Expectation Maximization

Maschinelles Lernen 1 -
Grundverfahren WS20/21

Prof. Gerhard Neumann
KIT, Institut für Anthrophomatik und Robotik

Learning Outcomes

What will we learn today?

- Understand latent variable models and why they are hard to train
- Understand mixture models and how to train it using EM
- Analysis of the EM algorithm and why it converges

Agenda for today

Mixture Models

- Gaussian Mixture Models (GMM)
- Expectation Maximization

Latent Variable Models and Generalized EM

- EM decomposition
- E- and M-step
- Convergence analysis
- EM for dimensionality reduction

Mixture Models

Parametric models

- Gaussian, Neural Networks, ...
- ✓ Good analytic properties
- ✓ Simple
- ✓ Small memory requirements
- ✓ Fast
- × Limited representation power (most parametric distributions have only one mode)

Non-Parametric models

- Kernel-density estimation, k-NN
- ✓ General (can represent any distribution)
- × Curse of dimensionality
- × Large memory requirements
- × Slow

- Mixture models combine the advantages of both worlds
- **Key idea:** Create a complex distribution by combining simple ones (e.g. Gaussians)

Mixture model

A mixture distribution is the sum of individual distributions:

Number of components

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} p(k)p(\boldsymbol{x}|k)$$

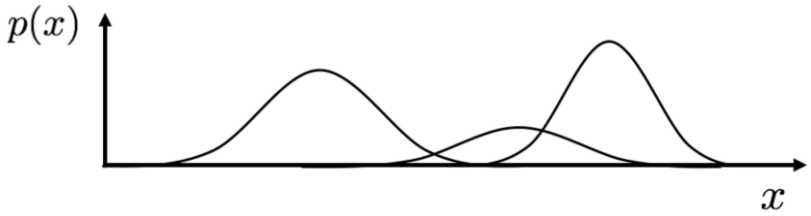
Mixture coefficient

k-th mixture component

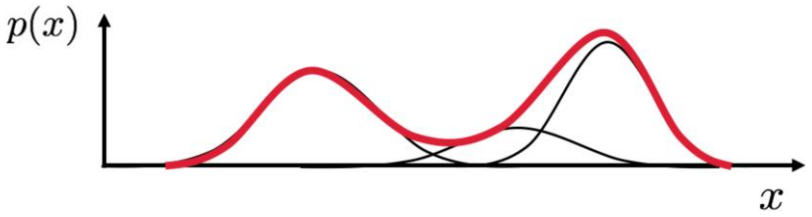
- In the limit with many / infinite components, this can approximate any smooth density

Example: Mixture of Gaussians (MoG)

- Individual Gaussians



- Sum of Gaussians



Gaussian Mixture Models (GMMs)

- **Mixture coefficient:**

$$p(k) = \pi_k, \text{ with } 0 \leq \pi_k \leq 1, \sum_k \pi_k = 1$$

- **Mixture component:**

$$p(\boldsymbol{x}|k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- **Mixture distribution:**

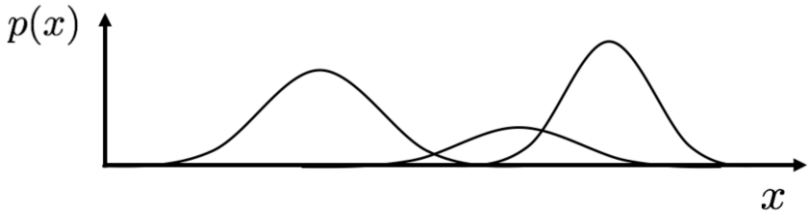
$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

  - Always integrates to 1
  - Parameters of the mixture

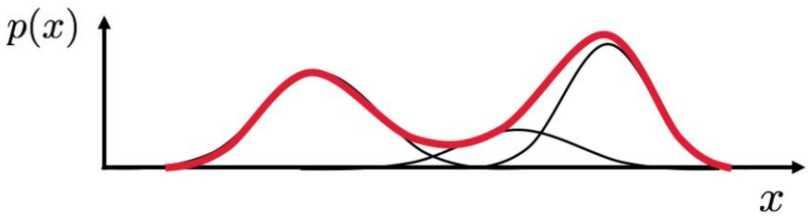
$$\boldsymbol{\theta} = \{\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_K, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K\}$$

Example: Mixture of Gaussians (MoG)

- Individual Gaussians



- Sum of Gaussians



Maximum Likelihood of a mixture

- (Marginal-)Log-Likelihood with N iid. points

$$\mathcal{L} = \log L(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}_{\text{marginal}} = \sum_{i=1}^{N} \log \underbrace{\left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)}_{\text{non-exponential family}}$$

- Q: Can we do gradient descent?

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_j} &= \sum_{i=1}^{N} \frac{\pi_j \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j) \\
&= \sum_{i=1}^{N} \frac{p(j) p(\boldsymbol{x}_i | j)}{p(\boldsymbol{x}_i)} \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j) \\
&= \sum_{i=1}^{N} \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j) p(j | \boldsymbol{x}_i)
\end{aligned}$$

× Gradient depends on all other components (cyclic dependency)

× No closed form solution

× Typically very slow convergence

► A: Yes, but the sum (marginalization) does not go well with the log

EM for Gaussian Mixture Models

Estimating Gaussian Mixture Models

So why is optimizing $\mathcal{L} = \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ **so hard?**

- Because we do not know which mixture component k created which data-point
- If we would have data from the joint distribution $p(\boldsymbol{x}_i, k_i|\boldsymbol{\theta})$, then it would be easy ...

In this case, we can simply perform a maximum likelihood estimate:

- Coefficients: $\pi_k = \frac{\sum_i q_{ik}}{N}$
- Means: $\boldsymbol{\mu}_k = \frac{\sum_i q_{ik} \boldsymbol{x}_i}{\sum_i q_{ik}}$
- Covariances: $\boldsymbol{\Sigma}_k = \frac{\sum_i q_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T}{\sum_i q_{ik}}$

where $q_{ik} = \mathbb{I}(k, k_i)$ is 1 if the ith sample belongs to the kth component and 0 otherwise

Expectation-Maximization: Chicken and Egg...

Yet, we do not know which component belongs to which sample

- Can we estimate that? Given a current mixture model, yes!

- **Expectation Step:**
  - Compute cluster probabilities aka responsibilities for each sample (Bayes rule)

$$q_{ik} = p(k_i = k|\boldsymbol{x}_i) = \frac{p(\boldsymbol{x}_i|k)p(k)}{\sum_j p(\boldsymbol{x}_i|j)p(j)} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

  - Responsibilities q_{ik} are now continues between 0 and 1
  - But we need to know the Gaussian components

Expectation-Maximization: Chicken and Egg...

Yet, we do not know which component belongs to which sample

- Can we estimate that? Given a current mixture model, yes!

- **Maximization Step:**
  - Compute (weighted) maximum likelihood estimate

$$\pi_k = \frac{\sum_i q_{ik}}{N} \qquad \boldsymbol{\mu}_k = \frac{\sum_i q_{ik}\boldsymbol{x}_i}{\sum_i q_{ik}} \qquad \boldsymbol{\Sigma}_k = \frac{\sum_i q_{ik}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T}{\sum_i q_{ik}}$$

  - But we need to know the responsibilities

Algorithm: EM for GMMs

- **Initialize:** Mixture Components + Mixture coefficients
  - E.g. Use k-means for the component means and some initial covariance

- **Repeat** until convergence**:**
  - **Expectation-step:** Compute responsibilities

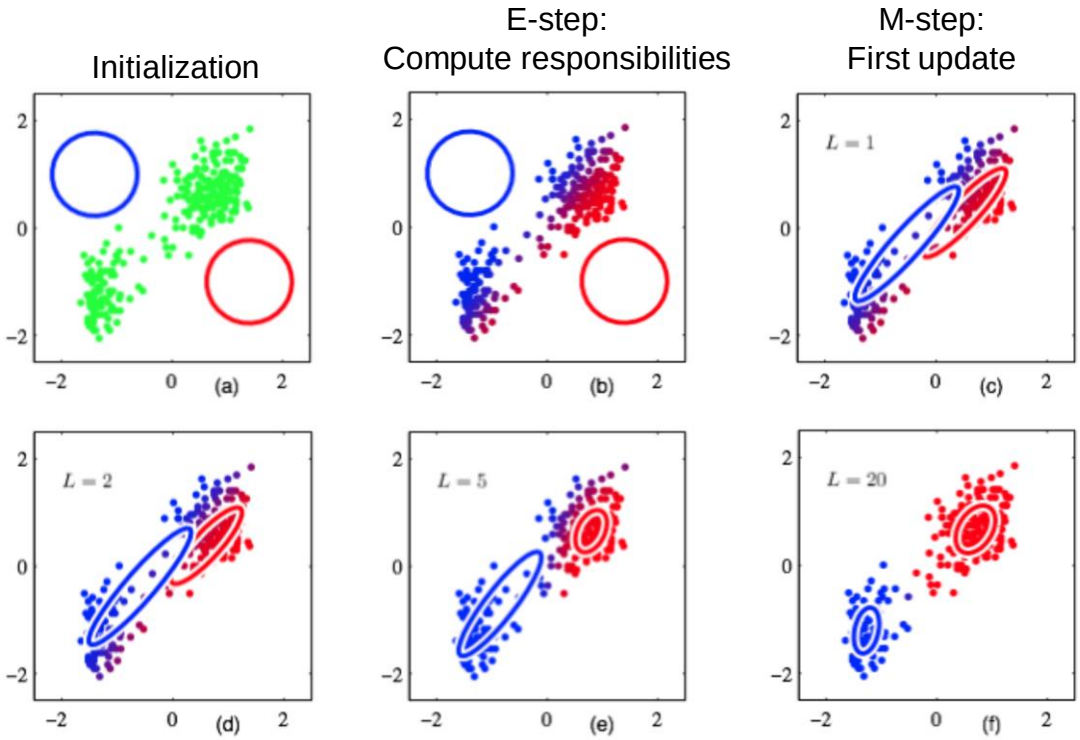
$$q_{ik} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

  - **Maximization-step:** Update coefficients, components means and component variance

$$\pi_k = \frac{\sum_i q_{ik}}{N} \qquad \boldsymbol{\mu}_k = \frac{\sum_i q_{ik} \boldsymbol{x}_i}{\sum_i q_{ik}} \qquad \boldsymbol{\Sigma}_k = \frac{\sum_i q_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T}{\sum_i q_{ik}}$$

Illustration

- Each component represents a cluster in the data set
- EM is very sensitive to the initialization



EM versus k-means

K-means can be seen as special case of EM with:

- Co-variances are always set to 0 (in the limit)
- **E-Step / Assignment Step:**
  - responsibilities q_{ik} of nearest cluster k are set to 1, all other values are 0

- **M-Step / Adjustment Step:**
  - Update for the mean is the same
  - Co-Variances are ignored (set to close to 0)

- EM is harder to learn than k-means but it also gives you variances and densities
- Often k-means is used to initialize the means for EM

K-means is known to converge, does also EM always converge?

EM for Gaussian Mixture Models

- **Mixture coefficient:**

$$p(k) = \pi_k, \text{ with } 0 \leq \pi_k \leq 1, \sum_k \pi_k = 1$$

- **Mixture component:**

$$p(\boldsymbol{x}|k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- **Mixture distribution:**

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

  - Always integrates to 1
  - Parameters of the mixture

$$\boldsymbol{\theta} = \{\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_K, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K\}$$

E-Step

- Compute “responsibilities”

$$q_{ik} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = p(z = k|\boldsymbol{x}_i)$$

- How much component k contributes to generation of $\mathbf{x}_i$ according to current mixture model

EM for Gaussian Mixture Models

- **Mixture coefficient:**

$$p(k) = \pi_k, \text{ with } 0 \leq \pi_k \leq 1, \sum_k \pi_k = 1$$

- **Mixture component:**

$$p(\boldsymbol{x}|k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- **Mixture distribution:**

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

  - Always integrates to 1
  - Parameters of the mixture

$$\boldsymbol{\theta} = \{\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_K, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K\}$$

M-Step: $$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} \sum_i \sum_k q_{ik} \log p(k) p(\boldsymbol{x}_i|k)$$

- We can separate updates of single components and coefficients
  - just additive objectives in lower bound
- **Update coefficients:**

$$\boldsymbol{\pi} = \arg\max_{\boldsymbol{\pi}} \sum_i \sum_k q_{ik} \log \pi_k$$

- **Update components:**

$$\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k = \arg\max_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k} \sum_i q_{ik} \log \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

  - Each data-point is weighted by q_{ik}
  - Weighted maximum likelihood estimate

EM for Gaussian Mixture Models

Weighted Maximum Likelihood updates:

- **Update coefficients:** $\boldsymbol{\pi} = \arg\max_{\boldsymbol{\pi}} \sum_i \sum_k q_{ik} \log \pi_k$
  - Result: $\pi_k = \frac{\sum_i q_{ik}}{\sum_k \sum_i q_{ik}} = \frac{\sum_i q_{ik}}{N}$
- **Update components:** $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k = \arg\max_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k} \sum_i q_{ik} \log \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
  - Mean: $\boldsymbol{\mu}_k = \frac{\sum_i q_{ik} \boldsymbol{x}_i}{\sum_i q_{ik}}$
  - Covariance: $\boldsymbol{\Sigma}_k = \frac{\sum_i q_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T}{\sum_i q_{ik}}$

Agenda for today

Mixture Models

- Gaussian Mixture Models (GMM)

The Expectation Maximization (EM) Algorithm

- EM decomposition
- E- and M-step
- Convergence analysis
- EM for GMMs

Latent Variable Models and a generalized view on EM

Mixture Models are Latent Variable Models

Mixture models are an instance of latent variable models

- Examples: mixture models, missing data, latent factors,
- Observed variables: **x,** Latent variables: **z** (e.g., index of mixture component)
- Parametric model: $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$
- Marginal distribution: $\underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{z} p(\boldsymbol{x}, z),}_{\text{discrete latent variable}} \quad \underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int_{\boldsymbol{z}} p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}}_{\text{continuous latent variable}}$

(Marginal) Log-Likelihood:

$$\mathrm{L}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{i=1}^{N} \log \left(\sum_{z} p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, z) \right)$$

… which is hard to optimize for all latent variable models (due to log of a sum)

Expectation-Maximization (EM)

Expectation-Maximization (EM) is a general algorithm for estimating latent variable models

- **Most common application:** Gaussian Mixture models
- … but many other (deep) models as well
- Its extension is called Variational Bayes, which is underlying variational auto-encoder and other variational inference techniques
- Very hot research topic… pays off to look into the math of it

EM can be derived in 2 ways:

- Jensen's inequality (not covered)
- Decomposition in lower-bound and KL-term

Expectation-Maximization (EM)

EM uses a lower bound of the marginal log-likelihood for the optimization

- For simplicity, lets consider only a single data-point first

$$\underbrace{\log p(\boldsymbol{x}|\boldsymbol{\theta})}_{\text{marginal log-like}} = \underbrace{\sum_z q(z) \log \frac{p(\boldsymbol{x}, z|\boldsymbol{\theta})}{q(z)}}_{\text{Lower Bound } \mathcal{L}(q,\boldsymbol{\theta})} + \underbrace{\sum_z q(z) \log \frac{q(z)}{p(z|\boldsymbol{x})}}_{\text{KL Divergence: KL}(q(z)||p(z|\boldsymbol{x}))}$$

- Where $q(z)$ is called the variational / auxiliary distribution
  - This decomposition holds for any $q(z)$
  - By introducing q(z), the optimization will become much simpler
- Why is that the same?
  - We can use Bayes rule for $p(z|\boldsymbol{x}) = \frac{p(\boldsymbol{x}, z|\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{\theta})}$ and all terms except $p(\boldsymbol{x}|\boldsymbol{\theta})$ cancel

Basics: Kullback-Leibler Divergences

The KL-divergence is a important similarity measure for distributions

$$\mathrm{KL}(q(\boldsymbol{x})||p(\boldsymbol{x})) = \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}$$

- **Its always non-negative** $\mathrm{KL}(q||p) \geq 0$
- **If its zero, both distributions are the same:** $\mathrm{KL}(q||p) = 0 \iff q = p$
- **It is non-symmetric** (hence, its not a distance metric): $\mathrm{KL}(q||p) \neq \mathrm{KL}(p||q)$

- Can be used to find different approximations of distributions
- Used a lot in Variational Inference, Reinforcement Learning, Information theory…

EM-Decomposition

Derivation:

$$\log p(\boldsymbol{x}) = \sum_z q(z) \log p(\boldsymbol{x})$$

1. Introduce variational distribution q(z)

$$= \sum_z q(\boldsymbol{z}) \big(\log p(\boldsymbol{x}, z) - \log p(\boldsymbol{z}|\boldsymbol{x}) \big)$$

2. Use Bayesian theorem $p(\boldsymbol{x}) = \dfrac{p(\boldsymbol{x}, z)}{p(z|\boldsymbol{x})}$

$$= \sum_z q(z) \big(\log p(\boldsymbol{x}, z) - \log q(z) + \log q(z) - \log p(\boldsymbol{z}|x) \big)$$

3. Add and subtract $\log q(z)$

$$= \underbrace{\sum_z q(z) \log \frac{p(\boldsymbol{x}, z)}{q(z)}}_{\text{Lower Bound } \mathcal{L}(q)} + \underbrace{\sum_z q(z) \log \frac{q(z)}{p(z|\boldsymbol{x})}}_{\text{KL Divergence: } \mathrm{KL}(q(z)||p(z|\boldsymbol{x}))}$$

4. Write as 2 sums

EM Decomposition

Marginal Likelihood decomposes in 2 terms: $\log p(\boldsymbol{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + \mathrm{KL}(q(z)||p(z|\boldsymbol{x}))$

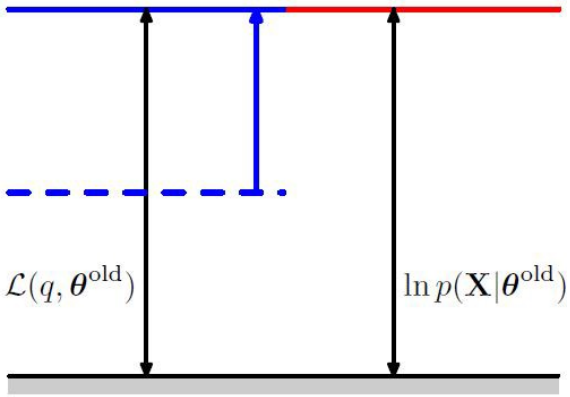
- **Lower bound** $\mathcal{L}(q, \boldsymbol{\theta}) = \sum_z q(z) \log p(\boldsymbol{x}, z|\boldsymbol{\theta}) - \sum_z q(z) \log q(z)$
  - Contains $\log p(\boldsymbol{x}, z|\boldsymbol{\theta})$ instead of $\log p(\boldsymbol{x}|\boldsymbol{\theta}) = \log \sum_z p(\boldsymbol{x}, z|\boldsymbol{\theta})$
  - … which is much easier to optimize (convex for most distributions)
  - Each $\log p(\boldsymbol{x}, z|\boldsymbol{\theta})$ is weighted by *q(z)*
- Why is it a lower bound?
  - Since $\mathrm{KL}(q||p) \geq 0$ it follows that $\mathcal{L}(q, \boldsymbol{\theta}) \leq \log p(\boldsymbol{x}|\boldsymbol{\theta})$

Expectation-Maximization Steps

EM iteratively applies 2 steps:

- **(E)xpectation-step:** $q(z) = \arg\min_q \mathrm{KL}(q(z)||p(z|\boldsymbol{x}))$
  - Find q(z) that minimizes KL
  - Can be done in closed form for discrete z (e.g. mixtures):

$$q(z) = p(z|\boldsymbol{x}, \boldsymbol{\theta}_{\text{old}}) = \frac{p(\boldsymbol{x}, z|\boldsymbol{\theta}_{\text{old}})}{\sum_z p(\boldsymbol{x}, z|\boldsymbol{\theta}_{\text{old}})}$$



- **Observations:**
  - The marginal log-likelihood $\log p(\boldsymbol{x}|\boldsymbol{\theta})$ is unaffected by the E-step
  - As KL is minimized, lower bound has to go up
  - After the E-step $\mathrm{KL}(q(z)||p(z|\boldsymbol{x})) = 0$ and therefore, the lower bound is tight, i.e.:

$$\log p(\boldsymbol{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta})$$

Expectation-Maximization Steps

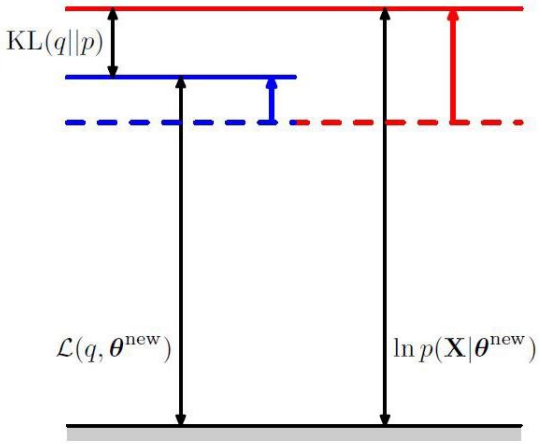
EM iteratively applies 2 steps:

- **(M)aximization-step:**

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(q, \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{z} q(z) \log p(\boldsymbol{x}, z|\boldsymbol{\theta}) + \text{const}$$

  - Maximize lower bound with respect to $\boldsymbol{\theta}$
  - Also called the complete-data likelihood
  - Each possible value of the missing data is weighted by

$$q(z) = p(z|\boldsymbol{x}, \boldsymbol{\theta}_{\text{old}})$$



EM Convergence Properties

- **EM improves the lower bound**

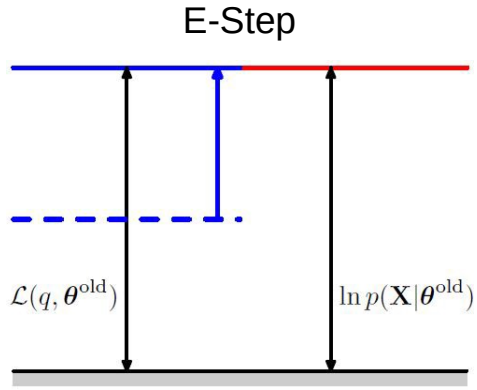
$$\mathcal{L}(q_{\text{new}}, \boldsymbol{\theta}_{\text{new}}) \geq \mathcal{L}(q_{\text{old}}, \boldsymbol{\theta}_{\text{old}})$$

  - M-step: Lower bound is maximized
  - E-step: KL is set to 0, lower bound has to go up

- **EM improves the marginal likelihood**

$$\log p(\boldsymbol{x}|\boldsymbol{\theta}_{\text{new}}) \geq \log p(\boldsymbol{x}|\boldsymbol{\theta}_{\text{old}})$$

  - M-step: Lower bound increases and KL increases (can't get smaller than 0)
  - E-step: Marginal likelihood is unaffected



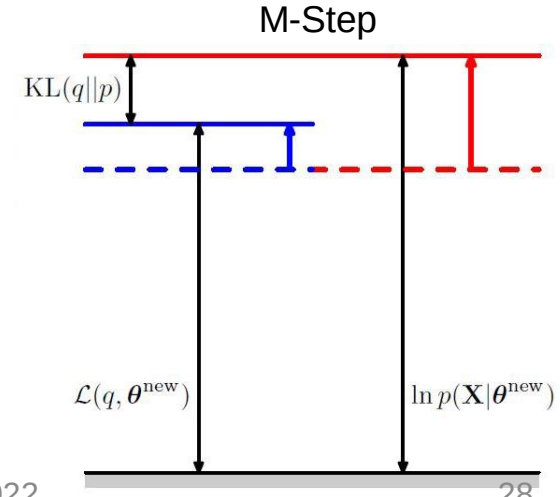
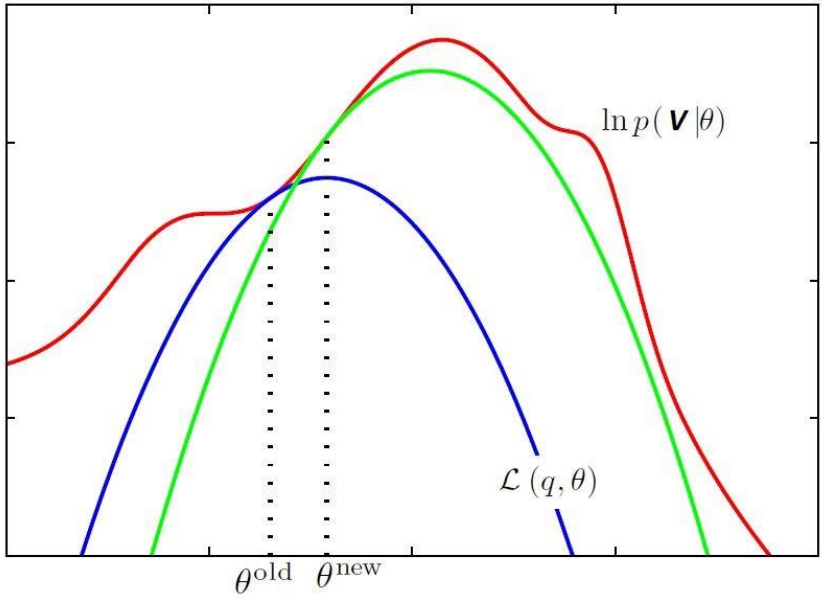


Illustration of EM

- Lower bound (blue curve) is a convex approximation of the marginal likelihood (red curve)
  - Maximum of lower bound can be easily obtained ($\boldsymbol{\theta}^{\text{new}}$)
  - Closed form solutions available, no gradient descent required
- Compute new lower bound for $\boldsymbol{\theta}^{\text{new}}$ (green curve)
- Due to the local approximation of the lower-bound, EM can only find local optima



EM for full dataset

For all data-points, the lower bound is given by:

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_i \left(\int_z q_i(z) \log p(\boldsymbol{x}_i, z|\boldsymbol{\theta}) dz - \int_z q_i(z) \log q_i(z) dz \right)$$

- One latent variable z_i per data-point
- If z is discrete with K different values, than

$$q_i(z = k) = p(z = k|\boldsymbol{x}_i, \boldsymbol{\theta}_{\text{old}})$$

  can be represented as a N x K matrix
- We will write $q_{ik} = q_i(z = k)$

Practical considerations…

How many mixture components do we need?

- More components will typically lead to a better likelihood
- But are more components necessarily better? Not always, because of overfitting!
- It's again a model-selection problem (cross-validate on a validation-set)
- Bayesian methods can be used to integrate out number of components (tricky to get them to work)

How do we initialize:

- EM can give very poor results with wrong initialization
- Most common approach:
  - Use k-means (simple clustering algorithm) to initialize the centers
  - Use a fixed value for the covariance

EM for Dimensionality Reduction

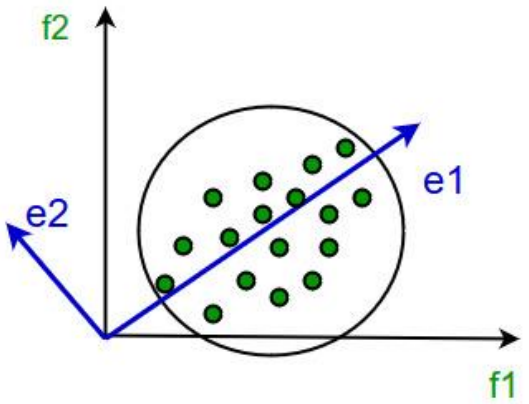
EM for Dimensionality Reduction (aka. probabilistic PCA)

We can also formulate dimensionality reduction using latent variables

Idea: Introduce a latent variable model to relate a D-dimensional observation vector to a corresponding M-dimensional gaussian latent variable (with M < D)

$$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$$

- $\boldsymbol{z}$ is a d' latent variable (our low dimensional representation)
- $\boldsymbol{W}$ is a *D x M m*atrix relating the latent space $\boldsymbol{z}$ with the original space $\boldsymbol{x}$
- $\boldsymbol{\mu}$ is a constant offset vector
- $\boldsymbol{\epsilon}$ is a d-dimensional Gaussian noise vector $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$



EM for Dimensionality Reduction

Probabilistic Dimensionality Reduction Model:

$$\boldsymbol{z} \in \mathbb{R}^M, \quad \boldsymbol{x} \in \mathbb{R}^D, \quad M < D$$

- **Continuous Latent Variable:**

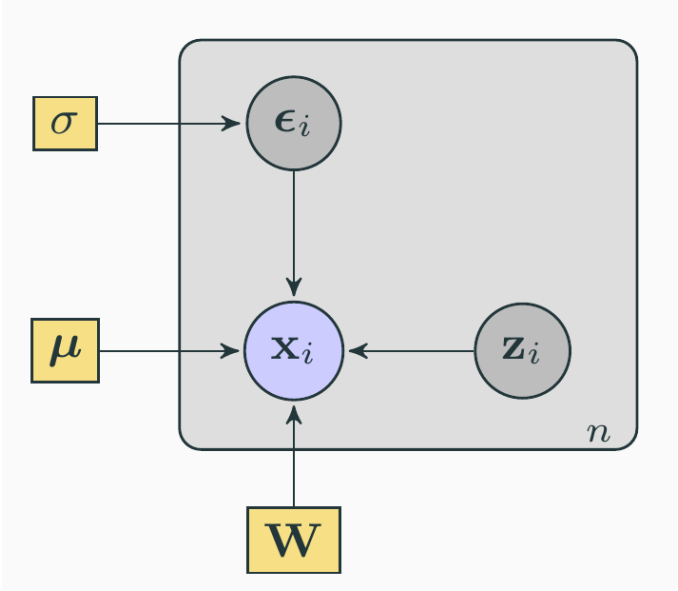
$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

  - Assume 0 mean, unit variance distribution in latent space

- **Observation Model**

$$p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$$

  - with parameters $\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{\mu}, \sigma^2\}$



Generative Process

Our model can be interpreted in terms of a generative process

1. Sample latent variable

$$\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

2. Linearly project to high-D space

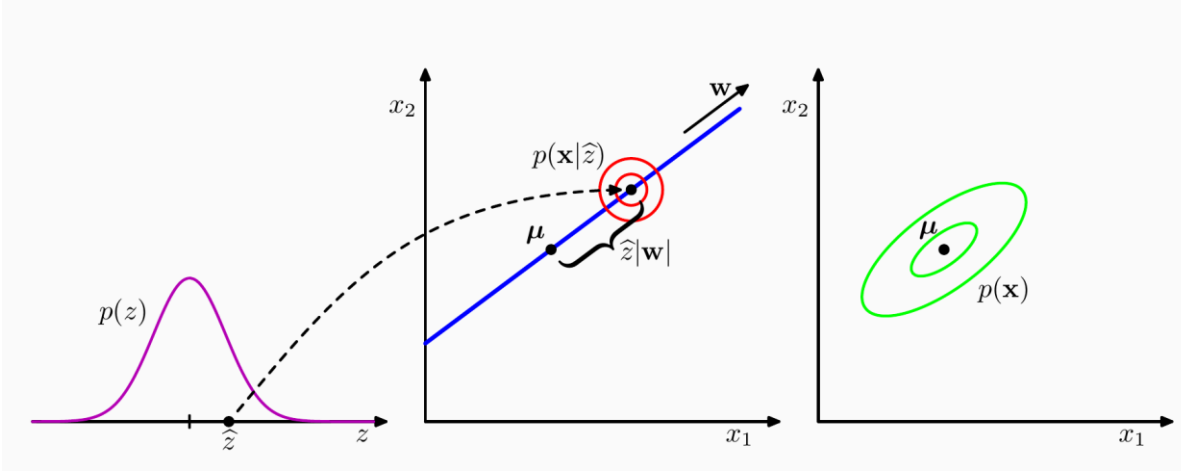
$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}$$

3. Sample noise

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$$

4. Add noise to obtain **x**

$$\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{\epsilon}$$



Marginal likelihood

Marginal likelihood is given by:

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \int_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\theta})p(\boldsymbol{z})d\boldsymbol{z} = \int_{\boldsymbol{z}} \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z}+\boldsymbol{\mu},\sigma^2\boldsymbol{I})\mathcal{N}(\boldsymbol{z}|\boldsymbol{0},\boldsymbol{I})d\boldsymbol{z}$$

Maximize the marginal log-likelihood:

$$\text{loglike}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \left(\int_{\boldsymbol{z}} \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z}+\boldsymbol{\mu},\sigma^2\boldsymbol{I})\mathcal{N}(\boldsymbol{z}|\boldsymbol{0},\boldsymbol{I})d\boldsymbol{z} \right)$$

This is a typical case for using EM

- It can however also be solved in closed form as everything is Gaussian and linear
- But it is somehow complex and using EM is a much more general solution
- Its a good example to understand EM

Expectation-step

We need to compute the posterior distribution

$$q_i(\boldsymbol{z}) = \underbrace{p(\boldsymbol{z}|\boldsymbol{x}_i, \boldsymbol{\theta})}_{\text{posterior}} = \frac{\overbrace{p(\boldsymbol{x}_i|\boldsymbol{z}, \boldsymbol{\theta})}^{\text{likelihood}} \overbrace{p(\boldsymbol{z})}^{\text{prior}}}{\underbrace{p(\boldsymbol{x}_i|\boldsymbol{\theta})}_{\text{evidence}}}$$

- Application of Bayes' Rule with Gaussian distributions

$$p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}), \quad p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

- Posterior is Gaussian with mean and variance

$$\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x}_i} = (\boldsymbol{W}^T\boldsymbol{W} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{\mu}), \qquad \boldsymbol{\Sigma}_{\boldsymbol{z}|\boldsymbol{x}_i} = \sigma^2(\boldsymbol{W}^T\boldsymbol{W} + \sigma^2\boldsymbol{I})^{-1}$$

  - Not covered now, see Lecture 9, Bayesian Learning
  - Only the case because ***x*** is linear in ***z***!

Maximization Step

Maximize the lower bound with respect to $\boldsymbol{\theta}$ ...

$$\begin{aligned}
\mathcal{L}(q, \boldsymbol{\theta}) &= \sum_i \left(\int_{\boldsymbol{z}} q_i(\boldsymbol{z}) \log p(\boldsymbol{x}_i, z|\boldsymbol{\theta}) dz - \int_{\boldsymbol{z}} q_i(\boldsymbol{z}) \log q_i(z) dz \right) \\
&= \sum_i \left(\int_{\boldsymbol{z}} q_i(\boldsymbol{z}) \big(\log p(\boldsymbol{x}_i|\boldsymbol{z}, \boldsymbol{\theta}) + \log p(\boldsymbol{z}) \big) dz - \int_z q_i(z) \log q_i(z) dz \right) \\
&= \sum_i \int_{\boldsymbol{z}} q_i(\boldsymbol{z}) \log p(\boldsymbol{x}_i|\boldsymbol{z}, \boldsymbol{\theta}) d\boldsymbol{z} + \underbrace{\text{const}}_{\text{independent of } \boldsymbol{\theta}} = \sum_i \mathbb{E}_{q_i(\boldsymbol{z})} \left[\log p(\boldsymbol{x}_i|\boldsymbol{z}, \boldsymbol{\theta}) \right] + \underbrace{\text{const}}_{\text{independent of } \boldsymbol{\theta}}
\end{aligned}$$

- **Continuous latent variable: How can we solve the integral?**
  - a) $q_i(z)$ is Gaussian, can be solved in principle in closed form (not covered)
  - **b) Simpler:** Sampling! I.e, we can use Monte Carlo Estimates

Recap: Monte-carlo estimation (Lecture 2)

Expectations can always be approximated by samples:

$$\mathbb{E}_p[f(x)] = \int p(x) f(x) dx \approx \frac{1}{N} \sum_{x_i \sim p(x)} f(x_i)$$

- Necessary if no analytical solution exists to compute the integral (typical case)

Maximization Step

Monte-Carlo estimate for the lower bound

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_i \mathbb{E}_{q_i(\boldsymbol{z})} \left[\log p(\boldsymbol{x}_i | \boldsymbol{z}, \boldsymbol{\theta}) \right] \approx \sum_i \frac{1}{N} \sum_{z_{ik} \sim q_i(\boldsymbol{z})} \log p(\boldsymbol{x}_i | \boldsymbol{z}_{ik}, \boldsymbol{\theta})$$

- If we only use a single sample $\boldsymbol{z}_i$ per i (i.e. N = 1), we get

$$\mathcal{L}(q, \boldsymbol{\theta}) \approx \sum_i \log p(\boldsymbol{x}_i | \boldsymbol{z}_i, \boldsymbol{\theta}), \text{ where } \boldsymbol{z}_i \sim q_i(\boldsymbol{z})$$

- Maximizing $\mathcal{L}(q, \boldsymbol{\theta})$ is a standard maximum likelihood problem with Gaussian linear models.
- We know the solution already (standard least squares):

$$\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{W} \end{bmatrix} = (\boldsymbol{Z}^T \boldsymbol{Z})^{-1} \boldsymbol{Z}^T \boldsymbol{X}, \quad \text{with } \boldsymbol{Z} = \begin{bmatrix} 1 & \boldsymbol{z}_1^T \\ \vdots & \vdots \\ 1 & \boldsymbol{z}_n^T \end{bmatrix} \text{ and } \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix}$$

$$\sigma^2 = \frac{1}{nd} \sum_{i=1}^{n} \sum_{k=1}^{d} (y_{ik} - x_{ik})^2, \quad \text{with } \boldsymbol{y}_i = \boldsymbol{W} \boldsymbol{z}_i + \boldsymbol{\mu}$$

k-th element of $\boldsymbol{y}_i$

Algorithm: EM for PCA

- **Initialize:** Use average of **x** for $\boldsymbol{\mu}$, random matrix **W**

- **Repeat** until convergence**:**
  - **Expectation-step:**
    - Compute posterior mean and covariance

$$\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x}_i} = (\boldsymbol{W}^T\boldsymbol{W} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{\mu}), \qquad \boldsymbol{\Sigma}_{\boldsymbol{z}|\boldsymbol{x}_i} = \sigma^2(\boldsymbol{W}^T\boldsymbol{W} + \sigma^2\boldsymbol{I})^{-1}$$

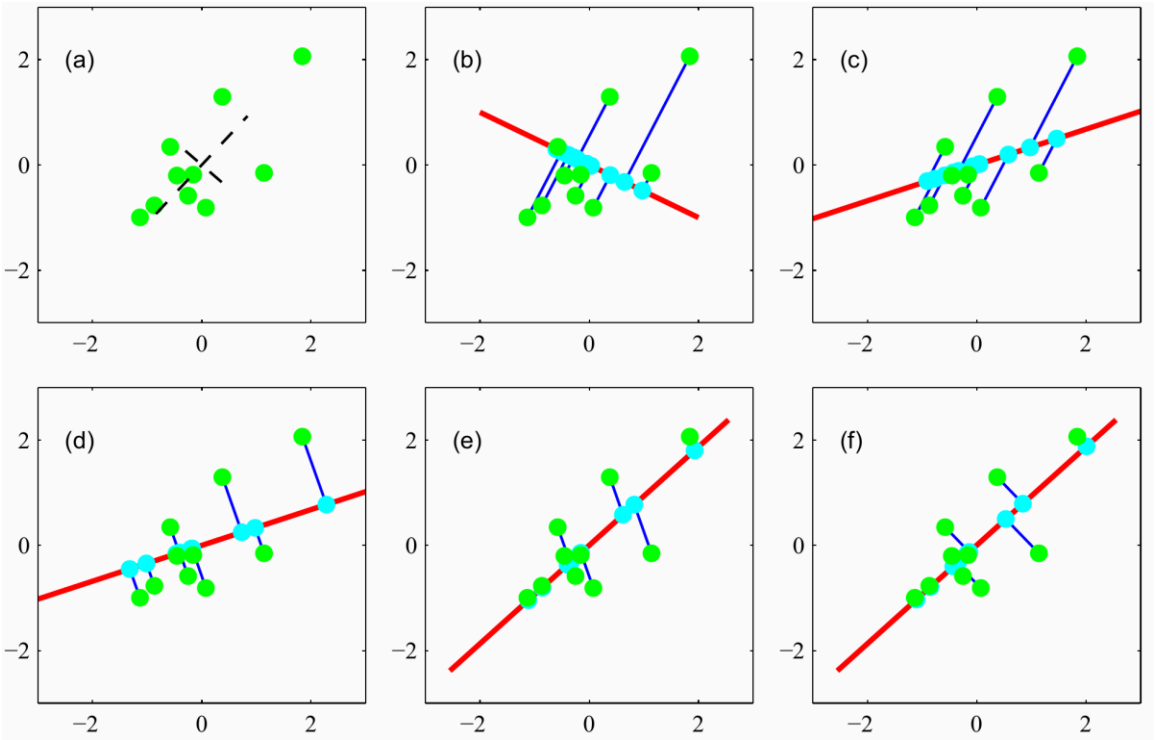
    - Generate latent samples:

$$\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x}_i}, \boldsymbol{\Sigma}_{\boldsymbol{z}|\boldsymbol{x}_i})$$

  - **Maximization-step:** Update $\boldsymbol{W}$, $\boldsymbol{\mu}$ and σ^2

$$\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{W} \end{bmatrix} = (\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}^T\boldsymbol{X}, \qquad \sigma^2 = \frac{1}{nd}\sum_{i=1}^{n}\sum_{k=1}^{d}(y_{ik} - x_{ik})^2$$

Illustration: EM for PCA



Probabilistic PCA vs. PCA

Typically, PCA using eigenvector decomposition is preferred:

- Single one-step solution
- Very fast

However, looking at EM for dimensionality reduction makes sense if:

- We need a density
- Helps us to understand EM
- Helps us to understand more complex dimensionality reduction methods (variational auto encoders use the same principles)

Additional Notes for EM

EM assumes that E-step can set the KL to zero:

- I.e. we can evaluate the posterior analytically
- Lower bound is tight
- Marginal likelihood always improves (good to check for debugging!)
- Only possible if z is discrete or we have linear Gaussian models!

For more complex latent variable models (e.g. Deep Neural Networks), this is typically not possible:

- Extension of EM called Variational Bayes / Variational Inference can still do that
- Approximates the posterior, i.e. KL will be > 0 after E-step
- Very active research, underlying algorithm of many deep learning architectures (e.g. variational autoencoder)
- Will be covered in the end of the lecture

Takeaway messages

You know now:

- The difference between parametric and non-parametric models
- Different non-parametric models (histogram, kernel density estimation and k-nearest neighbors)
- What mixture models and latent variable models are
- What the Expectation-Maximization idea and algorithm are
- Why does EM converge
- How to apply EM to GMMs (discrete latent variables) and PCA (continuous latent variables)

Self-test questions

- What are mixture models?
- Should gradient methods be used for training mixture models?
- How does the EM algorithm work?
- What is the biggest problem of mixture models?
- How does EM decomposes the marginal likelihood?
- Why does EM always improve the lower bound?
- Why does EM always improve the marginal likelihood?
- Why can we optimize each mixture component independently with EM
- Why do we need sampling for continuous latent variables?