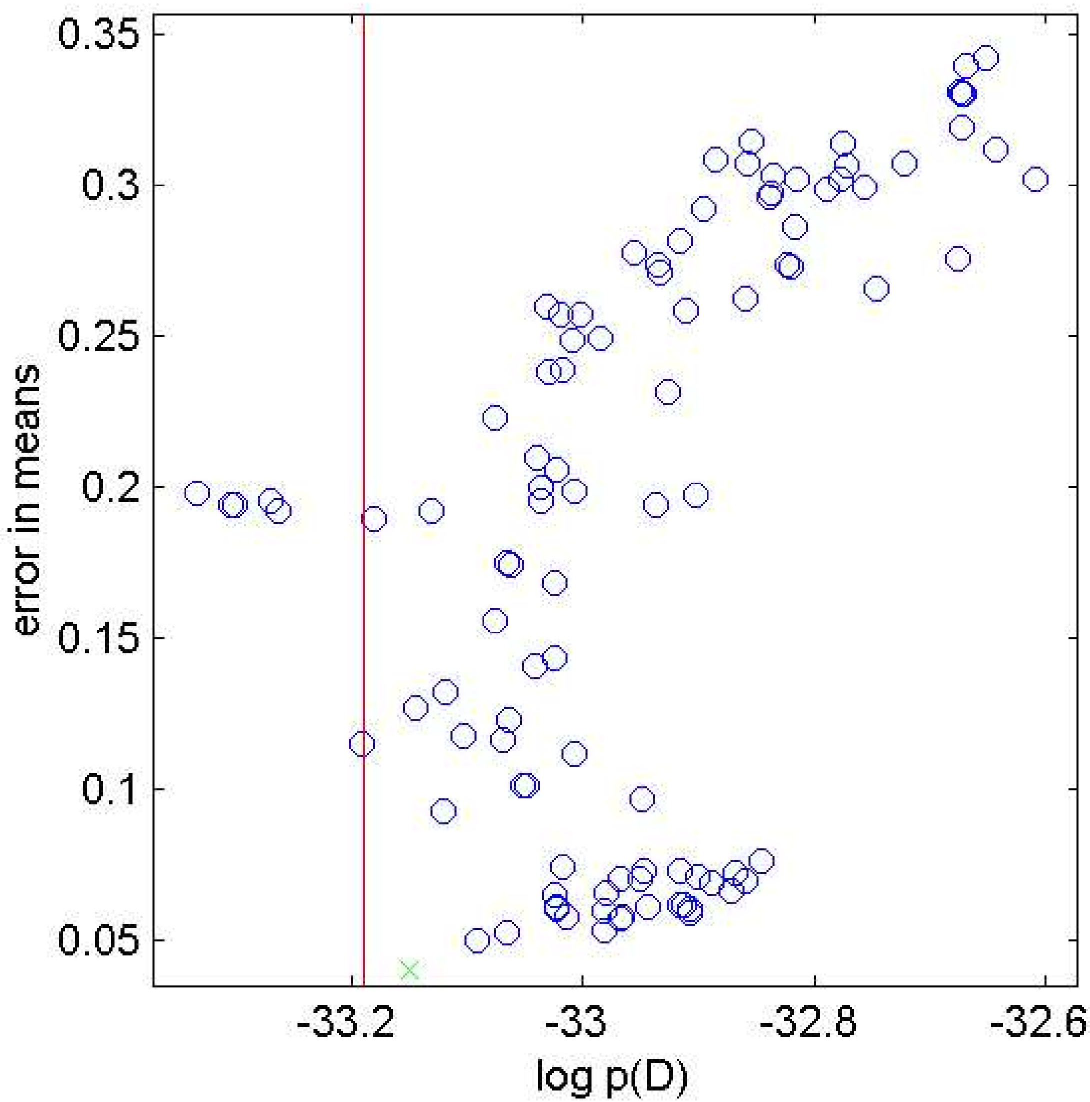


Alpha = 0.38
means 1.82
components

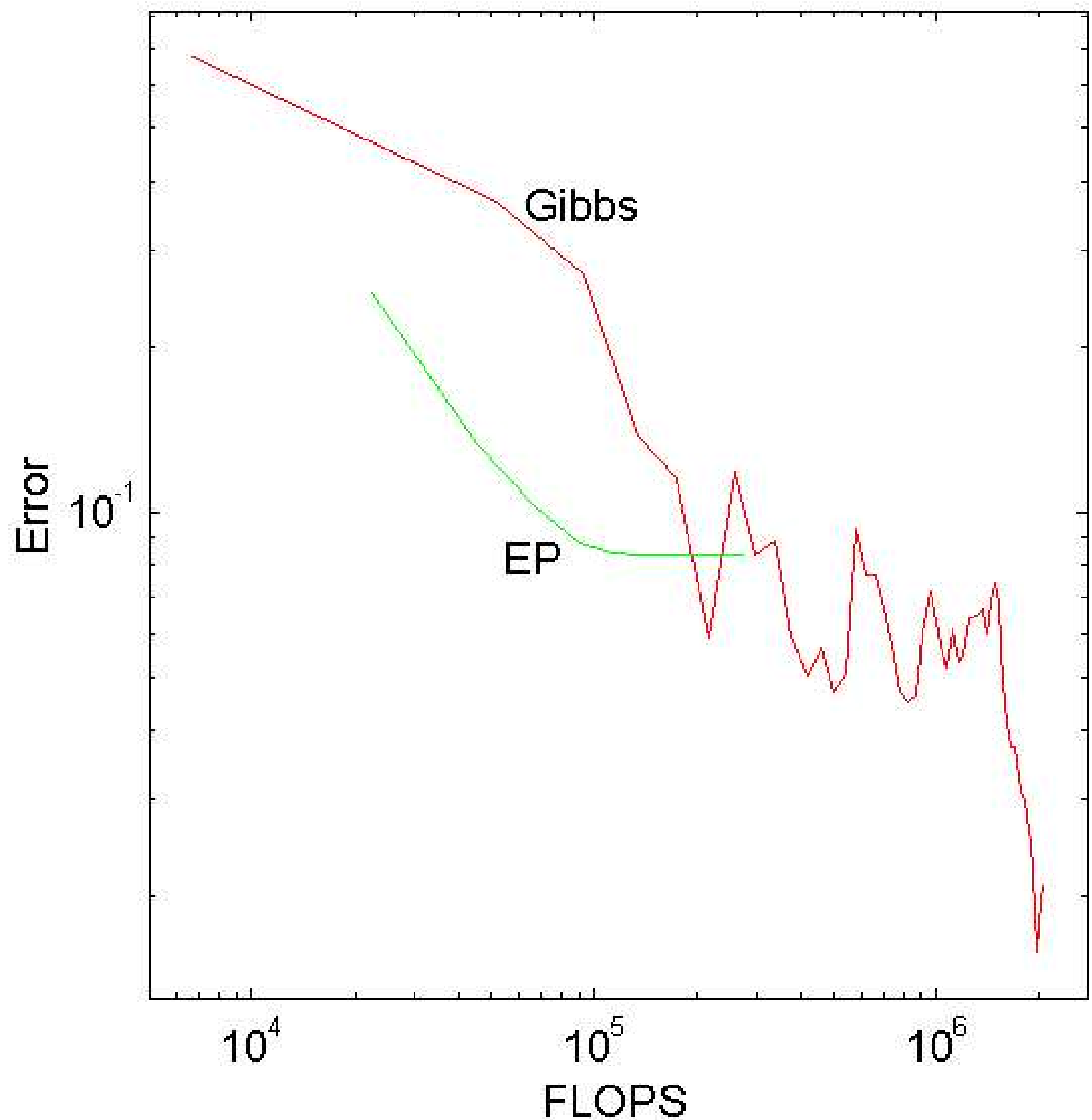
Order dependence

- Dirichlet process is exchangeable, but approximation quality does depend on order
- Best orderings are anti-correlated
  - Nearby points are far apart in the ordering
- Ordering is chosen by greedy selection of furthest point from picked points

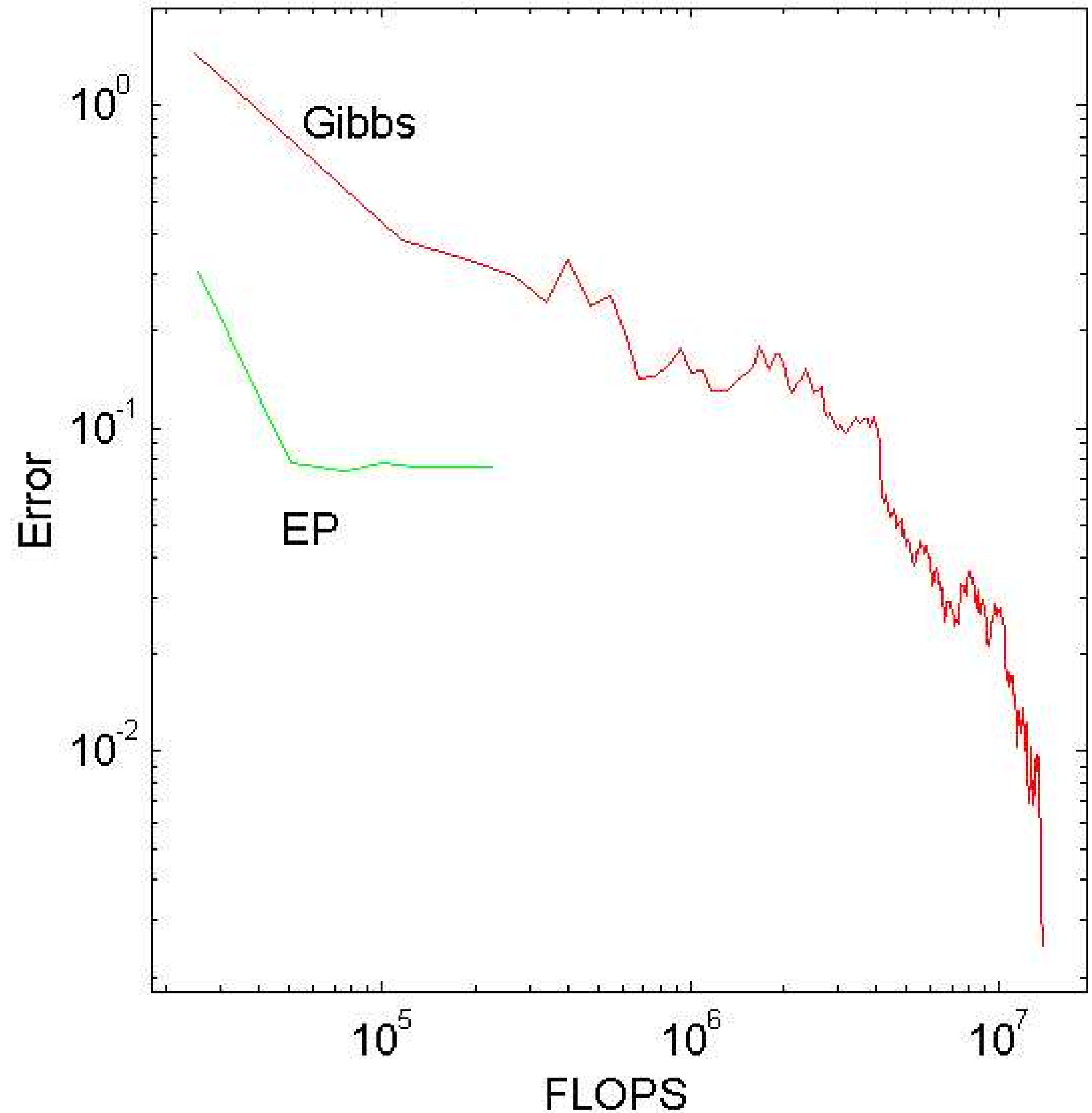
Random orderings



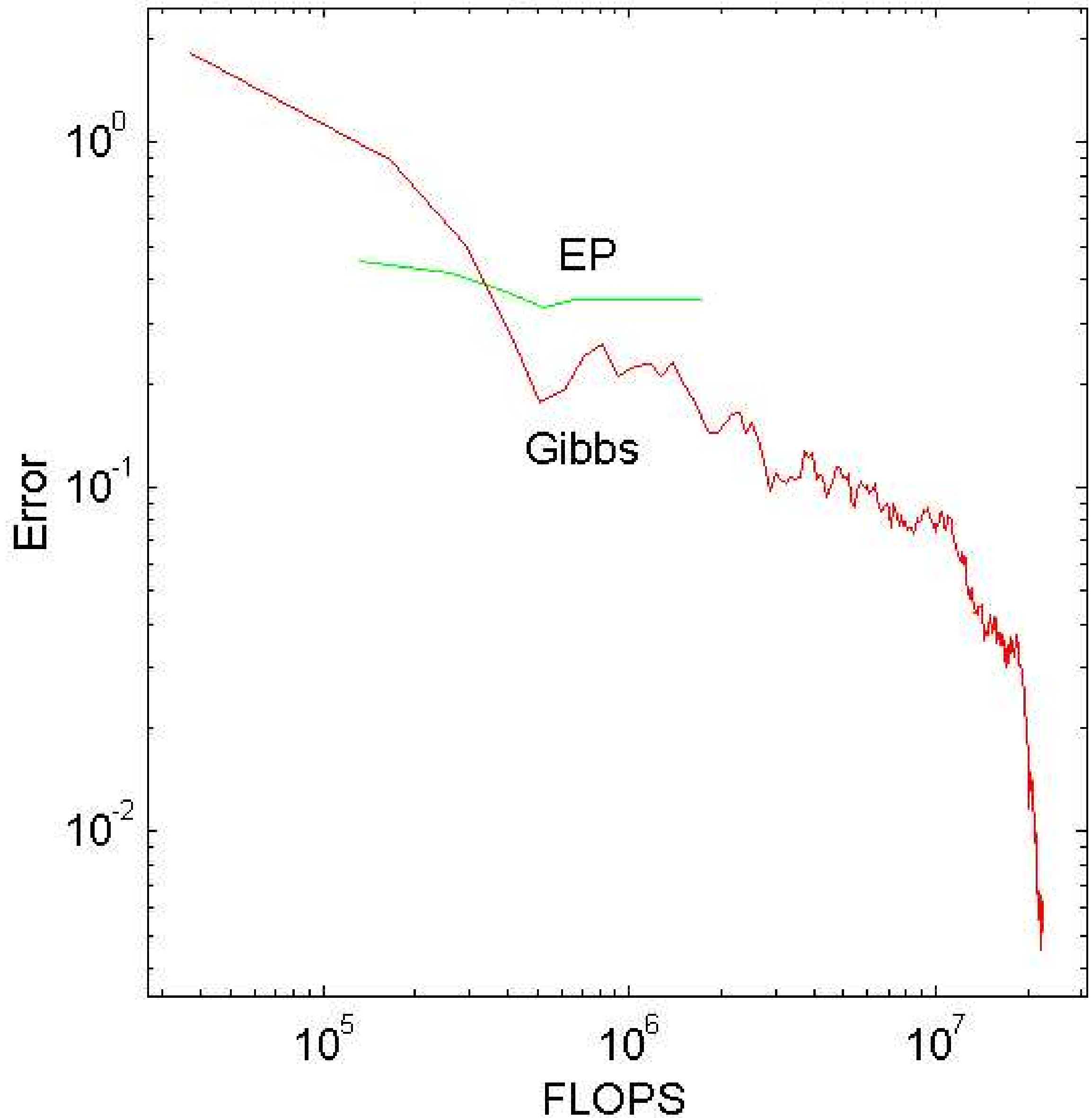
'x' ordering chosen by furthest-point heuristic


Gibbs
EP
Error
10^-1
10^4
10^5
10^6
FLOPS

20 points in one dimension
(ground truth from Gibbs)



20 points in two dimensions
(ground truth from Gibbs)



Computational cost

- Cost for EP grows faster than Gibbs
- Because it makes soft assignments, EP pays cost of maximum number of clusters (n)
- Because it makes hard assignments, Gibbs pays cost of actual number of clusters (<<n)
- Similar to EM versus k-means clustering
- Ignoring unlikely assignments would help

Accuracy of EP is limited

- Message to θ_j is weighted by prob of picking it (not prob of being in same cluster)
- Consider close-packed data
- θ_1 picks θ_2 , so that $\theta_1 = \theta_2$
- θ_3 can pick θ_1 or θ_2 equally, will send "half-weighted" message to each
- x_3 is weighted half as much as it should be

Conclusions

- EP with factorized approximation can give rough estimate faster than Gibbs
- Estimating hyperparameters is very easy
- But for high accuracy or high dimension, Gibbs is still method of choice

Suggestions for improvement

- The Dirichlet recursion can be written in different ways
  - But doesn't seem to help
- Posterior can be represented in terms of assignment variables, instead of parameters
- Approximation can be tree-structured instead of factorized (NIPS'03), allowing equality constraints to be remembered
  - Structure of tree must be learned from data