Variational Bayesian Learning

Tapani Raiko
April 17, 2008

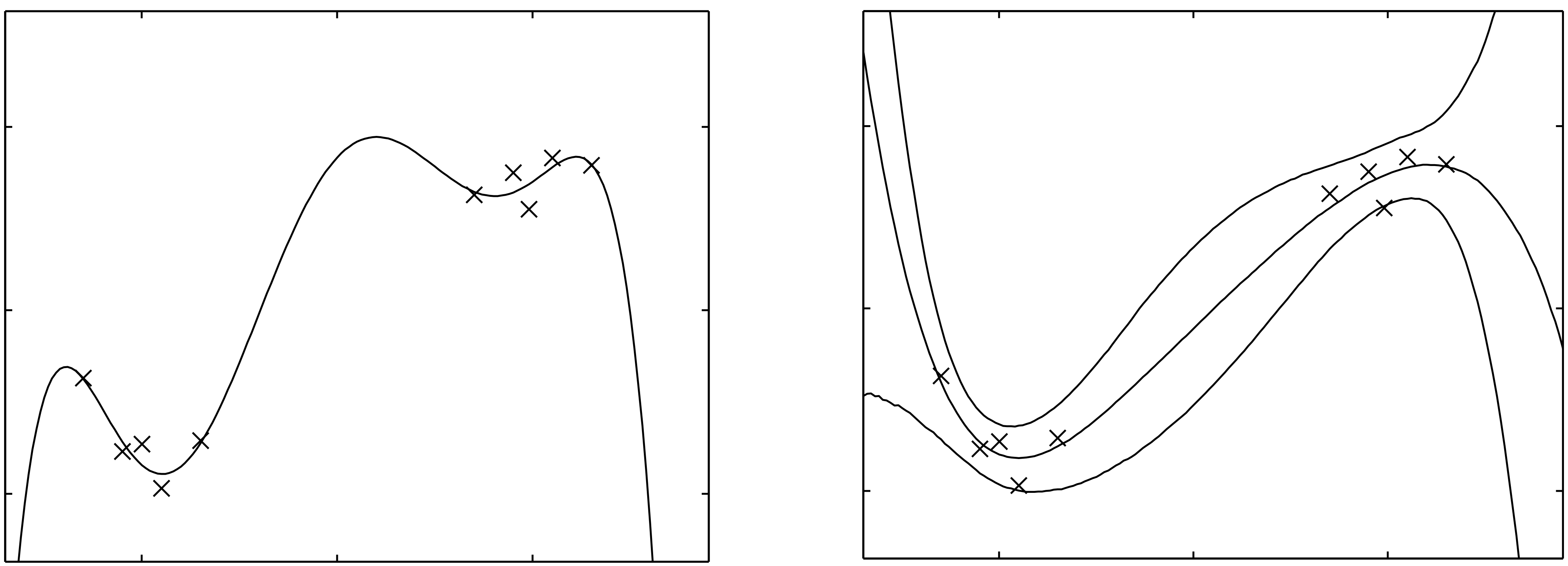
Machine learning: Advanced probabilistic methods

Motivation

- The main issue in probabilistic machine learning models is to find the posterior distribution over the model parameters and latent variables
- EM uses a point estimate for parameters which may be prone to over-fitting. Also, the E-step may not be solvable for some models.
- Sampling is prohibitively slow for large latent variable models
- Variational Bayesian (VB) learning is a good compromise

Overfitting

- An overfitted model explains the current data but does not generalize well to new data
- 6th order polynomial is fitted to 10 points by maximum likelihood and sampling



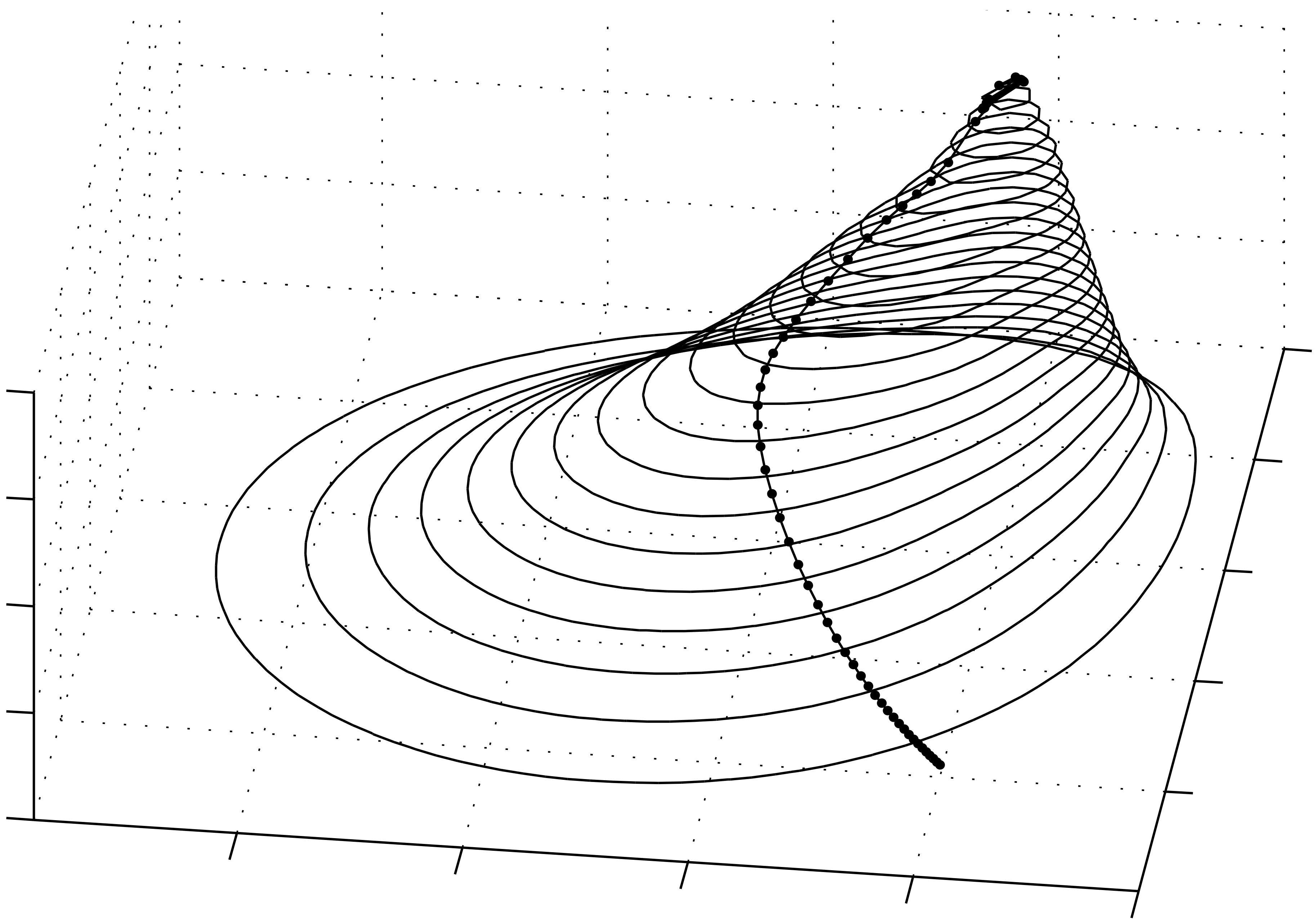
Posterior mass matters

- You want to make predictions about new data Y based on existing data X
- This is solved by fitting a model to the data and then predicting based on that

$$p(\mathbf{Y} \mid \mathbf{X}) = \int p(\mathbf{Y} \mid \mathbf{X}, \mathbf{Z}, \boldsymbol{\theta}) p(\mathbf{Z}, \boldsymbol{\theta} \mid \mathbf{X}) d\mathbf{Z} d\boldsymbol{\theta}$$

- Note how you need to integrate over the posterior $p(\mathbf{Z}, \boldsymbol{\theta} \mid \mathbf{X})$
- If you need to select a single solution $\mathbf{Z}, \boldsymbol{\theta}$, it should represent the posterior mass well

Why early stopping might help



Example: Probabilistic Principal Component Analysis (PCA)

$$\mathbf{x}_j = \mathbf{A}\mathbf{s}_j + \boldsymbol{\epsilon}_j \, .$$

$$p(\mathbf{s}_j) = \mathcal{N}\left(\mathbf{s}_j; 0, \mathbf{I}\right) \, , \qquad p(\boldsymbol{\epsilon}_j) = \mathcal{N}\left(\boldsymbol{\epsilon}_j; 0, v\mathbf{I}\right)$$

- Continuous-valued data vectors x are modelled as a linear mixture of source vectors s and noise
- Traditional PCA is the case where the noise goes to zero

Recap: EM-algorithm

- EM-algorithm solves latent variable models by alternating between two steps:
  - E-step updates the distribution over the latent variables Z
  - M-step updates the estimate of parameters $\boldsymbol{\theta}$

$$\text{E-step: } Q(\mathbf{Z}) \leftarrow P(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})$$

$$\text{M-step: } \boldsymbol{\theta} \leftarrow \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, E_{Q(\mathbf{Z})} \left\{ \ln P(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) \right\}$$

EM for PPCA

(don't learn the formulas by heart)

- The source posterior is a Gaussian:

$$p(\mathbf{S}|\mathbf{X}, \mathbf{A}, v) = \prod_{j=1}^{n} \mathcal{N}\left(\mathbf{s}_j; \overline{\mathbf{s}}_j, \mathbf{\Sigma}_\mathbf{s}\right)$$

- E-step:

$$\overline{\mathbf{S}} = \mathbf{\Psi}^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{X}\,, \qquad \mathbf{\Sigma}_\mathbf{s} = v\mathbf{\Psi}^{-1}\,, \qquad \mathbf{\Psi} = \mathbf{A}^{\mathrm{T}}\mathbf{A} + v\mathbf{I}\,.$$

- M-step:

$$\mathbf{A} = \mathbf{X}\mathbf{S}^{\mathrm{T}}(n\mathbf{\Sigma}_\mathbf{s} + \mathbf{S}\mathbf{S}^{\mathrm{T}})^{-1}$$

$$v = \frac{1}{nd}\sum_{i=1}^{d}\sum_{j=1}^{n}\left(x_{ij} - \mathbf{a}_i^{\mathrm{T}}\,\overline{\mathbf{s}}_j\right)^2 + \frac{1}{d}\,\mathrm{tr}(\mathbf{A}\mathbf{\Sigma}_\mathbf{s}\mathbf{A}^{\mathrm{T}})\,.$$

X=AS!?

- The model equation X=AS is symmetric with respect to A and S
- Why are A and S treated so differently?
- Would it be possible to model the posterior of both A and S with a Gaussian?

VB-EM algorithm

- The VB-EM algorithm alternates between updates for the latent variables and parameters

- Steps are symmetric and they resemble the E-step of the EM algorithm

- VB-E step:

$$q(\mathbf{Z}) \leftarrow \underset{q(\mathbf{Z})}{\operatorname{argmin}} E_{q(\boldsymbol{\theta})} \left\{ \mathrm{KL}\left(q(\mathbf{Z}) \parallel p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}) \right) \right\}$$

- VB-M step:

$$q(\boldsymbol{\theta}) \leftarrow \underset{q(\boldsymbol{\theta})}{\operatorname{argmin}} E_{q(\mathbf{Z})} \left\{ \mathrm{KL}\left(q(\boldsymbol{\theta}) \parallel p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{Z}) \right) \right\}$$

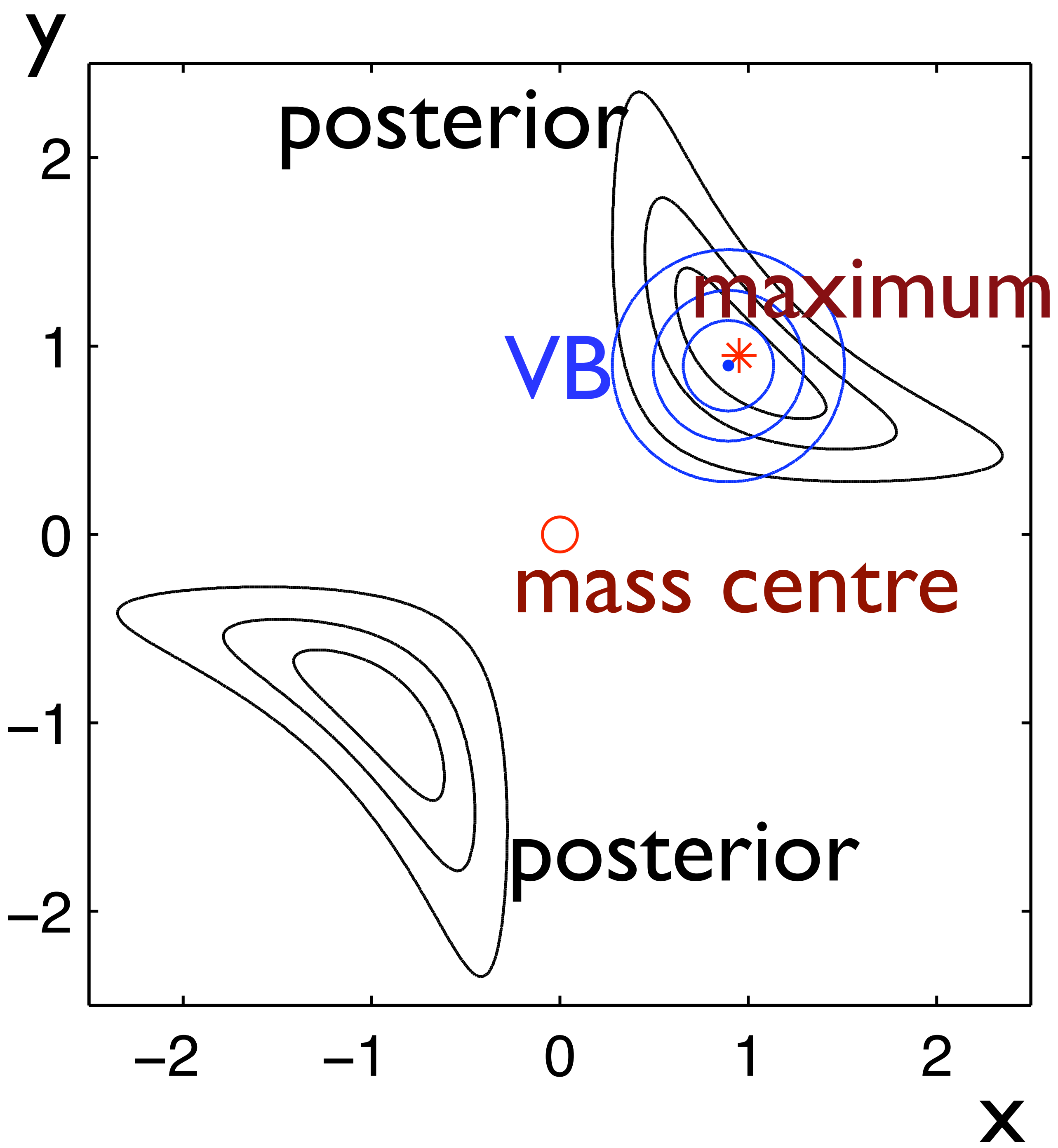
Variational Bayes (key slide!)

- VB works by fitting a distribution q over the unknown variables to the true posterior by minimizing the KL divergence:

$$\mathrm{KL}\left(q(\mathbf{Z},\boldsymbol{\theta}) \parallel p(\mathbf{Z},\boldsymbol{\theta} \mid \mathbf{X})\right) = E_{q(\mathbf{Z},\boldsymbol{\theta})}\left\{\ln \frac{q(\mathbf{Z},\boldsymbol{\theta})}{p(\mathbf{Z},\boldsymbol{\theta} \mid \mathbf{X})}\right\}$$

- The form of q can be chosen such that the expectations are tractable
- For instance, $q(\mathbf{Z},\boldsymbol{\theta}) = q(\mathbf{Z})q(\boldsymbol{\theta})$ is assumed almost always, allowing the VB-EM algorithm

Example 1



- model

$$p(z) = \mathcal{N}(z; xy, 0.02)$$

- prior

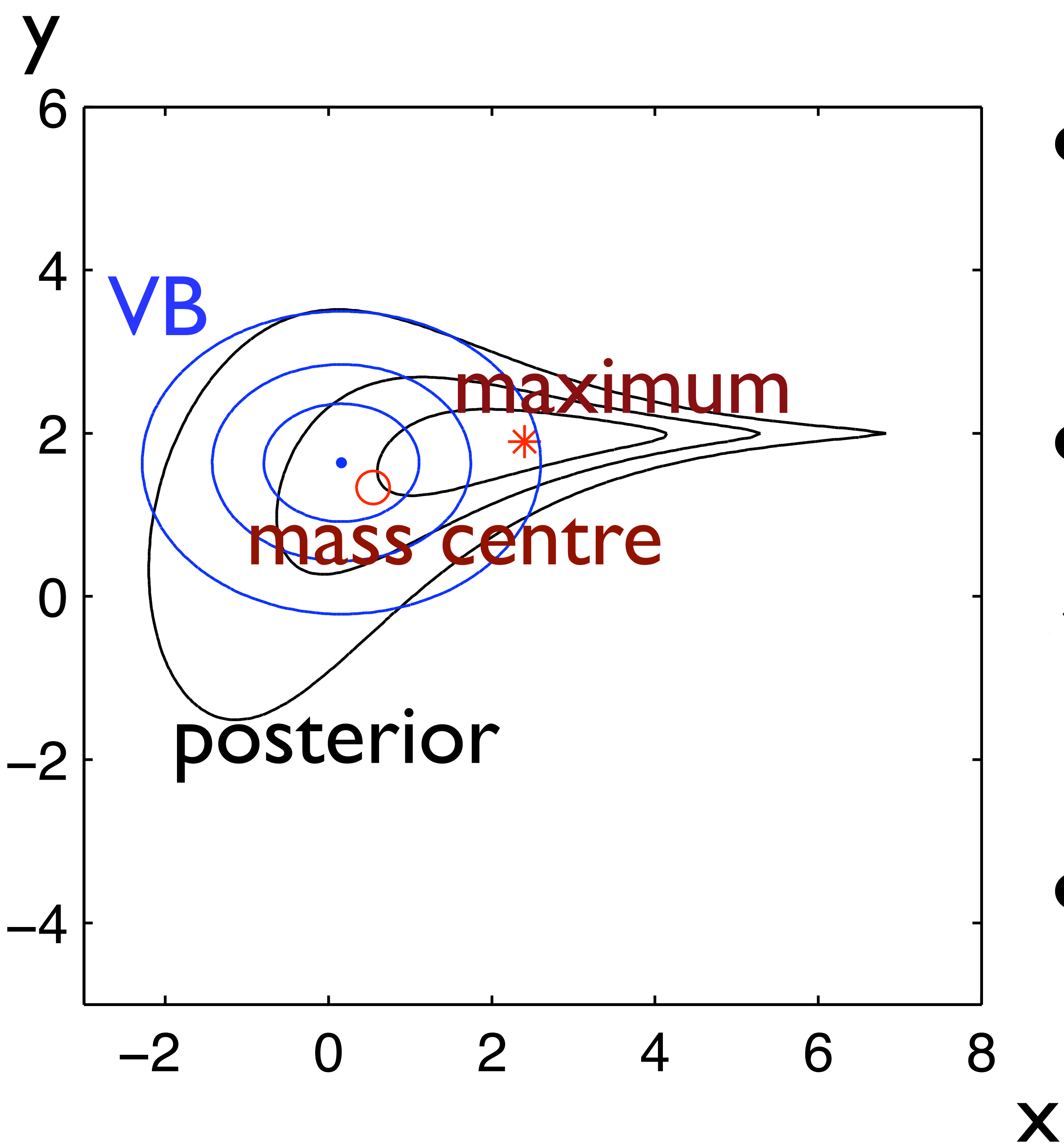
$$p(x) = \mathcal{N}(x; 0, 1),$$

$$p(y) = \mathcal{N}(y; 0, 1).$$

- data

$$z = 1$$

Example 2



- model

$$p(z) = \mathcal{N}\left(z; y, \exp(-x)\right)$$

- prior

$$p(x) = \mathcal{N}\left(x; -1, 5\right)$$

$$p(y) = \mathcal{N}\left(y; 0, 5\right).$$

- data

$$z = 2.$$

VB-EM for PCA

(don't learn the formulas by heart)

$$q(\mathbf{A}, \mathbf{S}) = \prod_{i=1}^{d} \mathcal{N}\left(\mathbf{a}_i; \overline{\mathbf{a}}_i, \boldsymbol{\Sigma}_{\mathbf{a}}\right) \prod_{j=1}^{n} \mathcal{N}\left(\mathbf{s}_j; \overline{\mathbf{s}}_j, \boldsymbol{\Sigma}_{\mathbf{s}}\right) \ .$$

$$\overline{\mathbf{S}} = \boldsymbol{\Psi}^{-1}\overline{\mathbf{A}}^{\mathrm{T}}\mathbf{X}\,, \qquad \boldsymbol{\Sigma}_{\mathbf{s}} = v\boldsymbol{\Psi}^{-1}$$

$$\boldsymbol{\Psi} = \overline{\mathbf{A}}^{\mathrm{T}}\overline{\mathbf{A}} + d\boldsymbol{\Sigma}_{\mathbf{a}} + v\mathbf{I}\,.$$

$$\overline{\mathbf{A}} = \boldsymbol{\Phi}^{-1}\overline{\mathbf{S}}\mathbf{X}\,, \qquad \boldsymbol{\Sigma}_{\mathbf{a}} = v\boldsymbol{\Phi}^{-1}$$

$$\boldsymbol{\Phi} = \overline{\mathbf{S}\mathbf{S}}^{\mathrm{T}} + n\boldsymbol{\Sigma}_{\mathbf{s}} + v\,\mathrm{diag}(w_k^{-1})$$

$$v = \frac{1}{nd}\sum_{i=1}^{d}\sum_{j=1}^{n}\left(x_{ij} - \overline{\mathbf{a}}_i^{\mathrm{T}}\,\overline{\mathbf{s}}_j\right)^2 + \frac{1}{d}\,\mathrm{tr}(\overline{\mathbf{A}}\boldsymbol{\Sigma}_{\mathbf{s}}\overline{\mathbf{A}}^{\mathrm{T}})\frac{1}{n}\,\mathrm{tr}(\overline{\mathbf{S}}^{T}\boldsymbol{\Sigma}_{\mathbf{a}}\overline{\mathbf{S}}) + \frac{1}{nd}\,\mathrm{tr}(\boldsymbol{\Sigma}_{\mathbf{s}}\boldsymbol{\Sigma}_{\mathbf{a}})\,.$$

Compare to EM

- The source posterior is a Gaussian:

$$p(\mathbf{S}|\mathbf{X},\mathbf{A},v) = \prod_{j=1}^{n} \mathcal{N}\left(\mathbf{s}_j; \overline{\mathbf{s}}_j, \mathbf{\Sigma}_\mathbf{s}\right)$$

- E-step:

$$\overline{\mathbf{S}} = \mathbf{\Psi}^{-1}\mathbf{A}^\mathrm{T}\mathbf{X}\,, \qquad \mathbf{\Sigma}_\mathbf{s} = v\mathbf{\Psi}^{-1}\,, \qquad \mathbf{\Psi} = \mathbf{A}^\mathrm{T}\mathbf{A} + v\mathbf{I}\,.$$

- M-step: $\mathbf{A} = \mathbf{X}\mathbf{S}^\mathrm{T}(n\mathbf{\Sigma}_\mathbf{s} + \mathbf{S}\mathbf{S}^\mathrm{T})^{-1}$

$$v = \frac{1}{nd}\sum_{i=1}^{d}\sum_{j=1}^{n}\left(x_{ij} - \mathbf{a}_i^\mathrm{T}\,\overline{\mathbf{s}}_j\right)^2 + \frac{1}{d}\,\mathrm{tr}(\mathbf{A}\mathbf{\Sigma}_\mathbf{s}\mathbf{A}^\mathrm{T})\,.$$

Model selection

- The cost function that is minimized in practice is also includes a part for model evidence p(X|M)

$$
\begin{aligned}
\mathcal{C}_{VB} &= E_q\left\{\ln\frac{q(\mathbf{Z},\boldsymbol{\theta})}{p(\mathbf{X},\mathbf{Z},\boldsymbol{\theta}\mid M_i)}\right\} \\
&= \mathrm{KL}\left(q(\mathbf{Z},\boldsymbol{\theta})\,\|\,p(\mathbf{Z},\boldsymbol{\theta}\mid\mathbf{X},M_i)\right) - \ln p(\mathbf{X}\mid M_i) \\
&\geq -\ln p(\mathbf{X}\mid M_i)
\end{aligned}
$$

- By minimizing the cost, we get a lower bound for the model evidence
- We can thus compare different models M

Learning algorithms

- q can be parameterized for instance by posterior means and covariances
- Those variational parameters can then be updated by any means to minimize to cost

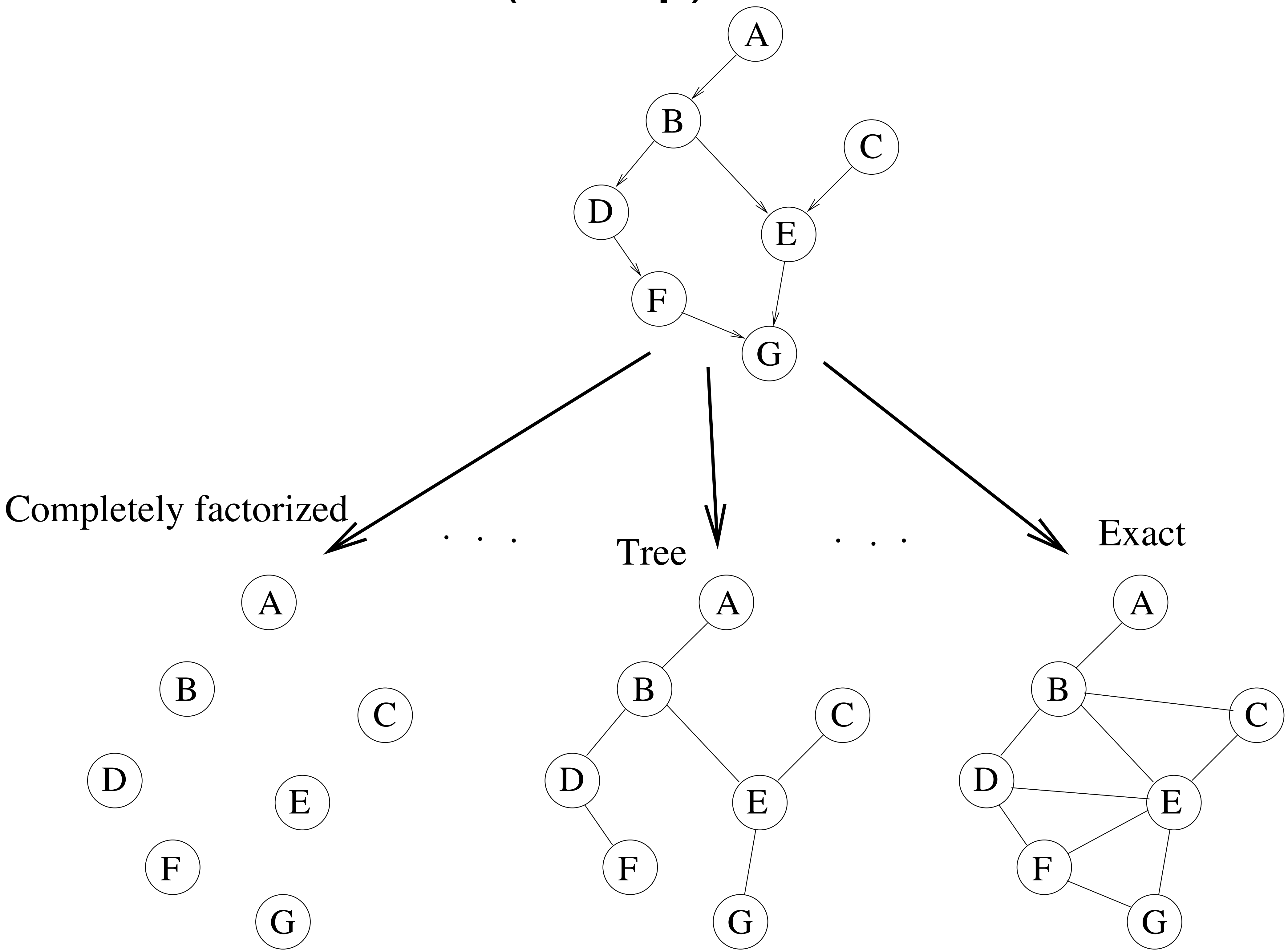
$$\mathcal{C}_{VB} = E_q \left\{ \ln \frac{q(\mathbf{Z}, \boldsymbol{\theta})}{p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta} \mid M_i)} \right\}$$

- This is useful if the VB-EM updates are intractable
- Gradient based methods can be faster, too

Discrete models

- Consider VB learning of Bayesian networks
- Instead of a single set of parameters (conditional probability tables), we would have distribution $q(\boldsymbol{\theta})$ over the parameters
- The certainty of CPTs would be estimated
- The VB cost function could be used to select the best model structure (it penalizes complex models automatically)

- By restricting the form of $q(\mathbf{Z})$, the inference (E-step) can be made faster



Pros and cons of VB

- + Robust against overfitting
- + Fast (compared to sampling)
- + Applicable to a large family of models
- - Intensive formulae (lots of integrals)
- - Prone to bad but locally optimal solutions (lot of work with arranging good initializations and other tricks to avoid them)

Software packages for VB on Bayesian networks (1/2)

- VIBES by Winn and Bishop
  - discrete and continuous values
  - posterior approximation is factorized such that disjoint groups of variables are independent but dependencies within the group are modelled
  - variational message passing algorithm

Software packages for VB on Bayesian networks (2/2)

- Bayes Block by Valpola et al.
  - concentrates on continuous values
  - fully factorial posterior approximation
  - includes nonlinearities
  - allows for variance modelling
  - message passing with line searches for speed-up