Advanced Machine Learning
Summer 2019

Part 14 – Latent Variable Models
29.05.2019

Prof. Dr. Bastian Leibe

RWTH Aachen University, Computer Vision Group
http://www.vision.rwth-aachen.de
Course Outline

• Regression Techniques
  – Linear Regression
  – Regularization (Ridge, Lasso)
  – Kernels (Kernel Ridge Regression)

• Deep Reinforcement Learning

• Probabilistic Graphical Models
  – Bayesian Networks
  – Markov Random Fields
  – Inference (exact & approximate)
  – Latent Variable Models

• Deep Generative Models
  – Generative Adversarial Networks
  – Variational Autoencoders
Topics of This Lecture

• **Recap: MCMC**
  – Gibbs Sampling

• **Recap: Mixtures of Gaussians**
  – Mixtures of Gaussians
  – ML estimation
  – EM algorithm for MoGs

• **An alternative view of EM**
  – Latent variables
  – General EM
  – Mixtures of Gaussians revisited
  – Mixtures of Bernoulli distributions

• **The EM algorithm in general**
  – Generalized EM
  – Relation to Variational inference
Recap: MCMC – Markov Chain Monte Carlo

• Overview
  – Allows to sample from a large class of distributions.
  – Scales well with the dimensionality of the sample space.

• Idea
  – We maintain a record of the current state \( z^{(\tau)} \)
  – The proposal distribution depends on the current state: \( q(z|z^{(\tau)}) \)
  – The sequence of samples forms a Markov chain \( z^{(1)}, z^{(2)}, \ldots \)

• Approach
  – At each time step, we generate a candidate sample from the proposal distribution and accept the sample according to a criterion.
  – Different variants of MCMC for different criteria.
Recap: Markov Chains – Properties

- **Invariant distribution**
  - A distribution is said to be invariant (or stationary) w.r.t. a Markov chain if each step in the chain leaves that distribution invariant.
  - Transition probabilities:
    \[ T \left( z^{(m)}, z^{(m+1)} \right) = p \left( z^{(m+1)} | z^{(m)} \right) \]
  - For homogeneous Markov chain, distribution \( p^*(z) \) is invariant if:
    \[ p^*(z) = \sum_{z'} T(z', z) p^*(z') \]

- **Detailed balance**
  - Sufficient (but not necessary) condition to ensure that a distribution is invariant:
    \[ p^*(z) T(z, z') = p^*(z') T(z', z) \]
  - A Markov chain which respects detailed balance is reversible.
Recap: MCMC – Metropolis Algorithm

• **Metropolis algorithm**
  - Proposal distribution is symmetric: \( q(z_A | z_B) = q(z_B | z_A) \)
  - The new candidate sample \( z^* \) is accepted with probability
    \[
    A(z^*, z^{(\tau)}) = \min \left( 1, \frac{\tilde{p}(z^*)}{\tilde{p}(z^{(\tau)})} \right)
    \]
  - New candidate samples always accepted if \( \tilde{p}(z^*) \geq \tilde{p}(z^{(\tau)}) \)
  - The algorithm sometimes accepts a state with lower probability.

• **Metropolis-Hastings algorithm**
  - Generalization: Proposal distribution not necessarily symmetric.
  - The new candidate sample \( z^* \) is accepted with probability
    \[
    A(z^*, z^{(\tau)}) = \min \left( 1, \frac{\tilde{p}(z^*) q_k(z^{(\tau)} | z^*)}{\tilde{p}(z^{(\tau)}) q_k(z^* | z^{(\tau)})} \right)
    \]
  - where \( k \) labels the members of the set of considered transitions.
Recap: Gibbs Sampling

• Approach
  – MCMC-algorithm that is simple and widely applicable.
  – May be seen as a special case of Metropolis-Hastings.

• Idea
  – Sample variable-wise: replace $z_i$ by a value drawn from the distribution $p(z_i|z_{\setminus i})$.
    ▪ This means we update one coordinate at a time.
  – Repeat procedure either by cycling through all variables or by choosing the next variable.
Recap: Gibbs Sampling

• Properties
  – The factor that determines the acceptance probability in the Metropolis-Hastings is determined by
  
  \[ A(z^*, z) = \frac{p(z^*) q_k(z \mid z^*)}{p(z) q_k(z^* \mid z)} = \frac{p(z_k^* \mid z_k^*) p(z_k^*) p(z_k^* \mid z_k^*)}{p(z_k \mid z_k^*) p(z_k) p(z_k \mid z_k^*)} = 1 \]

  – (we have used \( q_k(z^* \mid z) = p(z_k^* \mid z_{\setminus k}) \) and \( p(z) = p(z_k \mid z_{\setminus k}) p(z_{\setminus k}) \)).

  – I.e. we get an algorithm which always accepts!

⇒ If you can compute (and sample from) the conditionals, you can apply Gibbs sampling.
⇒ The algorithm is completely parameter free.
⇒ Can also be applied to subsets of variables.
• Gibbs sampling benefits from few free choices and convenient features of conditional distributions:
  – Conditionals with a few discrete settings can be explicitly normalized:
    \[
    p(x_i | x_j \neq i) = \frac{p(x_i, x_j \neq i)}{\sum_{x_i'} p(x_i', x_j \neq i)}
    \]
  – Continuous conditionals are often only univariate. ⇒ Amenable to standard sampling methods.
  – In case of graphical models, the conditional distributions depend only on the variables in the corresponding Markov blankets.

Slide adapted from Iain Murray
Gibbs Sampling

• Example
  – 20 iterations of Gibbs sampling on a bivariate Gaussian.

  – Note: strong correlations can slow down Gibbs sampling.
How Should We Run MCMC?

• Arbitrary initialization means starting iterations are bad
  – Discard a “burn-in” period.

• How do we know if we have run for long enough?
  – You don’t. That’s the problem.

• The samples are not independent
  – Solution 1: Keep only every $M^{th}$ sample (“thinning”).
  – Solution 2: Keep all samples and use the simple Monte Carlo estimator on MCMC samples
    ▪ It is consistent and unbiased if the chain has “burned in”.
    ⇒ Use thinning only if computing $f(x^{(s)})$ is expensive.

• For opinion on thinning, multiple runs, burn in, etc.
Summary: Approximate Inference

• Exact Bayesian Inference often intractable.

• Rejection and Importance Sampling
  – Generate independent samples.
  – Impractical in high-dimensional state spaces.

• Markov Chain Monte Carlo (MCMC)
  – Simple & effective (even though typically computationally expensive).
  – Scales well with the dimensionality of the state space.
  – Issues of convergence have to be considered carefully.

• Gibbs Sampling
  – Used extensively in practice.
  – Parameter free
  – Requires sampling conditional distributions.
Topics of This Lecture

• Recap: MCMC
  – Gibbs Sampling

• Recap: Mixtures of Gaussians
  – Mixtures of Gaussians
  – ML estimation
  – EM algorithm for MoGs

• An alternative view of EM
  – Latent variables
  – General EM
  – Mixtures of Gaussians revisited
  – Mixtures of Bernoulli distributions

• The EM algorithm in general
  – Generalized EM
  – Relation to Variational inference
Recap: Mixture of Gaussians (MoG)

• “Generative model”

\[ p(j) = \pi_j \]

\[ p(x|\theta_j) \]

\[ p(x) = \sum_{j=1}^{M} p(x|\theta_j)p(j) \]

“Weight” of mixture component

Mixture component

Mixture density
Recap: Mixture of Multivariate Gaussians

- Multivariate Gaussians

\[
p(x|\theta) = \sum_{j=1}^{M} p(x|\theta_j)p(j)
\]

\[
p(x|\theta_j) = \frac{1}{(2\pi)^{D/2}|\Sigma_j|^{1/2}} \exp \left\{ -\frac{1}{2}(x - \mu_j)^T \Sigma_j^{-1}(x - \mu_j) \right\}
\]

- Mixture weights / mixture coefficients:

\[
p(j) = \pi_j \text{ with } 0 \leq \pi_j \leq 1 \text{ and } \sum_{j=1}^{M} \pi_j = 1
\]

- Parameters:

\[
\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_M, \mu_M, \Sigma_M)
\]

Image source: C.M. Bishop, 2006
Recap: Mixtures of Gaussians

- “Generative model”

\[
p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)
\]

\[
p(j) = \pi_j
\]

\[
p(x|\theta) = \sum_{j=1}^{3} \pi_j p(x|\theta_j)
\]

Image source: C.M. Bishop, 2006
Slide credit: Bernt Schiele
Recap: ML for Mixtures of Gaussians

- **Maximum Likelihood**
  
  - Minimize \( E = - \ln L(\theta) = - \sum_{n=1}^{N} \ln p(x_n|\theta) \)

  - We can already see that this will be difficult, since

  \[
  \ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}
  \]

  This will cause problems!
Recap: ML for Mixtures of Gaussians

- Minimization:

\[
\frac{\partial E}{\partial \mu_j} = - \sum_{n=1}^{N} \frac{\partial}{\partial \mu_j} p(x_n | \theta_j) \frac{\sum_{k=1}^{K} p(x_n | \theta_k)}{\sum_{k=1}^{K} p(x_n | \theta_k)}
\]

\[
= - \sum_{n=1}^{N} \left( \Sigma^{-1} (x_n - \mu_j) \frac{p(x_n | \theta_j)}{\sum_{k=1}^{K} p(x_n | \theta_k)} \right)
\]

\[
= - \Sigma^{-1} \sum_{n=1}^{N} (x_n - \mu_j) \frac{\pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)} \overset{!}{=} 0
\]

- We thus obtain

\[
\Rightarrow \mu_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n) x_n}{\sum_{n=1}^{N} \gamma_j(x_n)}
\]

"responsibility" of component \(j\) for \(x_n\)
Recap: ML for Mixtures of Gaussians

- But...

$$\mu_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n) x_n}{\sum_{n=1}^{N} \gamma_j(x_n)}$$

$$\gamma_j(x_n) = \frac{\pi_j \mathcal{N}(x_n; \mu_j, \Sigma_j)}{\sum_{k=1}^{N} \pi_k \mathcal{N}(x_n; \mu_k, \Sigma_k)}$$

- I.e. there is no direct analytical solution!

$$\frac{\partial E}{\partial \mu_j} = f(\pi_1, \mu_1, \Sigma_1, \ldots, \pi_M, \mu_M, \Sigma_M)$$

  - Complex gradient function (non-linear mutual dependencies)
  - Optimization of one Gaussian depends on all other Gaussians!
  - It is possible to apply iterative numerical optimization here, but the EM algorithm provides a simpler alternative.
Recap: EM Algorithm

- **Expectation-Maximization (EM) Algorithm**
  - **E-Step**: softly assign samples to mixture components
    \[
    \gamma_j(x_n) \leftarrow \frac{\pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}{\sum_{k=1}^{N} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)} \quad \forall j = 1, \ldots, K, \quad n = 1, \ldots, N
    \]
  - **M-Step**: re-estimate the parameters (separately for each mixture component) based on the soft assignments
    \[
    \hat{\pi}_j^{\text{new}} \leftarrow \frac{\hat{N}_j}{\hat{N}} \quad \hat{N}_j \leftarrow \sum_{n=1}^{N} \gamma_j(x_n) = \text{soft #samples labeled } j
    \]
    \[
    \hat{\mu}_j^{\text{new}} \leftarrow \frac{1}{\hat{N}_j} \sum_{n=1}^{N} \gamma_j(x_n) x_n
    \]
    \[
    \hat{\Sigma}_j^{\text{new}} \leftarrow \frac{1}{\hat{N}_j} \sum_{n=1}^{N} \gamma_j(x_n)(x_n - \hat{\mu}_j^{\text{new}})(x_n - \hat{\mu}_j^{\text{new}})^T
    \]
Outlook for Today

• Criticism
  – This is all very nice, but in the ML lecture, the EM algorithm miraculously fell out of thin air.
  – Why do we actually solve it this way?

• This lecture
  – We will take a more general view on EM
    ▪ Different interpretation in terms of latent variables
    ▪ Detailed derivation
  – This will allow us to derive EM algorithms also for other cases.
Topics of This Lecture

• Recap: MCMC
  – Gibbs Sampling

• Recap: Mixtures of Gaussians
  – Mixtures of Gaussians
  – ML estimation
  – EM algorithm for MoGs

• An alternative view of EM
  – Latent variables
  – General EM
  – Mixtures of Gaussians revisited
  – Mixtures of Bernoulli distributions

• The EM algorithm in general
  – Generalized EM
  – Relation to Variational inference
• Mixture of Gaussians
  – Can be written as linear superposition of Gaussians in the form

  \[ p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \]

• Let’s write this in a different form…
  – Introduce a \( K \)-dimensional binary random variable \( z \) with a 1-of-\( K \) coding, i.e., \( z_k = 1 \) and all other elements are zero.
  
  – Define the joint distribution over \( x \) and \( z \) as

  \[ p(x, z) = p(x|z)p(z) \]

  – This corresponds to the following graphical model:
Gaussian Mixtures as Latent Variable Models

• Marginal distribution over \( z \)
  – Specified in terms of the mixing coefficients \( \pi_k \), such that
    \[
    p(z_k = 1) = \pi_k
    \]
    where \( 0 \cdot \pi_j \cdot 1 \) and \( \sum_{j=1}^{K} \pi_j = 1 \).
  – Since \( z \) uses a 1-of-\( K \) representation, we can also write this as
    \[
    p(z) = \prod_{k=1}^{K} \pi_k^{z_k}
    \]
  – Similarly, we can write for the conditional distribution
    \[
    p(x|z) = \prod_{k=1}^{K} \mathcal{N}(x|\mu_k, \Sigma_k)^{z_k}
    \]
Gaussian Mixtures as Latent Variable Models

• Marginal distribution of \( x \)
  – Summing the joint distribution over all possible states of \( z \)

\[
p(x) = \sum_z p(x, z) = \sum_z p(z)p(x|z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)
\]

• What have we gained by this?
  – The resulting formula looks still the same after all…
  ⇒ We have represented the marginal distribution in terms of latent variables \( z \).
  – Since \( p(x) = \sum_z p(x, z) \), there is a corresponding latent variable \( z_n \) for each data point \( x_n \).
  – We are now able to work with the joint distribution \( p(x, z) \) instead of the marginal distribution \( p(x) \).
  ⇒ This will lead to significant simplifications…
• Conditional probability of $z$ given $x$:
  – Use again the “responsibility” notation $\gamma(z_k)$

$$
\gamma(z_k) \equiv p(z_k = 1|x) = \frac{p(z_k = 1)p(x|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(x|z_j = 1)}
$$

$$
= \frac{\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)}
$$

  – We can view $\pi_k$ as the prior probability of $z_k = 1$ and $\gamma(z_k)$ as the corresponding posterior once we have observed $x$. 

Gaussian Mixtures as Latent Variable Models
Sidenote: Sampling from a Gaussian Mixture

• MoG Sampling
  – We can use ancestral sampling to generate random samples from a Gaussian mixture model.
    1. Generate a value $\hat{z}$ from the marginal distribution $p(z)$.
    2. Generate a value $\hat{x}$ from the conditional distribution $p(x|\hat{z})$.

Samples from the joint $p(x, z)$
Samples from the marginal $p(x)$
Evaluating the responsibilities $\gamma(z_{nk})$
Alternative View of EM

• Complementary view of the EM algorithm
  – The goal of EM is to find ML solutions for models having latent variables.

  – Notation
    ▪ Set of all data \( X = [x_1, \ldots, x_N]^T \)
    ▪ Set of all latent variables \( Z = [z_1, \ldots, z_N]^T \)
    ▪ Set of all model parameters \( \theta \)

  – Log-likelihood function
    \[
    \log p(X|\theta) = \log \left\{ \sum_Z p(X, Z|\theta) \right\}
    \]

  – Key observation: summation inside logarithm \( \Rightarrow \) difficult.
Alternative View of EM

- Now, suppose we were told for each observation in $\mathbf{X}$ the corresponding value of the latent variable $\mathbf{Z}$…
  - Call $\{\mathbf{X}, \mathbf{Z}\}$ the complete data set and refer to the actual observed data $\mathbf{X}$ as incomplete.

  - The likelihood for the complete data set now takes the form
    \[
    \log p(\mathbf{X}, \mathbf{Z}|\theta)
    \]

  $\Rightarrow$ Straightforward to marginalize…
Alternative View of EM

• In practice, however,…
  – We are not given the complete data set \{X,Z\}, but only the incomplete data \(X\).
  – Our knowledge of the latent variable values in \(Z\) is given only by the posterior distribution \(p(Z|X, \theta)\).
  – Since we cannot use the complete-data log-likelihood, we consider instead its expected value under the posterior distribution of the latent variables:
    \[
    Q(\theta, \theta^{\text{old}}) = \sum_Z p(Z|X, \theta^{\text{old}}) \log p(X, Z|\theta)
    \]
    – This corresponds to the E-step of the EM algorithm.
  – In the subsequent M-step, we then maximize the expectation to obtain the revised parameter set \(\theta^{\text{new}}\).
    \[
    \theta^{\text{new}} = \arg \max_{\theta} Q(\theta, \theta^{\text{old}})
    \]
General EM Algorithm

• Algorithm
  1. Choose an initial setting for the parameters \( \theta^{\text{old}} \)
  2. \textbf{E-step:} Evaluate \( p(Z|X, \theta^{\text{old}}) \)
  3. \textbf{M-step:} Evaluate \( \theta^{\text{new}} \) given by
     \[
     \theta^{\text{new}} = \arg \max_{\theta} Q(\theta, \theta^{\text{old}})
     \]
     where
     \[
     Q(\theta, \theta^{\text{old}}) = \sum_{Z} p(Z|X, \theta^{\text{old}}) \log p(X, Z|\theta)
     \]
  4. While not converged, let \( \theta^{\text{old}} \leftarrow \theta^{\text{new}} \) and return to step 2.
Remark: MAP-EM

- Modification for MAP
  - The EM algorithm can be adapted to find MAP solutions for models for which a prior $p(\theta)$ is defined over the parameters.
  - Only changes needed:

2. **E-step**: Evaluate $p(Z|X, \theta^{old})$

3. **M-step**: Evaluate $\theta^{new}$ given by

$$\theta^{new} = \arg \max_{\theta} Q(\theta, \theta^{old}) + \log p(\theta)$$

⇒ Suitable choices for the prior will remove the ML singularities!
Remark: Monte Carlo EM

• EM procedure
  – **M-step**: Maximize expectation of complete-data log-likelihood

  \[ Q(\theta, \theta^{\text{old}}) = \int p(Z|X, \theta^{\text{old}}) \log p(X, Z|\theta) \, dZ \]

  – For more complex models, we may not be able to compute this analytically anymore…

• Idea
  – Use sampling to approximate this integral by a finite sum over samples \( \{Z^{(l)}\} \) drawn from the current estimate of the posterior

  \[ Q(\theta, \theta^{\text{old}}) \sim \frac{1}{L} \sum_{l=1}^{L} \log p(X, Z^{(l)}|\theta) \]

  – This procedure is called the **Monte Carlo EM algorithm**.
Gaussian Mixtures Revisited

• Applying the latent variable view of EM
  – Goal is to maximize the log-likelihood using the observed data \( \mathbf{X} \)

\[
\log p(\mathbf{X}|\theta) = \log \left\{ \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta) \right\}
\]

  – Corresponding graphical model:

  – Suppose we are additionally given the values of the latent variables \( \mathbf{Z} \).
  – The corresponding graphical model for the complete data now looks like this:
Gaussian Mixtures Revisited

- Maximize the likelihood
  - For the complete-data set \( \{X, Z\} \), the likelihood has the form
    \[
    p(X, Z|\mu, \Sigma, \pi) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}(x_n|\mu_k, \Sigma_k)^{z_{nk}}
    \]
  - Taking the logarithm, we obtain
    \[
    \log p(X, Z|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left\{ \log \pi_k + \log \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}
    \]
  - Compared to the incomplete-data case, the order of the sum and logarithm has been interchanged.
    \[ \Rightarrow \text{Much simpler solution to the ML problem.} \]
  - Maximization w.r.t. a mean or covariance is exactly as for a single Gaussian, except that it involves only the subset of data points that are “assigned” to that component \((z_{nk} = 1)\).
Gaussian Mixtures Revisited

• Maximization w.r.t. mixing coefficients
  – More complex, since the $\pi_k$ are coupled by the summation constraint
    \[ \sum_{j=1}^{K} \pi_j = 1 \]
  – Solve with a Lagrange multiplier
    \[ \log p(X, Z|\mu, \Sigma, \pi) + \lambda \left( \sum_{k=1}^{K} \pi_k - 1 \right) \]
  – Solution (after a longer derivation):
    \[ \pi_k = \frac{1}{N} \sum_{n=1}^{N} z_{nk} \]
    \[ \Rightarrow \text{The complete-data log-likelihood can be maximized trivially in closed form.} \]
Gaussian Mixtures Revisited

• In practice, we don’t have values for the latent variables
  – Consider the expectation w.r.t. the posterior distribution of the latent variables instead.
  – The posterior distribution takes the form

\[
p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\mu}, \Sigma, \pi) \propto \prod_{n=1}^{N} \prod_{k=1}^{K} \left[ \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k) \right]^{z_{nk}}
\]

and factorizes over \( n \), so that the \{\mathbf{z}_n\} are independent under the posterior.

– Expected value of indicator variable \( z_{nk} \) under the posterior.

\[
\mathbb{E}[z_{nk}] = \frac{\sum_{z_{nk}} z_{nk} \left[ \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k) \right]^{z_{nk}}}{\sum_{z_{nj}} \left[ \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \Sigma_j) \right]^{z_{nj}}} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \Sigma_j)} = \gamma(z_{nk})
\]
Gaussian Mixtures Revisited

• Continuing the estimation
  – The expected value of the complete-data log-likelihood is therefore
  \[
  \mathbb{E}_Z[\log p(X, Z|\mu, \Sigma, \pi)] = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \{\log \pi_k + \log \mathcal{N}(x_n|\mu_k, \Sigma_k)\}
  \]

• Putting everything together
  – Start by choosing some initial values for $\mu^{old}$, $\Sigma^{old}$, and $\pi^{old}$.
  – Use these to evaluate the responsibilities (the E-Step).
  – Keep the responsibilities fixed and maximize the above for $\mu^{new}$, $\Sigma^{new}$, and $\pi^{new}$ (the M-Step).
  – This leads to the familiar closed-form solutions for $\mu^{new}$, $\Sigma^{new}$, and $\pi^{new}$.

$\Rightarrow$ This is precisely the EM algorithm for Gaussian mixtures as derived before. But we can now also apply it to other distributions.
References and Further Reading

• More information about EM and MoG estimation is available in Chapter 9 of Bishop’s book (recommendable to read).

Christopher M. Bishop
Pattern Recognition and Machine Learning
Springer, 2006

• Additional information