Probabilistic Graphical Models

Lecture 5: Basic Latent Variable Models

Volkan Cevher, Matthias Seeger
Ecole Polytechnique Fédérale de Lausanne

14/10/2011

Outline

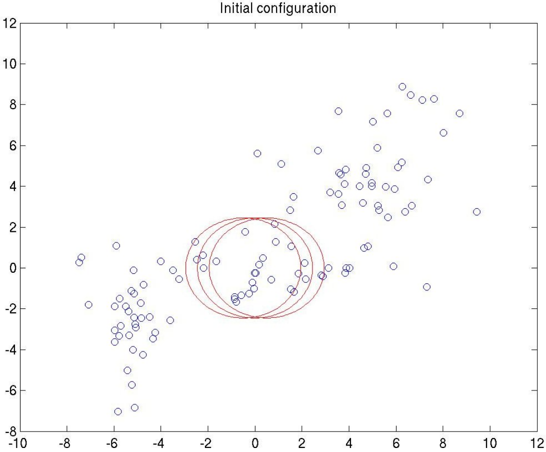
The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
$\Rightarrow$ You are sooo boring. I just use ML estimation.

The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
⇒ You are sooo boring. I just use ML estimation.

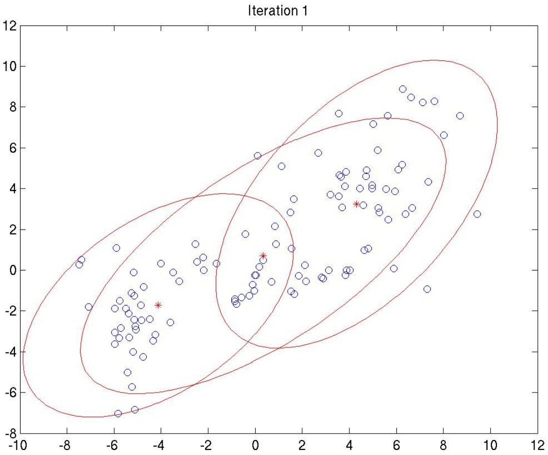
OK, have three Gaussians, don't tell you anything!
⇒ ... ??!?



The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
$\Rightarrow$ You are sooo boring. I just use ML estimation.

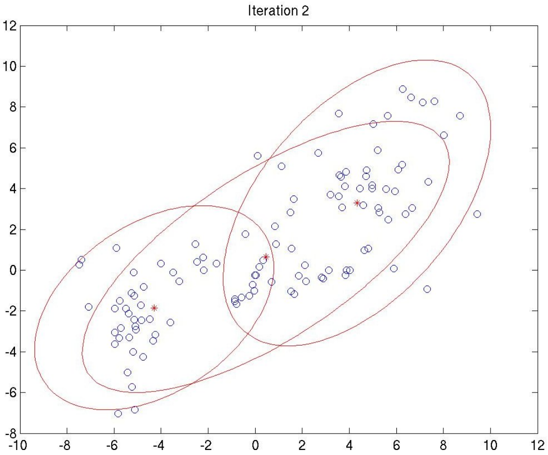
OK, have three Gaussians, don't tell you anything!
$\Rightarrow$ ... ??!?



The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
⇒ You are sooo boring. I just use ML estimation.

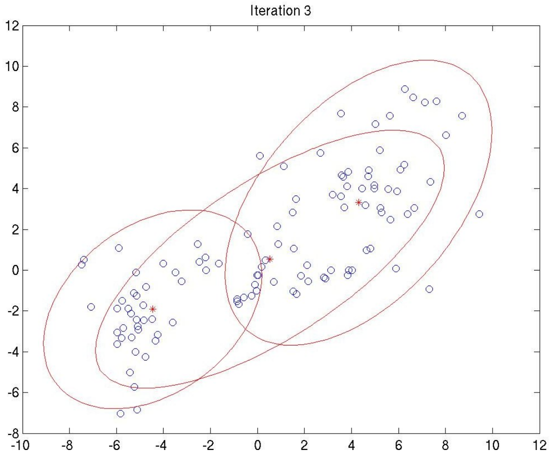
OK, have three Gaussians, don't tell you anything!
⇒ . . . ??!?



The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
$\Rightarrow$ You are sooo boring. I just use ML estimation.

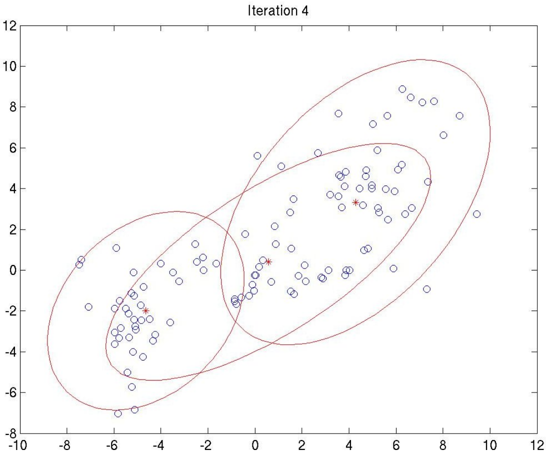
OK, have three Gaussians, don't tell you anything!
$\Rightarrow$ ... ??!?



The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
⇒ You are sooo boring. I just use ML estimation.

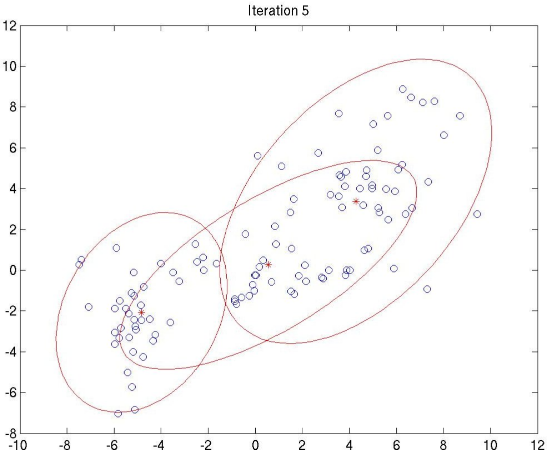
OK, have three Gaussians, don't tell you anything!
⇒ … ??!?



The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
$\Rightarrow$ You are sooo boring. I just use ML estimation.

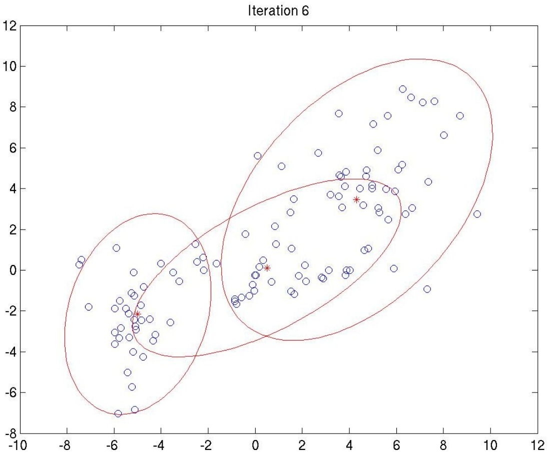
OK, have three Gaussians, don't tell you anything!
$\Rightarrow$ ... ??!?



The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
⇒ You are sooo boring. I just use ML estimation.

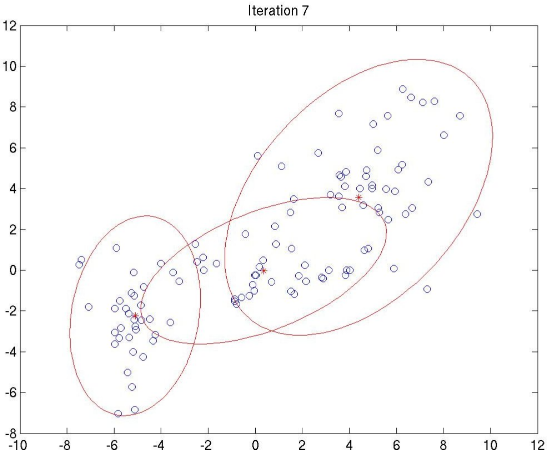
OK, have three Gaussians, don't tell you anything!
⇒ . . . ??!?



The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
⇒ You are sooo boring. I just use ML estimation.

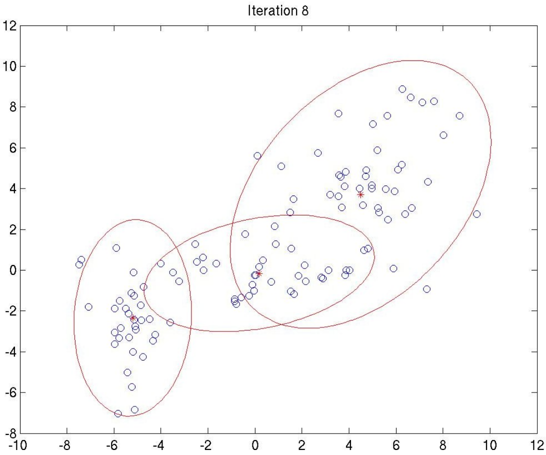
OK, have three Gaussians, don't tell you anything!
⇒ ... ??!?



The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
$\Rightarrow$ You are sooo boring. I just use ML estimation.

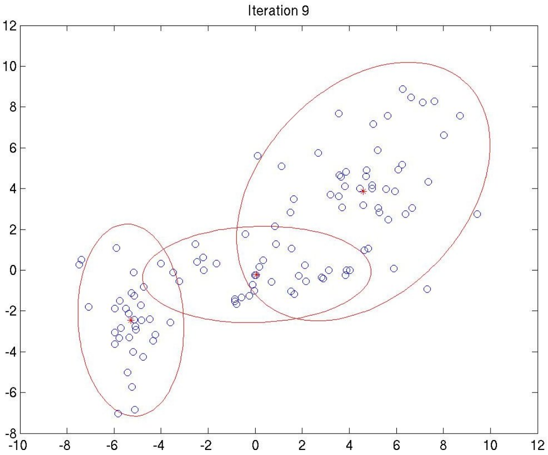
OK, have three Gaussians, don't tell you anything!
$\Rightarrow$ ... ??!?



The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
$\Rightarrow$ You are sooo boring. I just use ML estimation.

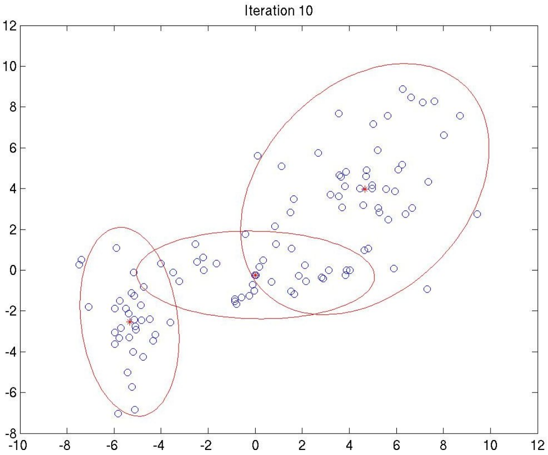
OK, have three Gaussians, don't tell you anything!
$\Rightarrow$ ... ??!?



The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
$\Rightarrow$ You are sooo boring. I just use ML estimation.

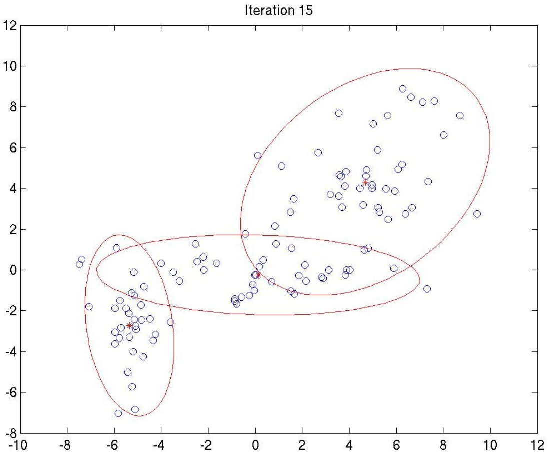
OK, have three Gaussians, don't tell you anything!
$\Rightarrow$ ... ??!?



The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
$\Rightarrow$ You are sooo boring. I just use ML estimation.

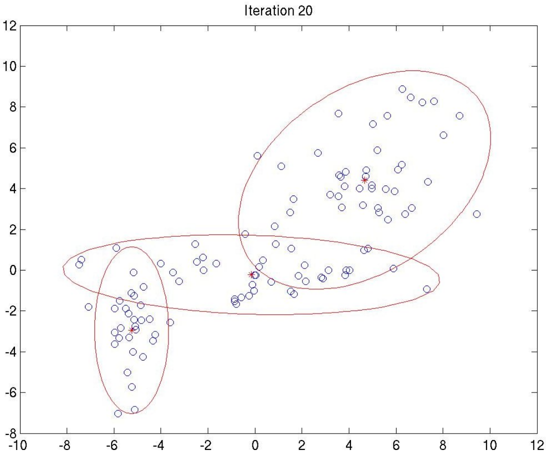
OK, have three Gaussians, don't tell you anything!
$\Rightarrow$ ... ??!?



The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
$\Rightarrow$ You are sooo boring. I just use ML estimation.

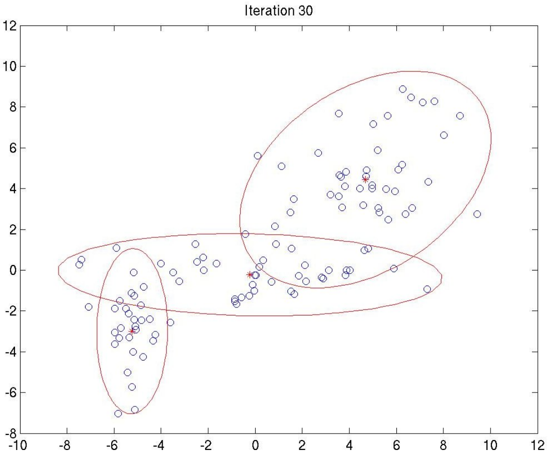
OK, have three Gaussians, don't tell you anything!
$\Rightarrow$ ... ??!?



The Power of Latent Variables

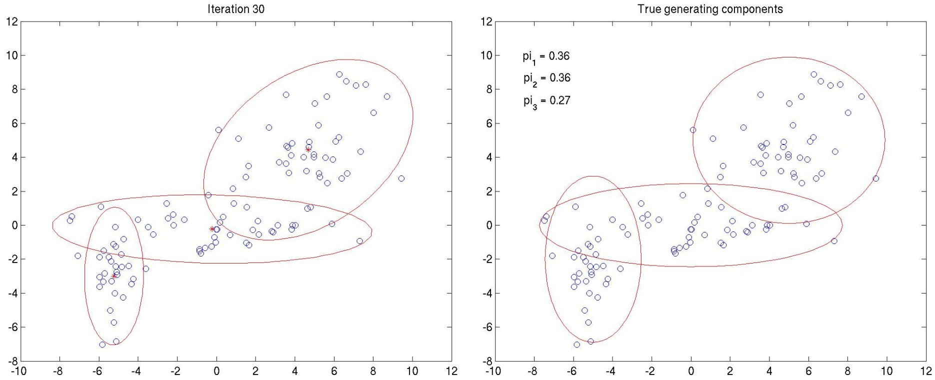
Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
$\Rightarrow$ You are sooo boring. I just use ML estimation.

OK, have three Gaussians, don't tell you anything!
$\Rightarrow$ ... ??!?



The Power of Latent Variables

Have Gaussian, don't tell you mean / covariance. Aetsch-baetsch!
$\Rightarrow$ You are sooo boring. I just use ML estimation.



Latent Variables

Latent variables make models interesting, expressive

- Latent nuisance variables:
  Create complex, realistic distributions from simple ingredients
- Latent query variables:
  Find hidden causes, groupings, explanations in data

Latent Variables

Latent variables make models interesting, expressive

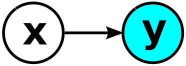
- Latent nuisance variables:
  Create complex, realistic distributions from simple ingredients
- Latent query variables:
  Find hidden causes, groupings, explanations in data

Latent variables need more than estimation. They really need proper inference (marginalization).

Bayesian Handle

- Condition on observed variables
- Marginalize over latent nuisance variables
- Make use of posterior over latent query variables

Vocabulary



- Joint likelihood $P(\boldsymbol{y}, \boldsymbol{x})$
  Typically decomposes (product) according to graph structure
- Marginal likelihood $P(\boldsymbol{y})$

$$P(\boldsymbol{y}) = \int P(\boldsymbol{y}, \boldsymbol{x})\, d\boldsymbol{x}$$

  Typically does not decompose (marginalization creates dependencies)
- Hierarchical model

F3

$$P(\boldsymbol{y}, \boldsymbol{x}, \theta) = P(\boldsymbol{y}|\boldsymbol{x}, \theta)P(\boldsymbol{x}|\theta)P(\theta)$$

  Example: $\boldsymbol{x}$ parameter, θ hyperparameter
  $P(\boldsymbol{x}|\theta)$ prior, $P(\theta)$ hyperprior

KISS: Occam's Razor

- Almost everything can be made latent: Model structure (edges), presence / type of variables (nodes), hierarchies ad infinitum
- Each makes sense for special tasks. But some claim Bayesian statistics should be like that in general.

KISS: Occam's Razor

- Almost everything can be made latent: Model structure (edges), presence / type of variables (nodes), hierarchies ad infinitum
- Each makes sense for special tasks. But some claim Bayesian statistics should be like that in general. I don't.

Occam's Razor

Plurality should not be posited without necessity.
Aka: Keep It Simple, Stupid!

KISS: Occam's Razor

- Almost everything can be made latent: Model structure (edges), presence / type of variables (nodes), hierarchies ad infinitum
- Each makes sense for special tasks. But some claim Bayesian statistics should be like that in general. I don't.

Occam's Razor

Plurality should not be posited without necessity.
Aka: Keep It Simple, Stupid!

KISS if you can:

- You should understand characteristics of your model
- You should (roughly) understand how your inference approximation method behaves. Nobody does that with hyper-complicated models

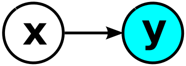
Mixture Models

Humans group, create categories, classify, mostly without any “true labels” existing (think about colours, species, . . .).

Mixture Models

Humans group, create categories, classify, mostly without any "true labels" existing (think about colours, species, ...).

Mixture model:



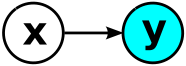
Discrete latent variable $x \in \{1, \dots, K\}$

- $P(\boldsymbol{y}|x)$: Class distribution / mixture component
- $P(x = k) = \pi_k$: Class prior

$$P(\boldsymbol{y}) = \sum_{k=1}^{K} \pi_k P(\boldsymbol{y}|x = k)$$

Mixture Models

Humans group, create categories, classify, mostly without any "true labels" existing (think about colours, species, ...).



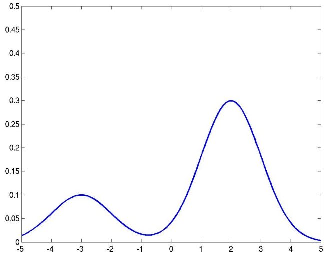
Mixture model:
Discrete latent variable $x \in \{1, \dots, K\}$

- $P(\boldsymbol{y}|x)$: Class distribution / mixture component
- $P(x = k) = \pi_k$: Class prior

$$P(\boldsymbol{y}) = \sum_{k=1}^{K} \pi_k P(\boldsymbol{y}|x = k)$$

Gaussian mixture model:
$P(\boldsymbol{y}|x) = N(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$

- Nuisance x: Used all over the place (whenever Gaussians alone don't work)
- Query x: Clustering, segmentation, classification



Clustering: K-Means

Gaussian mixture model: $P(\boldsymbol{y}|x) = N(\boldsymbol{\mu}_x, \boldsymbol{I})$, $P(x = k) = 1/K$

Observed data: $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n \in \mathbb{R}^d$
Latent indicators: $x_1, \dots, x_n \in \{1, \dots, K\}$

How to find cluster centers $\boldsymbol{\mu}_k$?

Clustering: K-Means

Gaussian mixture model: $P(\boldsymbol{y}|x) = N(\boldsymbol{\mu}_x, \boldsymbol{I})$, $P(x = k) = 1/K$

Observed data: $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n \in \mathbb{R}^d$
Latent indicators: $x_1, \dots, x_n \in \{1, \dots, K\}$

How to find cluster centers $\boldsymbol{\mu}_k$?

Simple Muenchhausen strategy: Iterate F6

1. Each datapoint to closest center F6b

$$x_i \leftarrow \mathrm{argmin}_k \|\boldsymbol{y}_i - \boldsymbol{\mu}_k\| = \mathrm{argmax}_k \, P(x_i = k|\boldsymbol{y}_i)$$

2. Each center: Average of its datapoints

$$\boldsymbol{\mu}_k \leftarrow \left(\sum_{x_i=k} 1\right)^{-1} \sum_{x_i=k} \boldsymbol{y}_i = \mathrm{argmax} \sum_{x_i=k} \log P(\boldsymbol{y}_i|x_i = k)$$

Maximum likelihood if we knew the x_i

The EM Algorithm

Gaussian mixture model: $P(\boldsymbol{y}|x) = N(\boldsymbol{\mu}_x, \boldsymbol{I})$, $P(x = k) = 1/K$

Observed data: $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n \in \mathbb{R}^d$
Latent indicators: $x_1, \dots, x_n \in \{1, \dots, K\}$

How to find cluster centers $\boldsymbol{\mu}_k$?

Fixing K-Means: Iterate

1. Expectation: Posterior distribution for each datapoint

$$Q(x_i = k) \leftarrow P(x_i = k|\boldsymbol{y}_i)$$

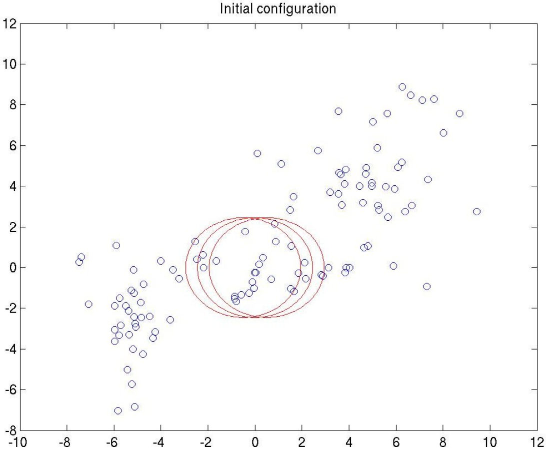
2. Maximization: Posterior average of all datapoints

$$\boldsymbol{\mu}_k \leftarrow n_k^{-1} \sum_i Q(x_i = k)\boldsymbol{y}_i = \text{argmax} \sum_i Q(x_i = k) \log P(\boldsymbol{y}_i|x_i = k),$$

$n_k = \sum_i Q(x_i = k)$. Posterior weighted maximum likelihood

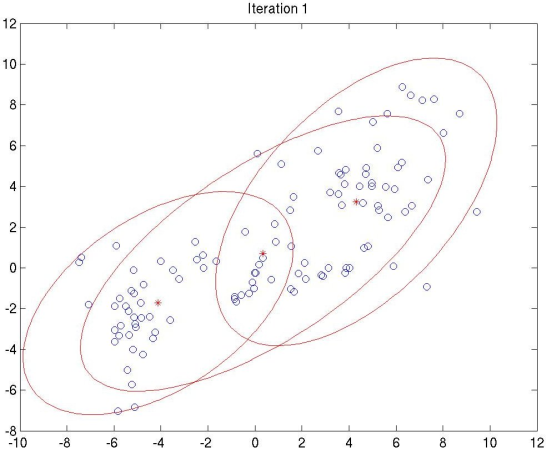
The EM Algorithm

EM in action



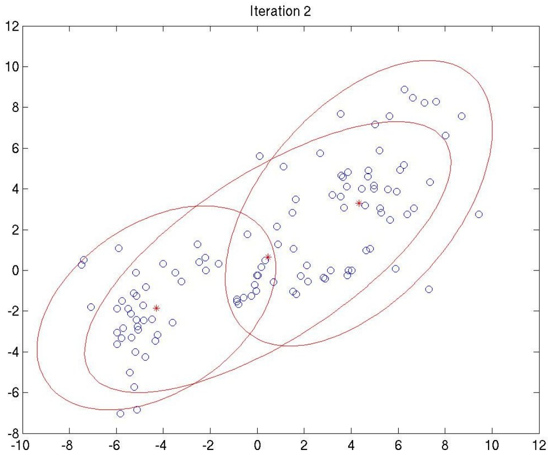
The EM Algorithm

EM in action



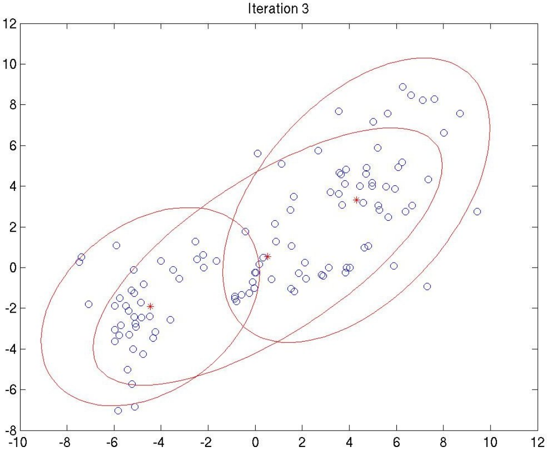
The EM Algorithm

EM in action



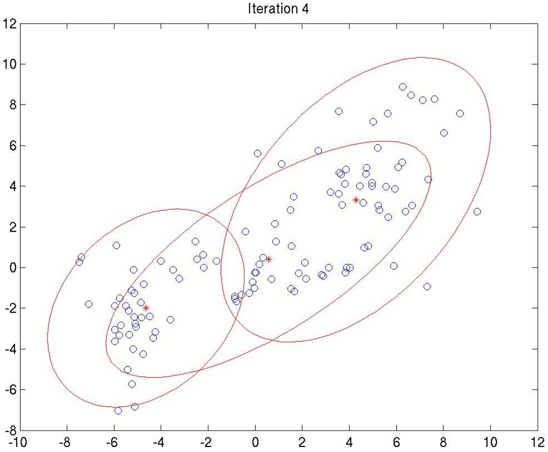
The EM Algorithm

EM in action



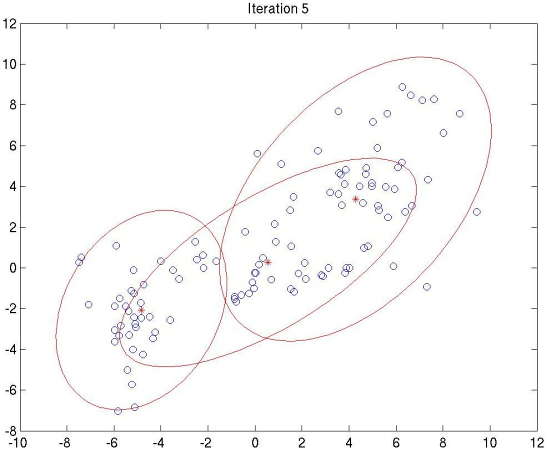
The EM Algorithm

EM in action



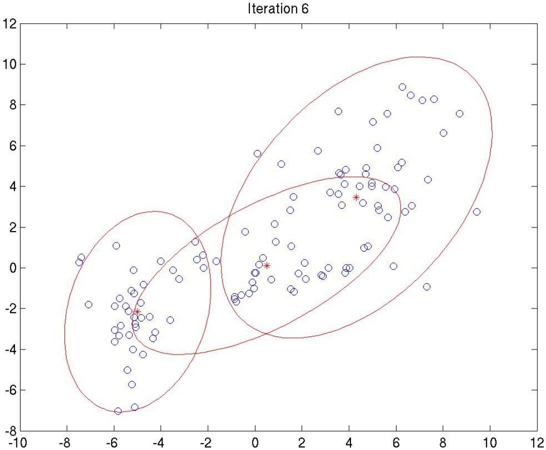
The EM Algorithm

EM in action



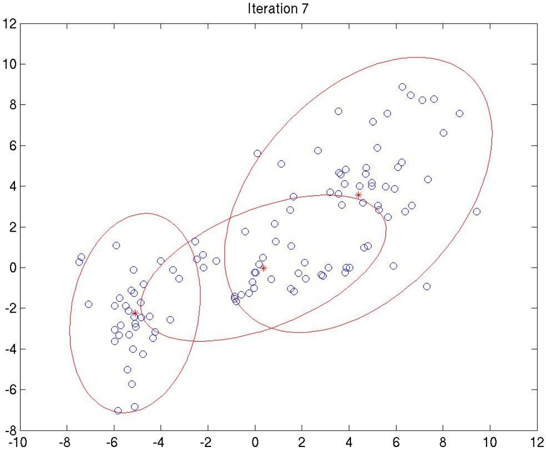
The EM Algorithm

EM in action



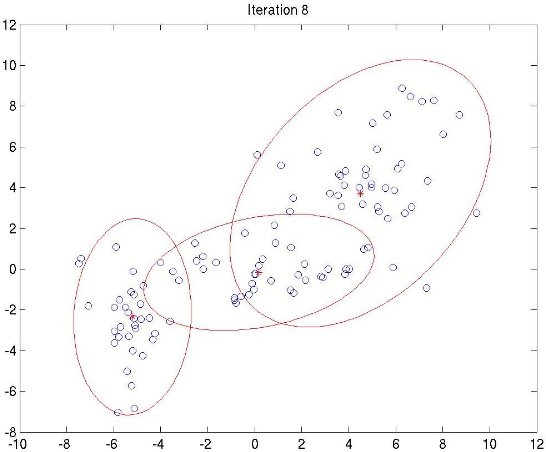
The EM Algorithm

EM in action



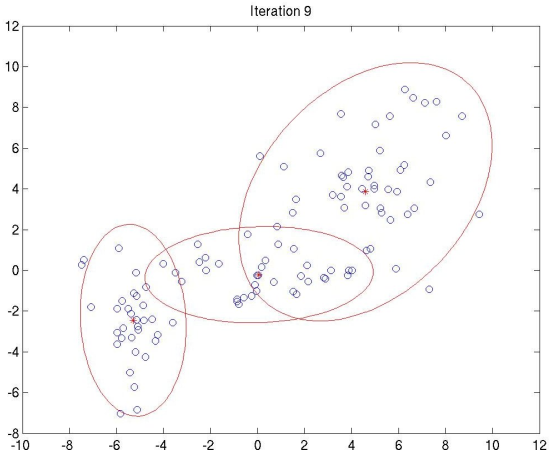
The EM Algorithm

EM in action



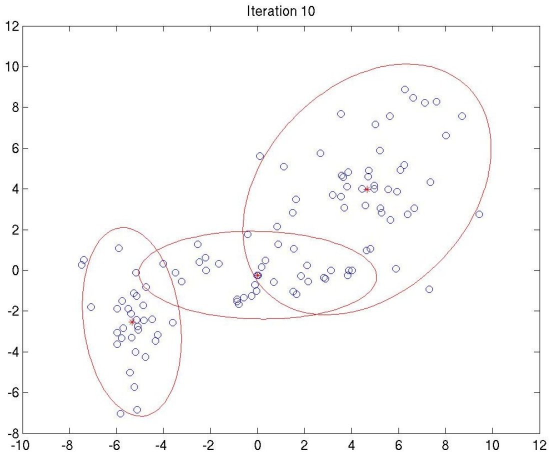
The EM Algorithm

EM in action



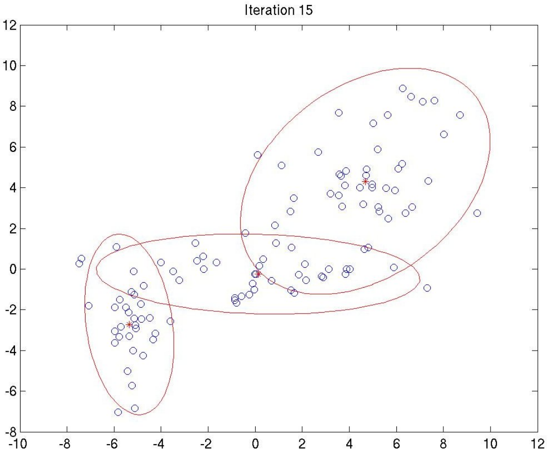
The EM Algorithm

EM in action



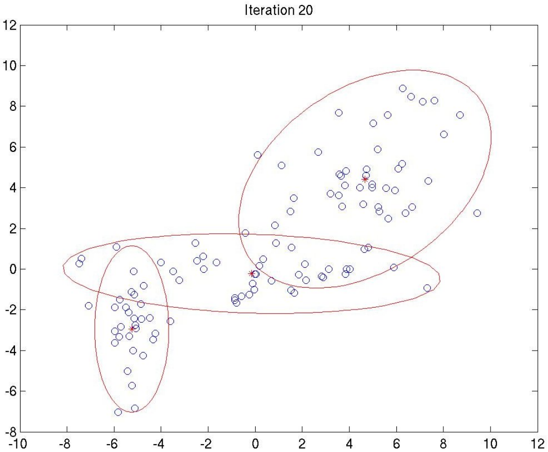
The EM Algorithm

EM in action



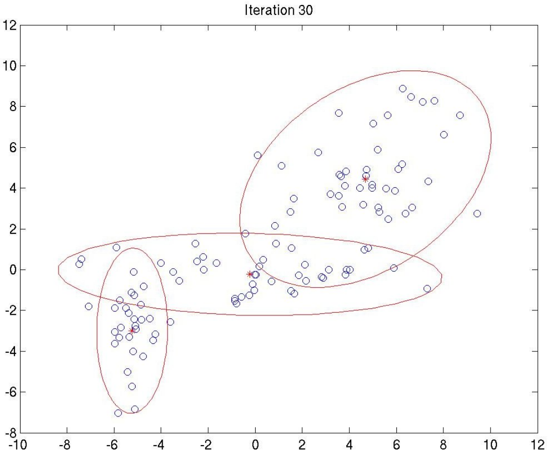
The EM Algorithm

EM in action

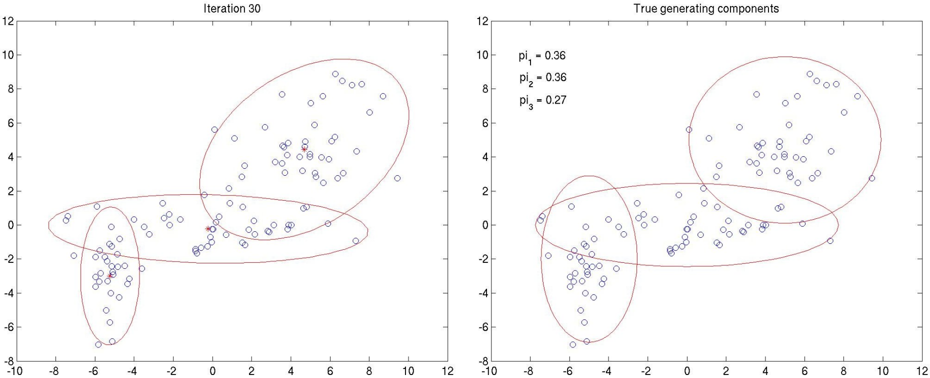


The EM Algorithm

EM in action



The EM Algorithm



For $P(\boldsymbol{y}|x) = N(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$:
No new idea, weighted ML update for $\boldsymbol{\Sigma}_k$ as well F8

Some Pointers

- How do I choose K if nobody tells me?
  Example of model selection.
  Bayesian possibility: $D = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_n\}$ F9
  - Determine marginal likelihood "high up"

$$\log P(D|K) = \log \int \prod_i \sum_k \pi_k(\boldsymbol{\theta}_K) P(\boldsymbol{y}_i | x_i = k, \boldsymbol{\theta}_K)\, d\boldsymbol{\theta}_K$$

    $\boldsymbol{\theta}_K$: Parameters for K-component model
  - Pick $K_* = \mathrm{argmax}_K \log P(D|K)$

  Problem: Hard to approximate. Workable approaches exist.
  Note: Chop this down $\to$ BIC, AIC, ...

Some Pointers

- How do I choose K if nobody tells me?
  Example of model selection.
  Bayesian possibility: $D = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_n\}$
  - Determine marginal likelihood "high up"

$$\log P(D|K) = \log \int \prod_i \sum_k \pi_k(\boldsymbol{\theta}_K) P(\boldsymbol{y}_i | x_i = k, \boldsymbol{\theta}_K)\, d\boldsymbol{\theta}_K$$

    $\boldsymbol{\theta}_K$: Parameters for K-component model
  - Pick $K_* = \mathrm{argmax}_K \log P(D|K)$

  Problem: Hard to approximate. Workable approaches exist.
  Note: Chop this down $\rightarrow$ BIC, AIC, ...
- Do I have to choose K at all? Can't it be nuisance latent?
  Nonparametric Bayesian methods: F9b
  - Prior ranging over mixture models of all component numbers K
  - Idea: Marginalize over K as well
  - Hard to do this right in practice, especially with Gaussian mixtures

Problem with Gaussian Models

- Gaussians: Too restrictive for real-world data
  $\Rightarrow$ Gaussian mixture models, ...

Problem with Gaussian Models

- Gaussians: Too restrictive for real-world data
  $\Rightarrow$ Gaussian mixture models, ...
- Gaussians: Too flexible for real-world data
  - In $\mathbb{R}^n$: Covariance has $\approx n^2/2$ parameters
    $\Rightarrow$ Cannot fit all from limited data [curse of dimensionality]
  - Even with enough data: Application might demand fast computation
  - Latent query: Want to discover stable causes

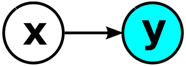
Problem with Gaussian Models

- Gaussians: Too restrictive for real-world data
  $\Rightarrow$ Gaussian mixture models, ...
- Gaussians: Too flexible for real-world data
  - In $\mathbb{R}^n$: Covariance has $\approx n^2/2$ parameters
    $\Rightarrow$ Cannot fit all from limited data [curse of dimensionality]
  - Even with enough data: Application might demand fast computation
  - Latent query: Want to discover stable causes

$\Rightarrow$ "Pancake models"

Pancake Models

Pancake model (aka. latent Gaussian model) F11

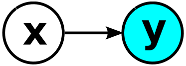


$$\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{W}\boldsymbol{x} + \varepsilon, \quad \boldsymbol{x} \sim N(\mathbf{0}, \boldsymbol{I}), \quad \varepsilon \sim N(\mathbf{0}, \Psi)$$

$\boldsymbol{W} \in \mathbb{R}^{d,p}$ Factor loadings ($p \ll d$)
$\boldsymbol{x} \in \mathbb{R}^{p}$ Latent (Gaussian) factors (degrees of variation)

Pancake Models

Pancake model (aka. latent Gaussian model)



$$\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{W}\boldsymbol{x} + \varepsilon, \quad \boldsymbol{x} \sim N(\boldsymbol{0}, \boldsymbol{I}), \quad \varepsilon \sim N(\boldsymbol{0}, \Psi)$$

$\boldsymbol{W} \in \mathbb{R}^{d,p}$ Factor loadings ($p \ll d$)

$\boldsymbol{x} \in \mathbb{R}^{p}$ Latent (Gaussian) factors (degrees of variation)

Probabilistic PCA

$$\Psi = \sigma^2 \boldsymbol{I}$$

- Maximum likelihood estimate: PCA (as you know it)!

Tipping, Bishop, 99

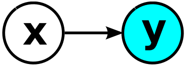
Factor Analysis

Ψ diagonal

- P-PCA is special case F11b
- Used heavily in psychometrics, social sciences, marketing “science”
- Maximum likelihood estimate: No closed form in general

Pancake Models

Pancake model (aka. latent Gaussian model)



$$\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{W}\boldsymbol{x} + \boldsymbol{\varepsilon}, \quad \boldsymbol{x} \sim N(\boldsymbol{0}, \boldsymbol{I}), \quad \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \boldsymbol{\Psi})$$

$\boldsymbol{W} \in \mathbb{R}^{d,p}$ Factor loadings ($p \ll d$)
$\boldsymbol{x} \in \mathbb{R}^{p}$ Latent (Gaussian) factors (degrees of variation)

Probabilistic PCA

$\boldsymbol{\Psi} = \sigma^2 \boldsymbol{I}$

- Maximum likelihood estimate: PCA (as you know it)!

Tipping, Bishop, 99

Independent CA (done right)

x_i independent, not Gaussian

- We'll come to a special case

Factor Analysis

$\boldsymbol{\Psi}$ diagonal

- P-PCA is special case
- Used heavily in psychometrics, social sciences, marketing "science"
- Maximum likelihood estimate: No closed form in general

Probabilistic PCA

$$\boldsymbol{y} = \mu + \boldsymbol{W}\boldsymbol{x} + \varepsilon, \quad \boldsymbol{x} \sim N(\boldsymbol{0}, \boldsymbol{I}), \quad \varepsilon \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$$

$$\boldsymbol{Y} = [\boldsymbol{y}_1 - \mu | \dots | \boldsymbol{y}_n - \mu], \quad \hat{\boldsymbol{S}} = n^{-1} \boldsymbol{Y}\boldsymbol{Y}^T$$

Tipping, Bishop (1999):
Maximum likelihood estimate of $\boldsymbol{W}$: Leading eigenvectors of $\hat{\boldsymbol{S}}$
$\Rightarrow$ Just standard PCA! F12

Factor Analysis

$$\boldsymbol{y} = \mu + \boldsymbol{W}\boldsymbol{x} + \varepsilon, \quad \boldsymbol{x} \sim N(\boldsymbol{0}, \boldsymbol{I}), \quad \varepsilon \sim N(\boldsymbol{0}, \boldsymbol{\Psi}), \ \boldsymbol{\Psi} \text{ diagonal}$$

Maximum likelihood: No closed-form estimator known

$\Rightarrow$ Have to use EM algorithm (Muenchhausen with pancakes)

- Expectation: $Q(\boldsymbol{x}_i) = P(\boldsymbol{x}_i|\boldsymbol{y}_i) = N(\boldsymbol{x}_i|?)$
- Maximization: Posterior weighted average
  $\boldsymbol{W} \leftarrow ?,\ \boldsymbol{\Psi} \leftarrow ?$

You'll do that in the exercises.

Density Estimation in High Dimensions

We learned about

1. Gaussian mixture models
2. Factor analysis / P-PCA

Density Estimation in High Dimensions

We learned about

1. Gaussian mixture models
2. Factor analysis / P-PCA

Combine them: Mixture of Factor Analysers (sic):
One of most powerful general-purpose density models

- Speech recognition (often, $\boldsymbol{W}_x = \mathbf{0}$)
- Probabilistic robotics
- Bio-Informatics (microarray data)
- Hand-written digits (MLers love them, don't ask why)

Good fitting not simple. But there are useful heuristic methods available.

The Naming Game

What do Boltzmann Machines, Products of Experts, Conditional Random Fields have in common?

- They are all fancy names

The Naming Game

What do Boltzmann Machines, Products of Experts, Conditional Random Fields have in common?

- They are all fancy names
- They are all the same (more or less): Markov random fields

$$P(\boldsymbol{x}) = Z^{-1} \prod_j \Phi_j(\boldsymbol{x}_{C_j}), \quad Z = \sum_{\boldsymbol{x}} \prod_j \Phi_j(\boldsymbol{x}_{C_j})$$

The Naming Game

What do Boltzmann Machines, Products of Experts, Conditional Random Fields have in common?

- They are all fancy names
- They are all the same (more or less): Markov random fields

$$P(\boldsymbol{x}) = Z^{-1} \prod_j \Phi_j(\boldsymbol{x}_{C_j}), \quad Z = \sum_{\boldsymbol{x}} \prod_j \Phi_j(\boldsymbol{x}_{C_j})$$

- They come with different graph structure / potential parameterization, so algorithms seem different.
  Trust me: They are not.
- Positive side:
  New approximations, applications, cross-fertilization. New views on old things

The Naming Game

What do Boltzmann Machines, Products of Experts, Conditional Random Fields have in common?

- They are all fancy names
- They are all the same (more or less): Markov random fields

$$P(\boldsymbol{x}) = Z^{-1} \prod_j \Phi_j(\boldsymbol{x}_{C_j}), \quad Z = \sum_{\boldsymbol{x}} \prod_j \Phi_j(\boldsymbol{x}_{C_j})$$

- They come with different graph structure / potential parameterization, so algorithms seem different.
  Trust me: They are not.
- Positive side:
  New approximations, applications, cross-fertilization. New views on old things
- We'll see how to learn MRFs in next lecture (related to EM)

The Boltzmann Machine

$$P(\boldsymbol{x}) = Z^{-1} e^{-E(\boldsymbol{x})/T}, \quad E(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{W} \boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x}$$

A Gaussian?

The Boltzmann Machine

$$P(\boldsymbol{x}) = Z^{-1} e^{-E(\boldsymbol{x})/T}, \quad E(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{W}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}$$

A Gaussian? No: $x_i \in \{\pm 1\}$ (binary spins)

Boltzmann (1844-1906), founded stat. mechanics / thermodynamics

$\boldsymbol{x}$	State (of system)
$E(\boldsymbol{x})$	Energy
$\boldsymbol{W}$	Weight / coupling matrix, $\boldsymbol{W}^T = \boldsymbol{W}$, $\mathrm{diag}^{-1}(\boldsymbol{W}) = \boldsymbol{0}$
T	Temperature

$\Rightarrow$ Comes from Ising model, but emphasis on learning $\boldsymbol{W}$.

The Boltzmann Machine

$$P(\boldsymbol{x}) = Z^{-1} e^{-E(\boldsymbol{x})/T}, \quad E(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{W}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}$$

A Gaussian? No: $x_i \in \{\pm 1\}$ (binary spins)

Boltzmann (1844-1906), founded stat. mechanics / thermodynamics

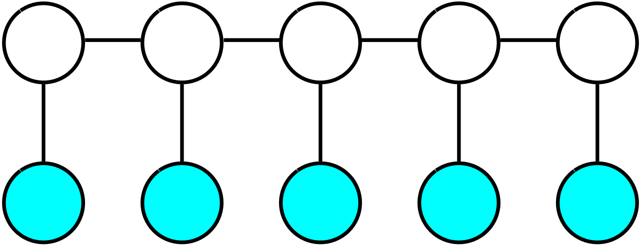
$\boldsymbol{x}$	State (of system)
$E(\boldsymbol{x})$	Energy
$\boldsymbol{W}$	Weight / coupling matrix, $\boldsymbol{W}^T = \boldsymbol{W}$, $\mathrm{diag}^{-1}(\boldsymbol{W}) = \boldsymbol{0}$
T	Temperature

$\Rightarrow$ Comes from Ising model, but emphasis on learning $\boldsymbol{W}$.

"Conversion" into MRF:

$$C_{ij} = \{i, j\},\ i < j,\ w_{ij} \neq 0, \quad C_i = \{i\},$$
$$\Phi_{ij}(C_{ij}) = e^{-w_{ij}x_i x_j/T}, \quad \Phi_i(C_i) = e^{b_i x_i/T}$$

Conditional Random Fields



- Undirected cousin of Hidden Markov Model [all that: lecture +2]
- Underlying graph: chain $\Rightarrow$ Inference, learning simple.
  Can be done on very large datasets
- Heavily used in applications for text, language, WWW information

Gaussian Markov Random Fields

- Gaussian with sparse, structured inverse covariance matrix $\boldsymbol{A} = \Sigma^{-1}$ (aka. precision matrix) [No edge $(ij) \Leftrightarrow a_{ij} = 0$]
- Used for spatial / spatiotemporal data, also for images
- Posterior mean computations in $O(n)$:
  Conjugate gradients, loopy belief propagation [part II]
- Modern approaches: Algorithms from numerical mathematics, convergent belief propagation for preconditioning

Gaussian Markov Random Fields

- Gaussian with sparse, structured inverse covariance matrix $\boldsymbol{A} = \Sigma^{-1}$ (aka. precision matrix) [No edge $(ij) \Leftrightarrow a_{ij} = 0$]
- Used for spatial / spatiotemporal data, also for images
- Posterior mean computations in $O(n)$:
  Conjugate gradients, loopy belief propagation [part II]
- Modern approaches: Algorithms from numerical mathematics, convergent belief propagation for preconditioning
- Fundamentally different from Gaussian process models:
  $P(\boldsymbol{x}_I)$ does not have precision matrix $\boldsymbol{A}_I$
  (but $(\boldsymbol{A}/\boldsymbol{A}_{\setminus I})^{-1}$, as we've learned)

Wrap-Up

- Latent variables: Salt in modelling soup
- Mixtures: Grouping, clustering, classification
- Latent Gaussian “pancake” models:
  Economical parameterization in high dimensions
- Markov random fields come in many disguises
- Next lecture: Inference and learning (why EM works)