Principal Component Analysis (PCA) for Sparse High-Dimensional Data

Tapani Raiko

Helsinki University of Technology, Finland
Adaptive Informatics Research Center

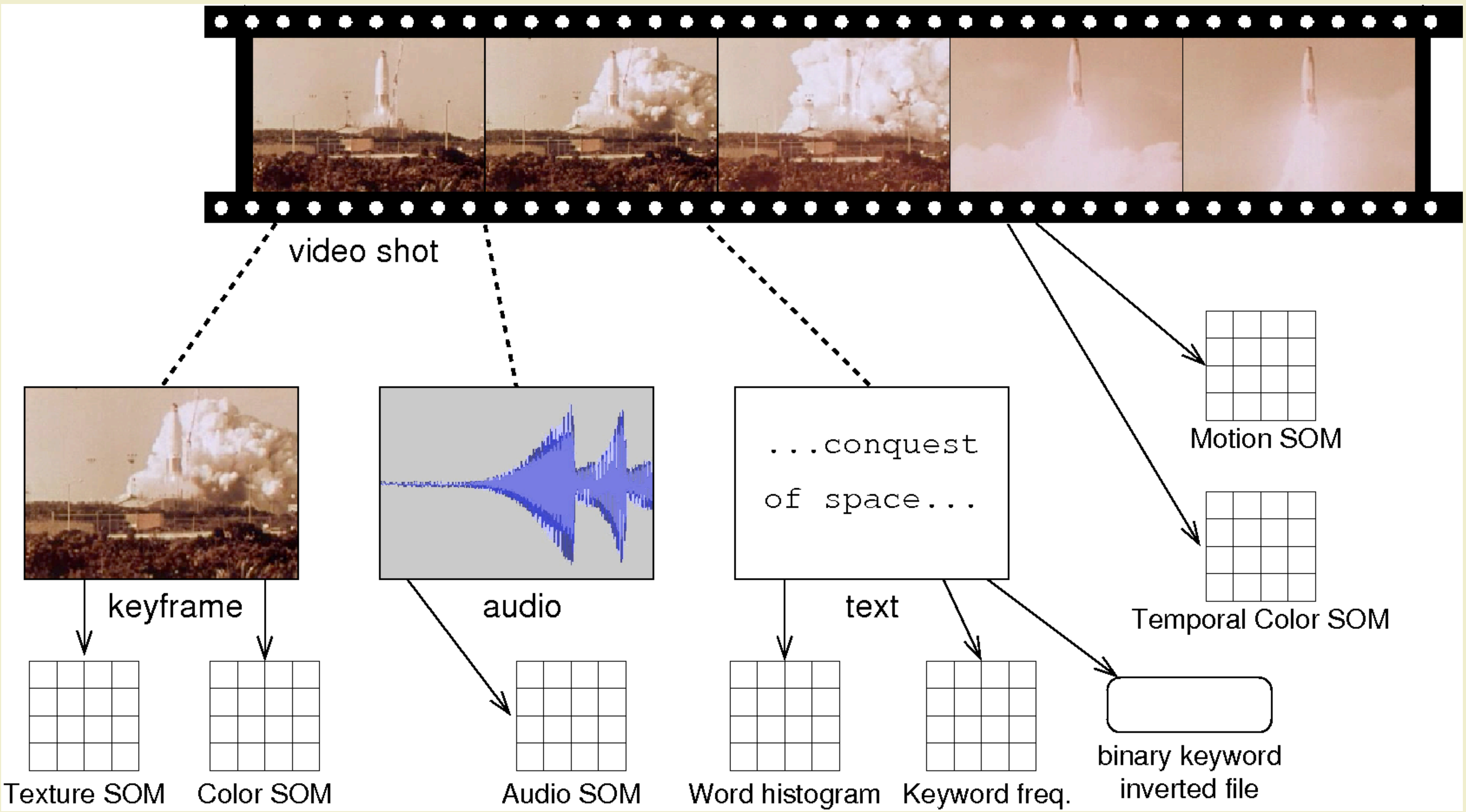
The Data Explosion

- We are facing an enormous challenge in the ever increasing amount of data in electronic form
- First wave: text, second wave: real-world data
- Basically, any information that may have value will be made available, e.g., through the Web
- We need ”adaptive informatics” which adds intelligence at the access point.

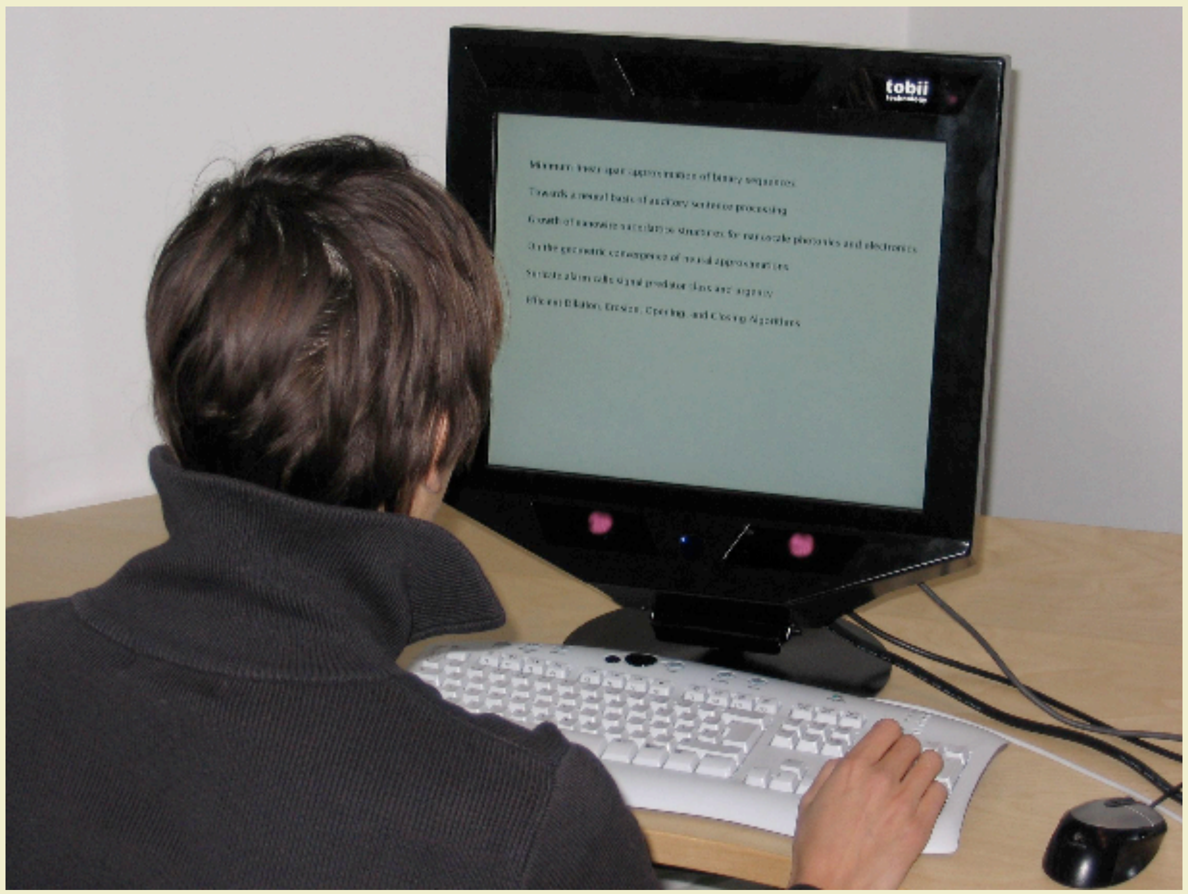
Adaptive Informatics:

- A field of research where automated learning algorithms are used to discover informative concepts, components, and their mutual relations from large amounts of real-world data
- The goal is to understand the underlying phenomena, structures, and patterns buried in the large data sets, in order to make the information usable.

Retrieval of multimodel objects:



Proactive Information Retrieval



Principal Component Analysis

- Data X consists of n d-dimensional vectors
- Matrix X is decomposed in to a product of smaller matrices such that the square reconstruction error is minimized

$$\mathbf{X} \approx \mathbf{A}\mathbf{S},$$

$$C = \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_F^2 = \sum_{i=1}^{d}\sum_{j=1}^{n}(x_{ij} - \sum_{k=1}^{c} a_{ik}s_{kj})^2$$

Algorithms for PCA

- Eigenvalue decomposition (standard approach)
  - Compute the covariance matrix and its eigenvectors

Algorithms for PCA

- Eigenvalue decomposition (standard approach)
  - Compute the covariance matrix and its eigenvectors
- EM algorithm
  - Iterates between:

$$\mathbf{A} \leftarrow \mathbf{X}\mathbf{S}^{\mathrm{T}}(\mathbf{S}\mathbf{S}^{\mathrm{T}})^{-1}, \qquad \mathbf{S} \leftarrow (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{X}.$$

Algorithms for PCA

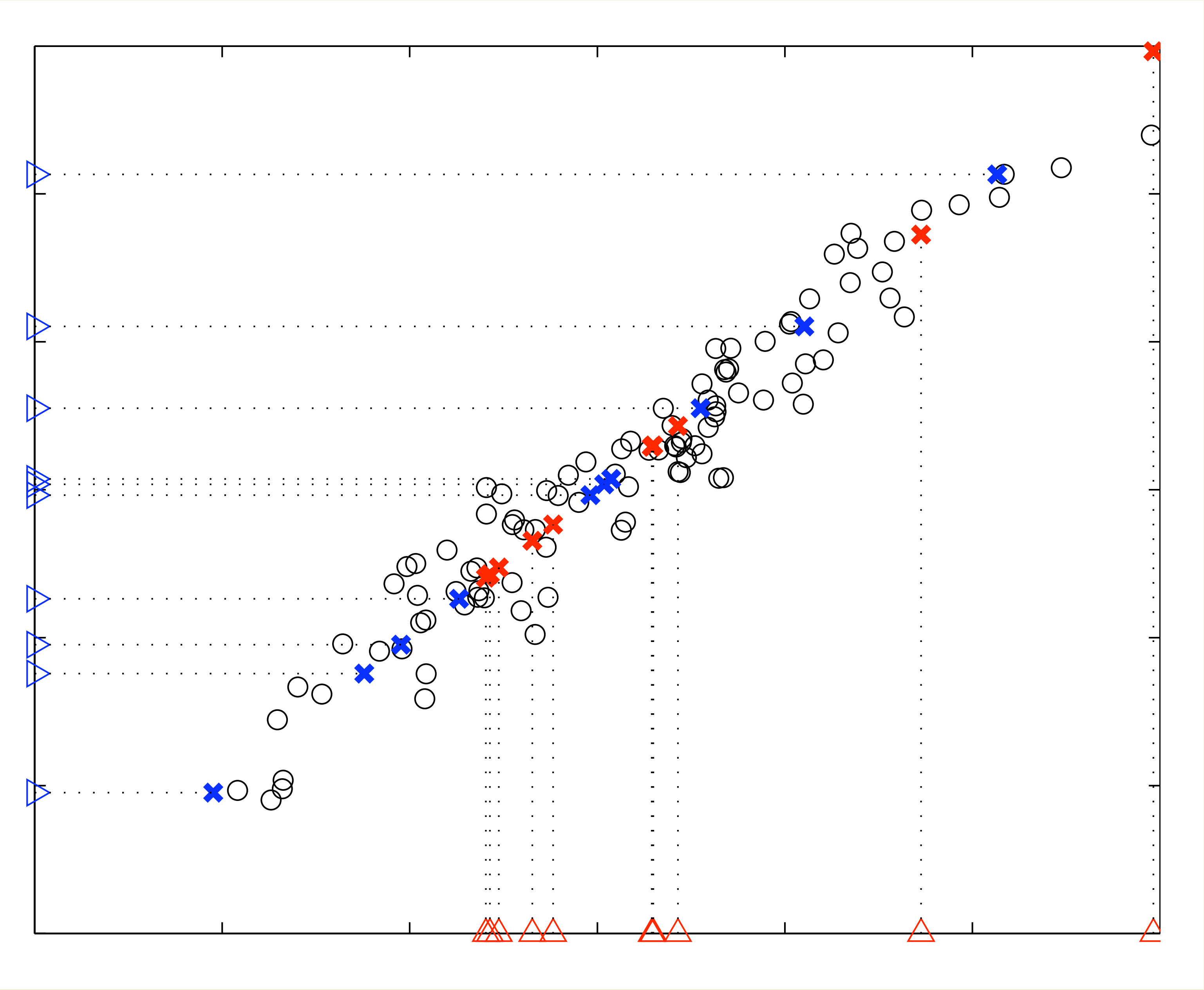
- Eigenvalue decomposition (standard approach)
  - Compute the covariance matrix and its eigenvectors
- EM algorithm
  - Iterates between:

$$\mathbf{A} \leftarrow \mathbf{X}\mathbf{S}^{\mathrm{T}}(\mathbf{S}\mathbf{S}^{\mathrm{T}})^{-1}, \qquad \mathbf{S} \leftarrow (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{X}.$$

- Minimization of cost C (Oja's subspace rule)

$$\mathbf{A} \leftarrow \mathbf{A} + \gamma(\mathbf{X} - \mathbf{A}\mathbf{S})\mathbf{S}^{\mathrm{T}}, \qquad \mathbf{S} \leftarrow \mathbf{S} + \gamma\mathbf{A}^{\mathrm{T}}(\mathbf{X} - \mathbf{A}\mathbf{S}).$$

PCA with Missing Values



- Red and blue data points are reconstructed based on only one of the two dimensions

Adapting the Algorithms for Missing Values

- Iterative imputation
  - Alternately 1) fill in missing values and 2) solve normal PCA with the standard approach

Adapting the Algorithms for Missing Values

- Iterative imputation
  - Alternately 1) fill in missing values and 2) solve normal PCA with the standard approach
- EM algorithm becomes computationally heavier

$\mathbf{S}$	$\mathbf{A}$
$\mathbf{s}_{:j} = (\mathbf{A}_j^{\mathrm{T}} \mathbf{A}_j)^{-1} \mathbf{A}_j^{\mathrm{T}} \mathring{\mathbf{X}}_{:j}$ $j = 1, \ldots, n$	$\mathbf{A}_{i:}^{\mathrm{T}} = \mathring{\mathbf{X}}_{i:}^{\mathrm{T}} \mathbf{S}_i^{\mathrm{T}} (\mathbf{S}_i \mathbf{S}_i^{\mathrm{T}})^{-1}$ $i = 1, \ldots, d$

Adapting the Algorithms for Missing Values

- Iterative imputation
  - Alternately 1) fill in missing values and 2) solve normal PCA with the standard approach
- EM algorithm becomes computationally heavier

$\mathbf{S}$	$\mathbf{A}$
$\mathbf{s}_{:j} = (\mathbf{A}_j^{\mathrm{T}}\mathbf{A}_j)^{-1}\mathbf{A}_j^{\mathrm{T}}\mathring{\mathbf{X}}_{:j}$ $j = 1, \ldots, n$	$\mathbf{A}_{i:}^{\mathrm{T}} = \mathring{\mathbf{X}}_{i:}^{\mathrm{T}}\mathbf{S}_i^{\mathrm{T}}(\mathbf{S}_i\mathbf{S}_i^{\mathrm{T}})^{-1}$ $i = 1, \ldots, d$

- Minimization of cost C
  - Easy to adapt: Take error over observed values only

Speeding up Gradient Descent

- Newton's method is known to converge fast, but
  - It requires computing the Hessian matrix which is computationally too demanding in high-dimensional problems
- We propose using only the diagonal part of the Hessian
- We also include a control parameter to interpolate between standard gradient descent (0) and the diagonal Newton's method (1)

The cost function:

$$C = \sum_{(i,j)\in O} e_{ij}^2 \,, \qquad \text{with} \qquad e_{ij} = x_{ij} - \sum_{k=1}^{c} a_{ik} s_{kj} \,.$$

The cost function:

$$C = \sum_{(i,j)\in O} e_{ij}^2 , \qquad \text{with} \qquad e_{ij} = x_{ij} - \sum_{k=1}^{c} a_{ik} s_{kj} .$$

Its partial derivatives:

$$\frac{\partial C}{\partial a_{il}} = -2 \sum_{j|(i,j)\in O} e_{ij} s_{lj} , \qquad \frac{\partial C}{\partial s_{lj}} = -2 \sum_{i|(i,j)\in O} e_{ij} a_{il} .$$

The cost function:

$$C = \sum_{(i,j)\in O} e_{ij}^2 \,, \qquad \text{with} \qquad e_{ij} = x_{ij} - \sum_{k=1}^{c} a_{ik} s_{kj} \,.$$

Its partial derivatives:

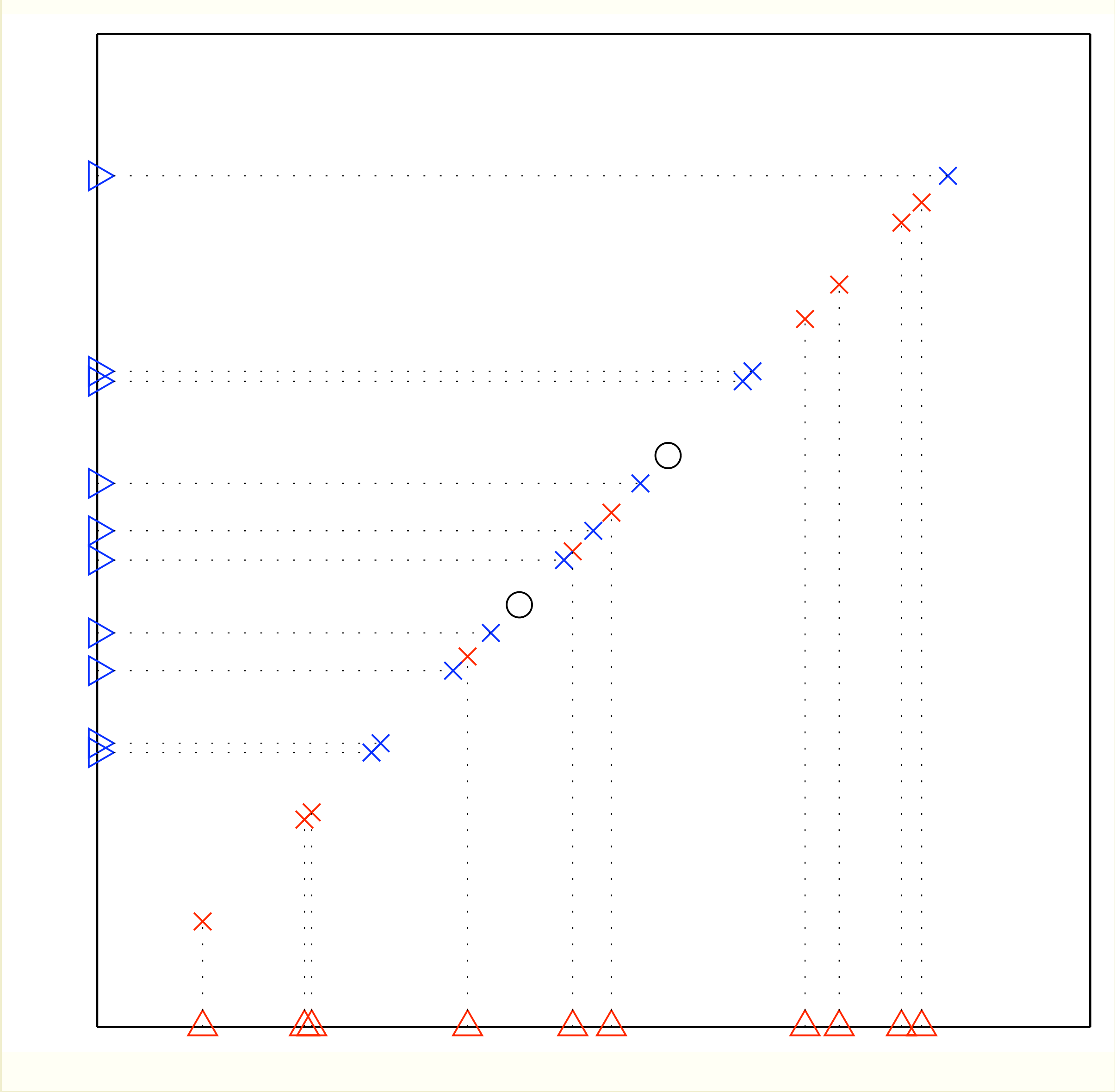
$$\frac{\partial C}{\partial a_{il}} = -2 \sum_{j|(i,j)\in O} e_{ij} s_{lj} \,, \qquad \frac{\partial C}{\partial s_{lj}} = -2 \sum_{i|(i,j)\in O} e_{ij} a_{il} \,.$$

Update rules:

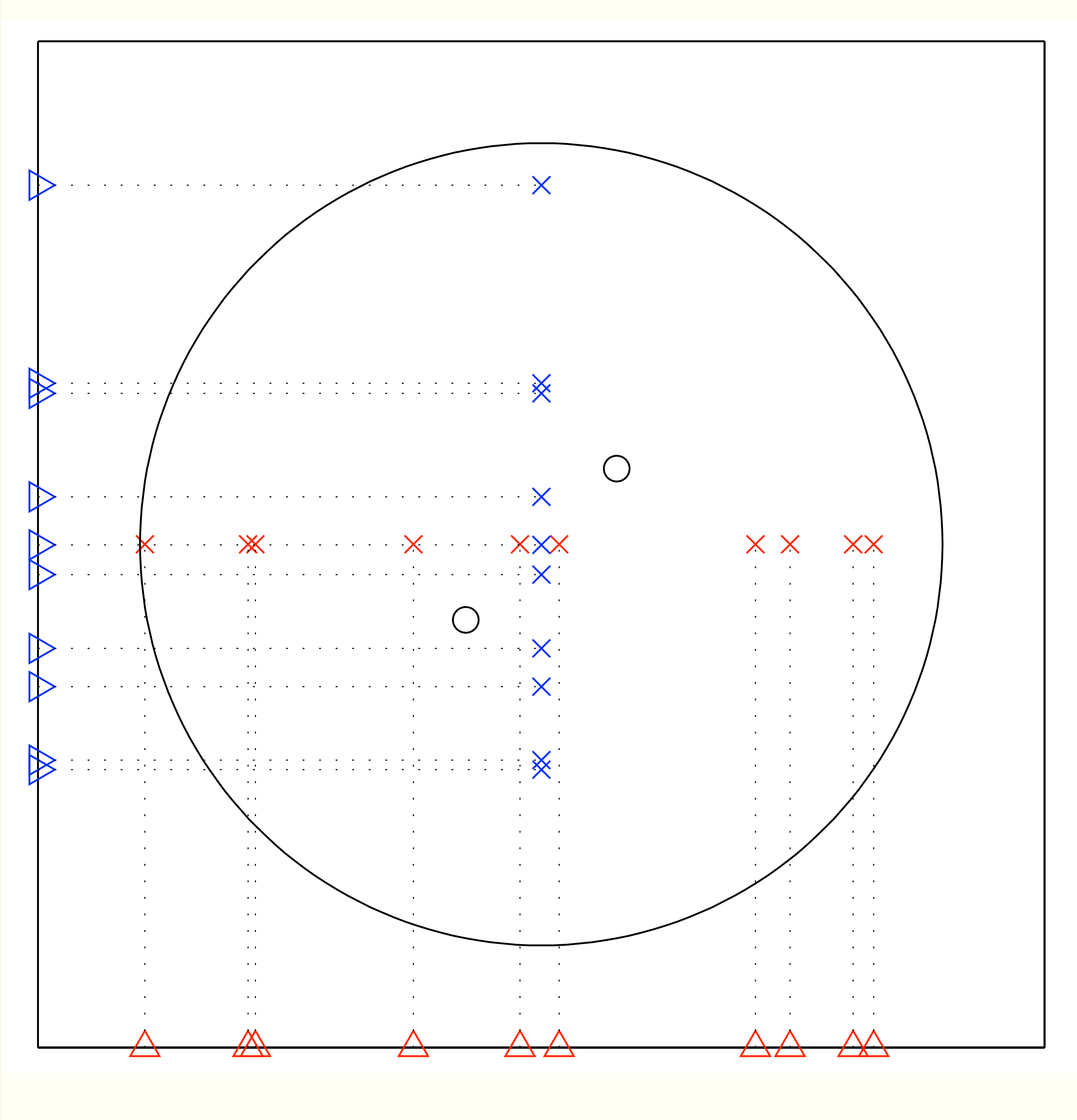
$$a_{il} \leftarrow a_{il} - \gamma' \left(\frac{\partial^2 C}{\partial a_{il}^2} \right)^{-\alpha} \frac{\partial C}{\partial a_{il}} = a_{il} + \gamma \frac{\sum_{j|(i,j)\in O} e_{ij} s_{lj}}{\left(\sum_{j|(i,j)\in O} s_{lj}^2 \right)^{\alpha}} \,,$$

$$s_{lj} \leftarrow s_{lj} - \gamma' \left(\frac{\partial^2 C}{\partial s_{lj}^2} \right)^{-\alpha} \frac{\partial C}{\partial s_{lj}} = s_{lj} + \gamma \frac{\sum_{i|(i,j)\in O} e_{ij} a_{il}}{\left(\sum_{i|(i,j)\in O} a_{il}^2 \right)^{\alpha}} \,.$$

Overfitting in Case of Sparse Data

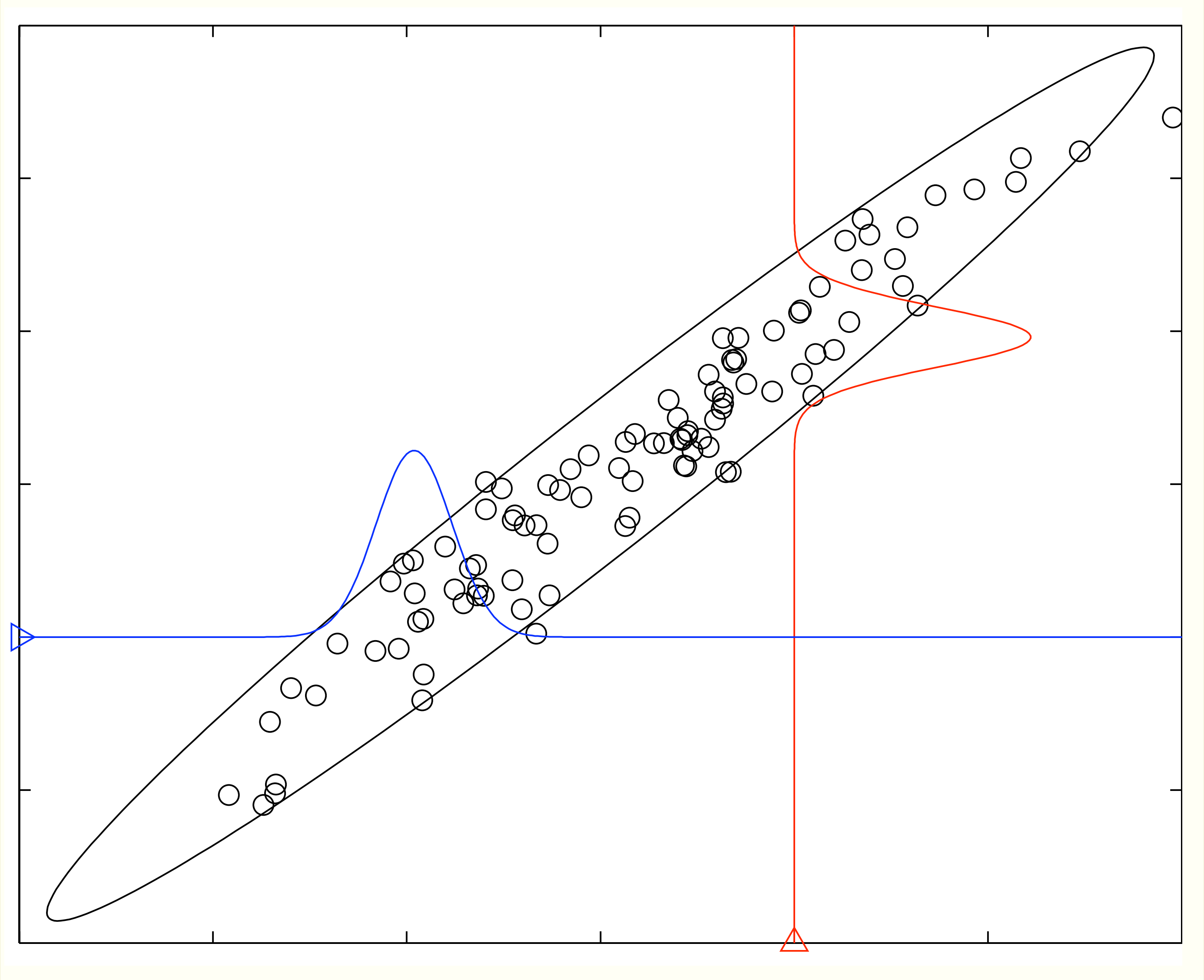


Overfitted solution



Regularized solution

Regularization against Overfitting



- Penalizing the use of large parameter values
- Estimating the distribution of unknown parameters (Variational Bayesian learning)

Experiments with Netflix Data
www.netflixprize.com

- Collaborative filtering task: predict people's preferences based on other people's preferences
- d = 18 000 movies, n = 500 000 customers, N = 100 000 000 movie ratings from 1 to 5
- 98.8% of the values are missing
- Find c=15 principal components

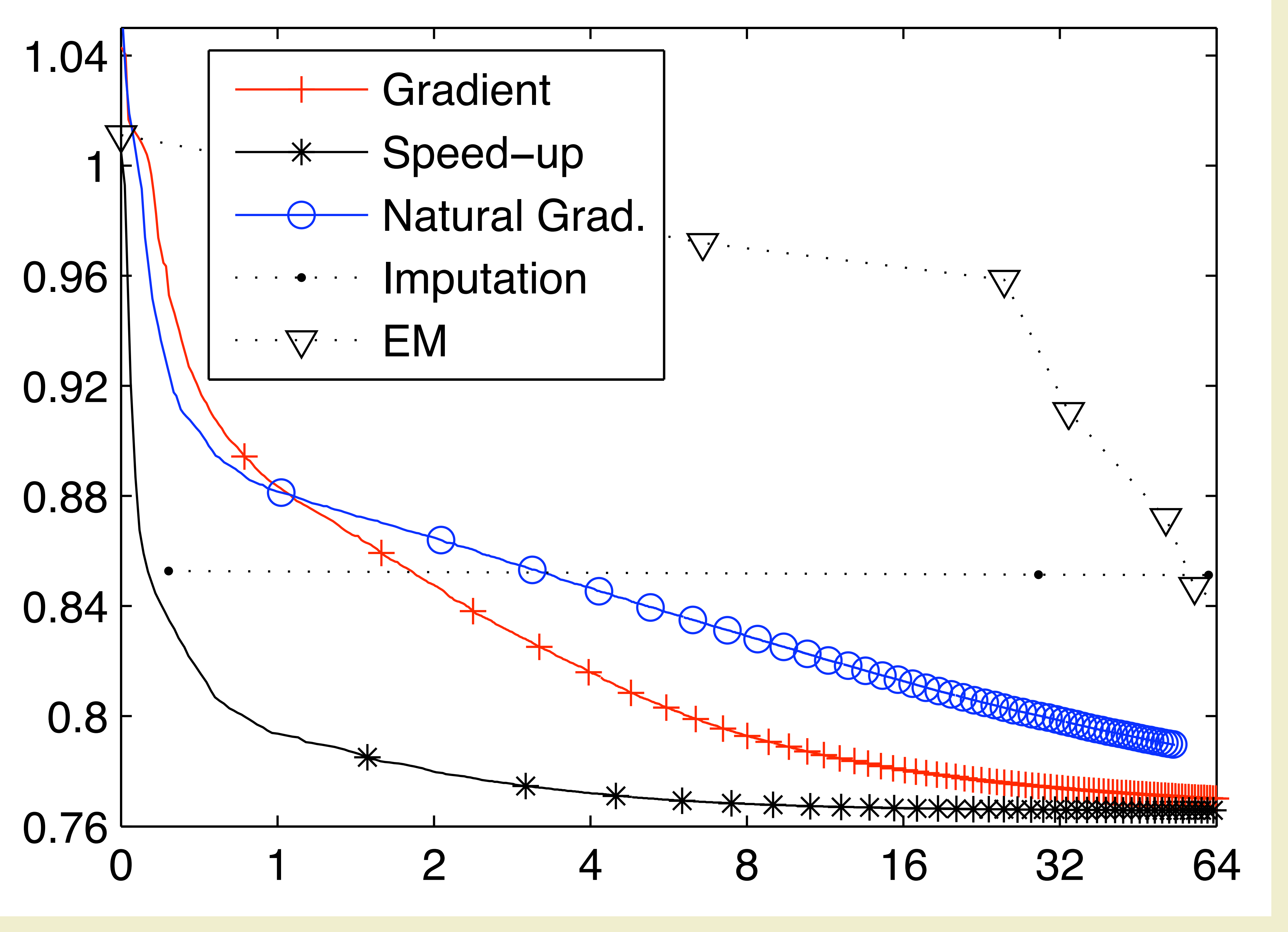
Computational Performance

Method	Complexity	Seconds/Iter	Hours to $E_O = 0.85$
Gradient	$O(Nc + nc)$	58	1.9
Speed-up	$O(Nc + nc)$	110	0.22
Natural Grad.	$O(Nc + nc^2)$	75	3.5
Imputation	$O(nd^2)$	110000	$\gg 64$
EM	$O(Nc^2 + nc^3)$	45000	58

- N=100 000 000, # of ratings
- c=15, # of components
- n=500 000, # of people
- d=18 000, # of movies

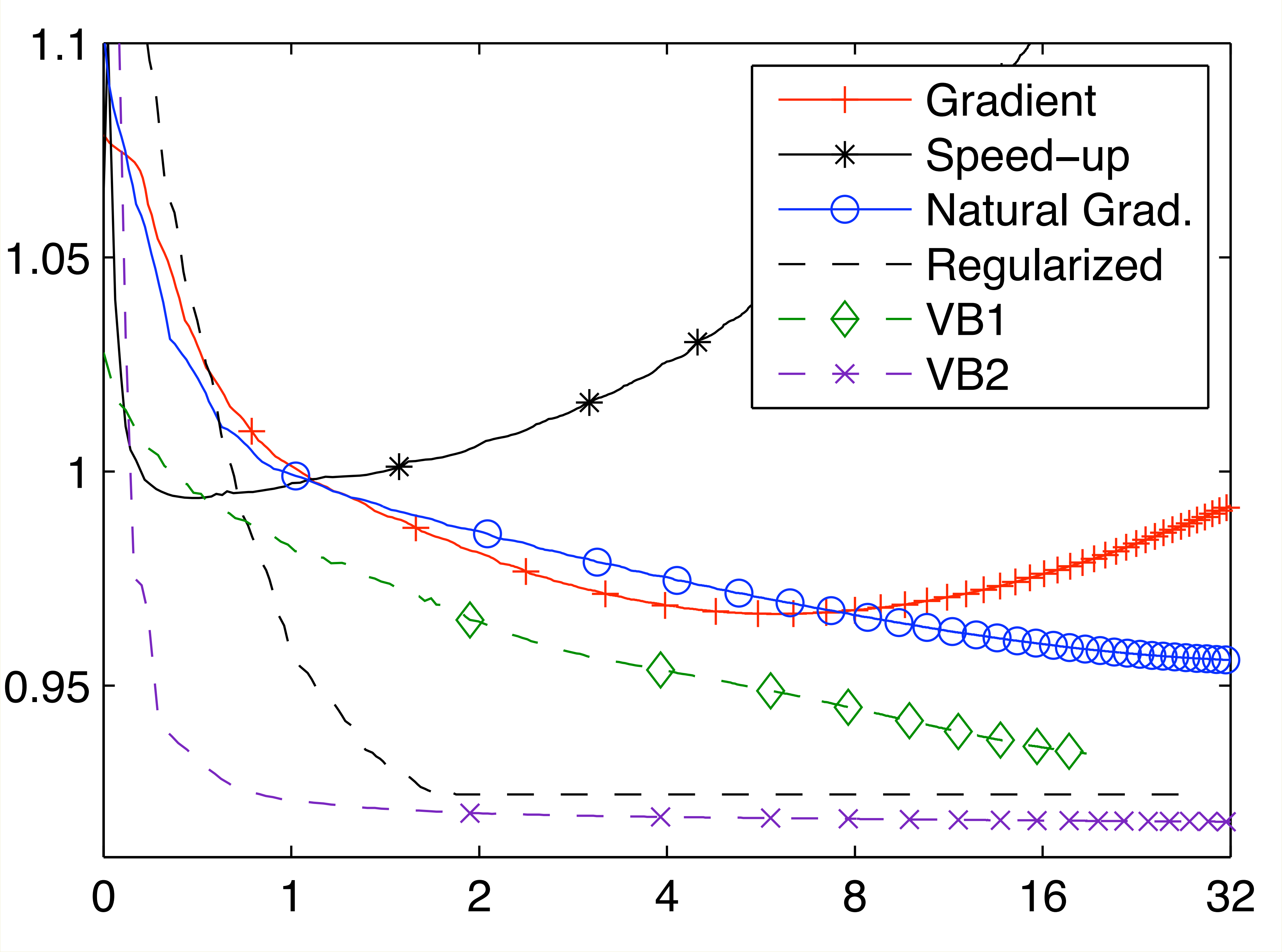
Error on Training Data

against computation time in hours



Error on Validation Data

against computation time in hours

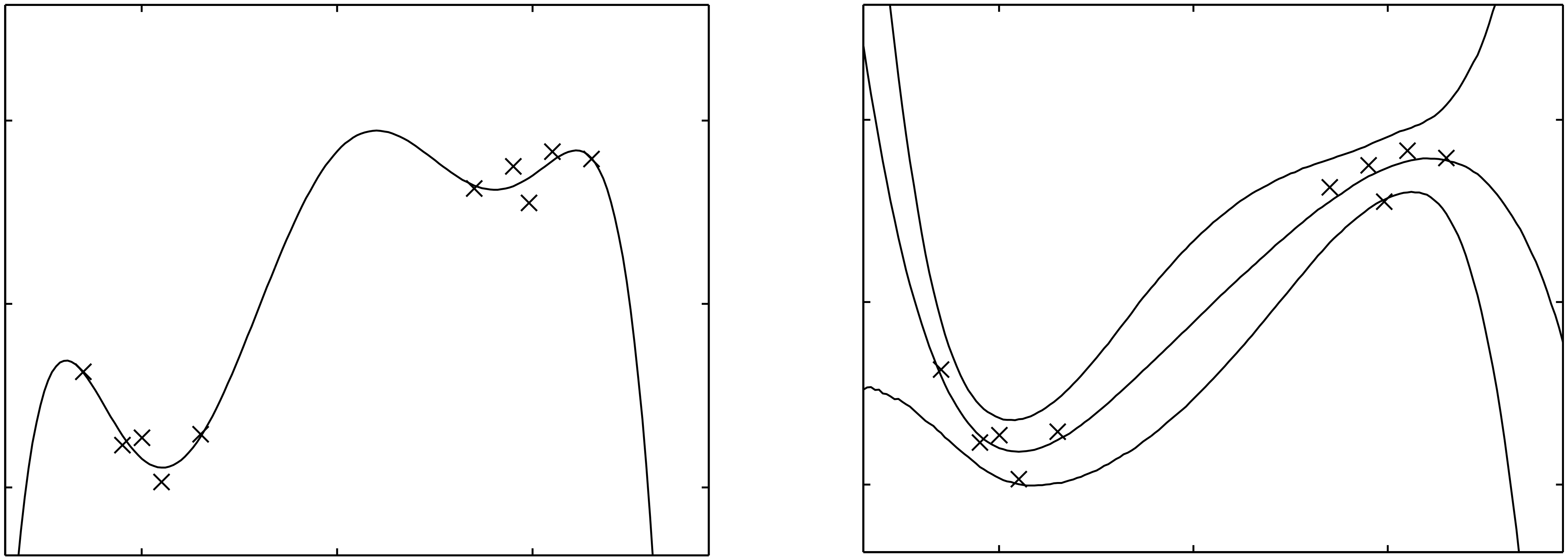


Variational Bayesian Learning

- The main issue in probabilistic machine learning models is to find the posterior distribution over the model parameters and latent variables
- Using a point estimate might overfit
- 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



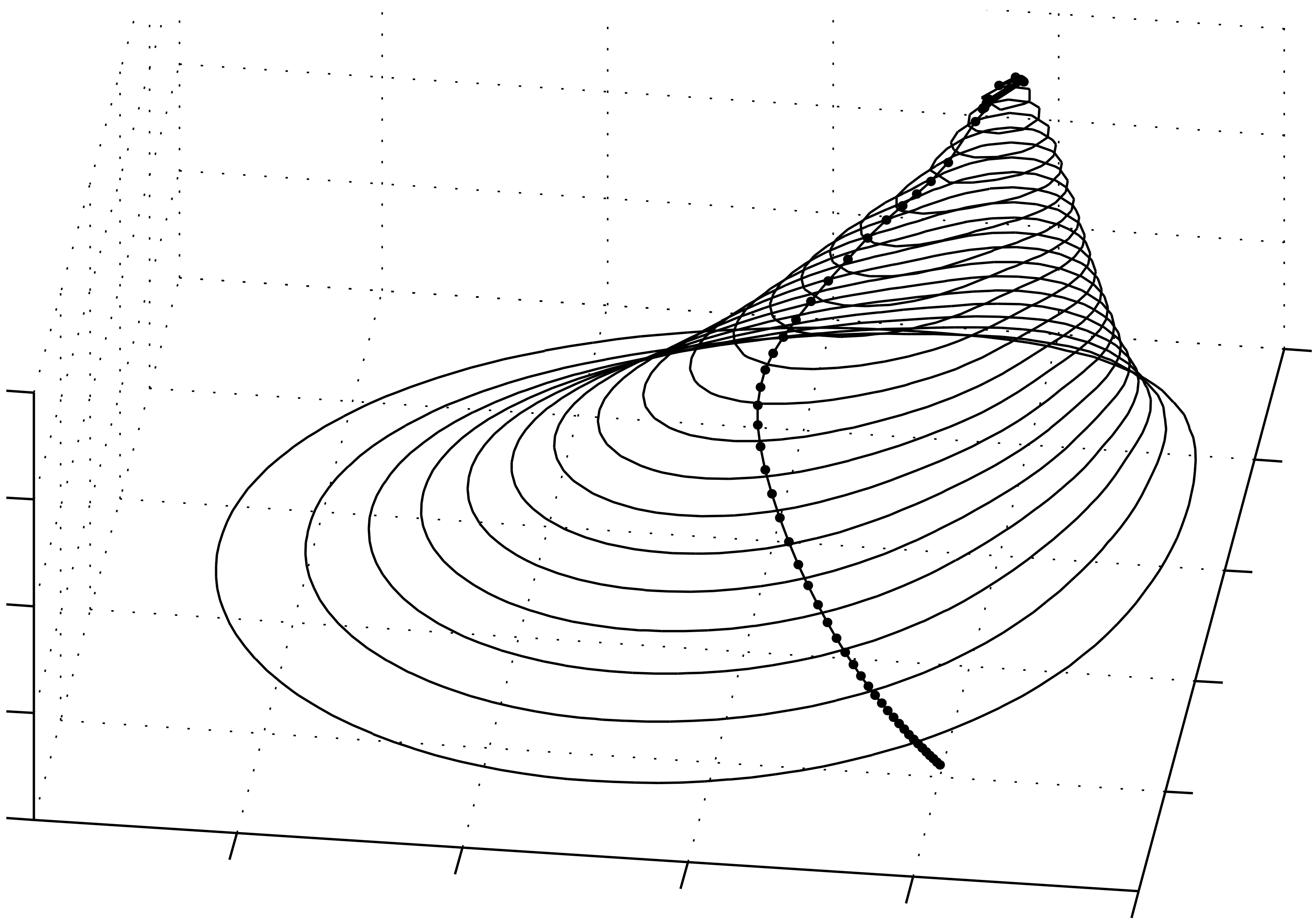
Posterior mass matters

- 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



Variational Bayes

- 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
- KL divergence can also be used for model comparison

VB-EM algorithm

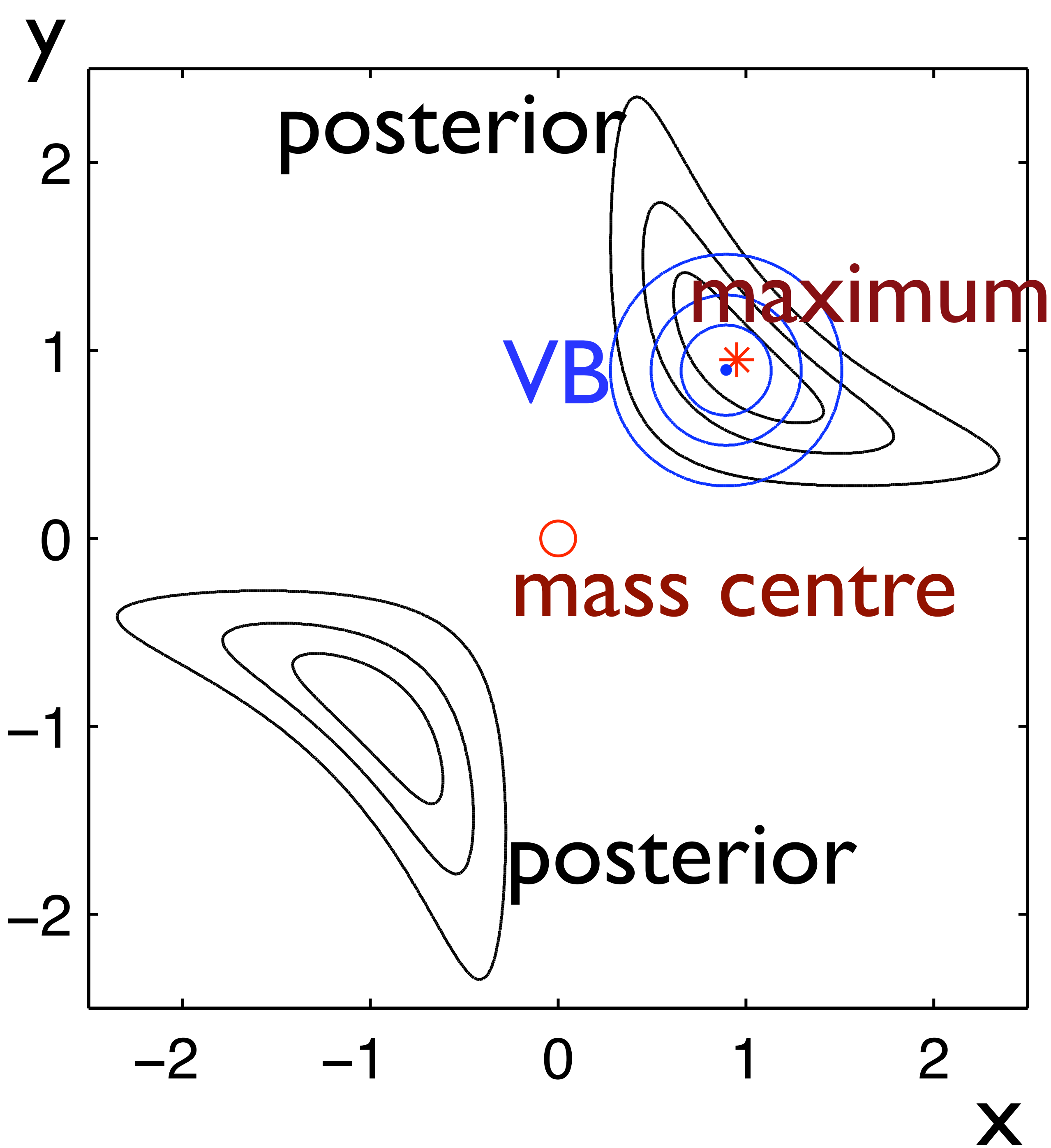
- The VB-EM algorithm alternates between updates for the latent variables and parameters
- Steps are symmetric and they resemble the E-step of the EM algorithm
- VB-E step:

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

- VB-M step:

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

Example 1



- model

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

- prior

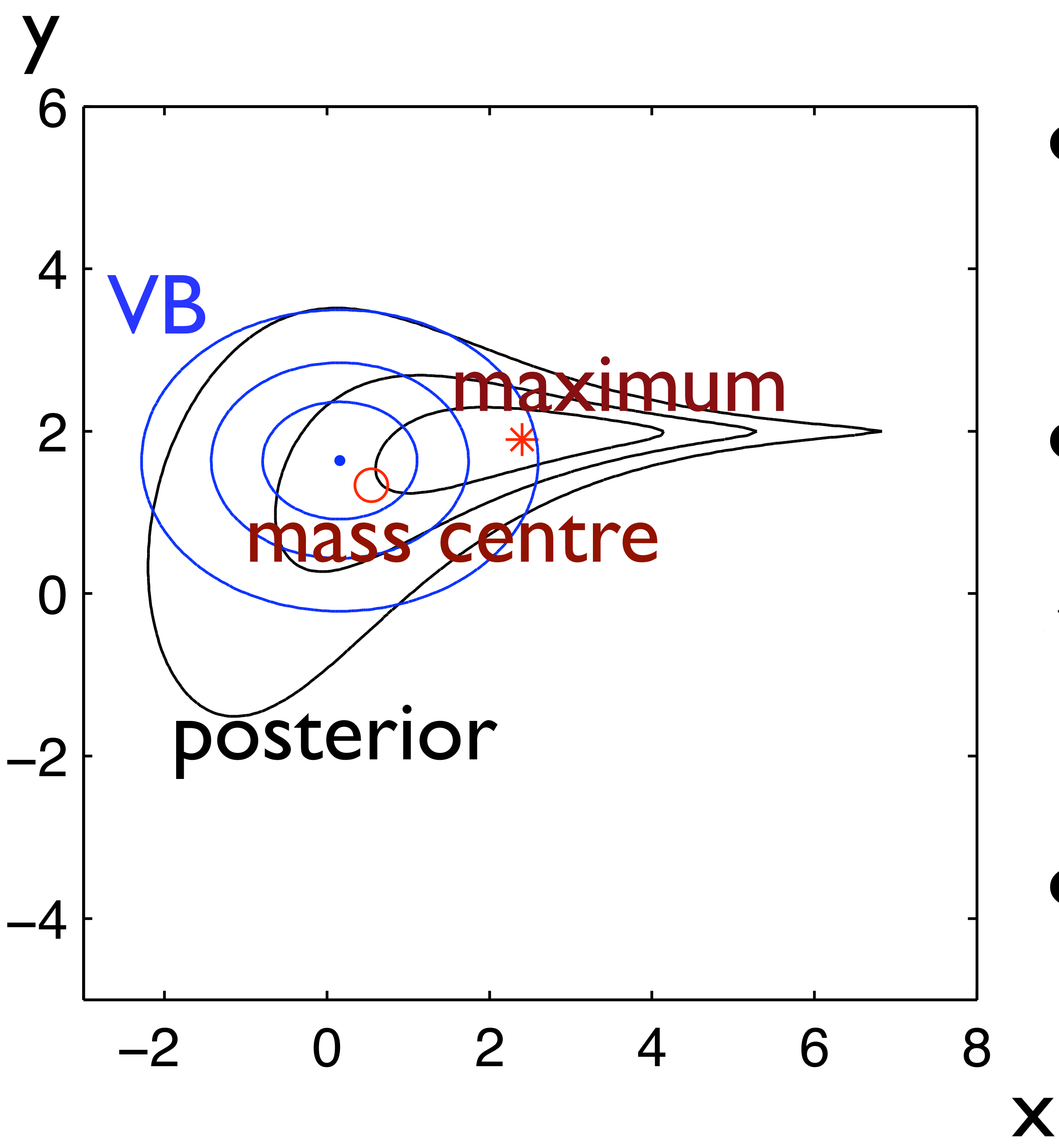
$$p(x) = \mathcal{N}(x; 0, 1),$$

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

- data

$$z = 1$$

Example 2



- model

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

- prior

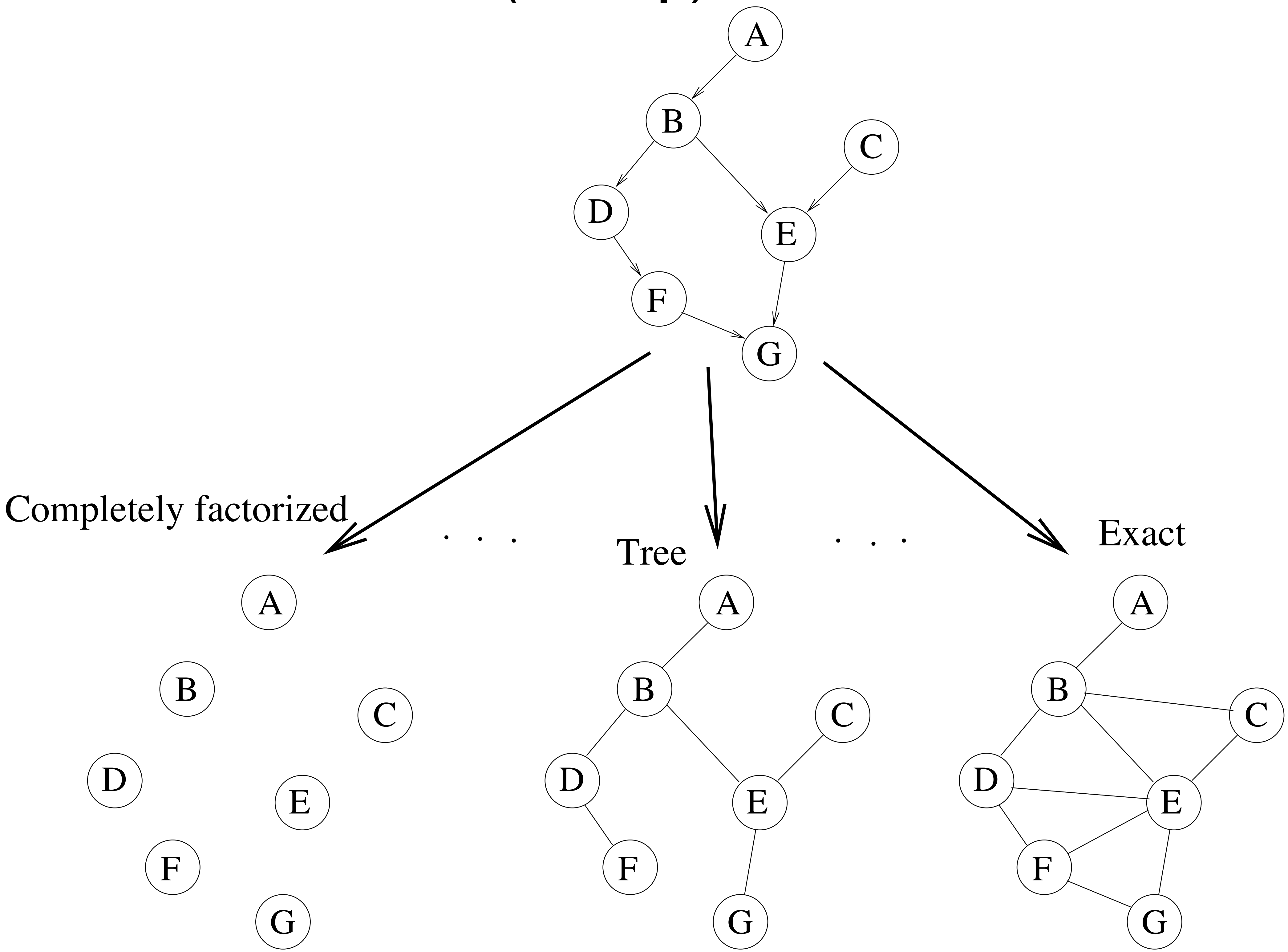
$$p(x) = \mathcal{N}\left(x; -1, 5\right)$$

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

- data

$$z = 2.$$

- 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)

Bayes Blocks Software Package

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