Introduction to Latent Variable Models

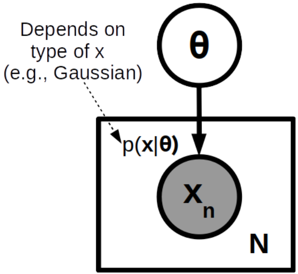
Piyush Rai

Probabilistic Machine Learning (CS772A)

Aug 29, 2017

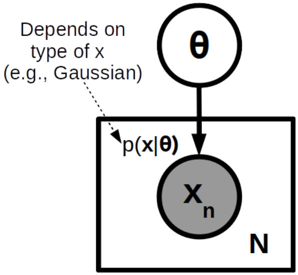
A Simple Generative Model

- All observations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ generated from a distribution $p(\boldsymbol{x}|\theta)$



A Simple Generative Model

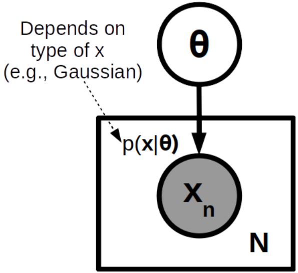
- All observations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ generated from a distribution $p(\boldsymbol{x}|\theta)$



- Unknowns: Parameters θ of the assumed data distribution $p(\boldsymbol{x}|\theta)$

A Simple Generative Model

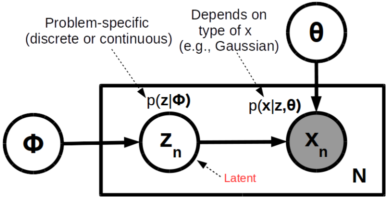
- All observations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ generated from a distribution $p(\boldsymbol{x}|\theta)$



- Unknowns: Parameters θ of the assumed data distribution $p(\boldsymbol{x}|\theta)$
- Many ways to estimate the parameters (MLE, MAP, or Bayesian inference)

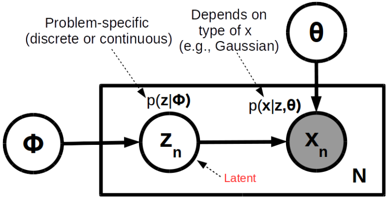
Generative Model with Latent Variables

- Assume each observation $\boldsymbol{x}_n$ to be associated with a latent variable $\boldsymbol{z}_n$



Generative Model with Latent Variables

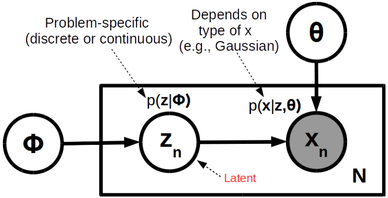
- Assume each observation $\boldsymbol{x}_n$ to be associated with a latent variable $\boldsymbol{z}_n$



- In this "latent variable model" of data generation, data $\boldsymbol{x}$ also depends some latent variable(s) $\boldsymbol{z}$
- $\boldsymbol{z}_n$ is akin to a latent representation or "encoding" of $\boldsymbol{x}_n$; controls what data "looks like".

Generative Model with Latent Variables

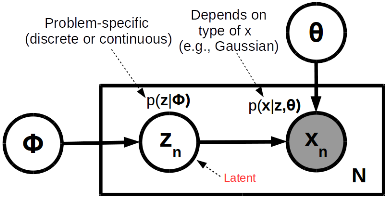
- Assume each observation $\boldsymbol{x}_n$ to be associated with a latent variable $\boldsymbol{z}_n$



- In this "latent variable model" of data generation, data $\boldsymbol{x}$ also depends some latent variable(s) $\boldsymbol{z}$
- $\boldsymbol{z}_n$ is akin to a latent representation or "encoding" of $\boldsymbol{x}_n$; controls what data "looks like".E.g,
  - $\boldsymbol{z}_n \in \{1, \ldots, K\}$ denotes the cluster $\boldsymbol{x}_n$ belongs to

Generative Model with Latent Variables

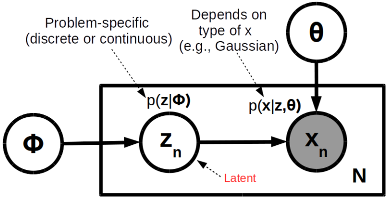
- Assume each observation $\boldsymbol{x}_n$ to be associated with a latent variable $\boldsymbol{z}_n$



- In this "latent variable model" of data generation, data $\boldsymbol{x}$ also depends some latent variable(s) $\boldsymbol{z}$
- $\boldsymbol{z}_n$ is akin to a latent representation or "encoding" of $\boldsymbol{x}_n$; controls what data "looks like".E.g,
  - $\boldsymbol{z}_n \in \{1, \ldots, K\}$ denotes the cluster $\boldsymbol{x}_n$ belongs to
  - $\boldsymbol{z}_n \in \mathbb{R}^K$ denotes a low-dimensional latent representation or latent "code" for $\boldsymbol{x}_n$

Generative Model with Latent Variables

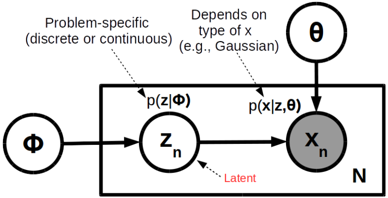
- Assume each observation $\boldsymbol{x}_n$ to be associated with a latent variable $\boldsymbol{z}_n$



- In this "latent variable model" of data generation, data $\boldsymbol{x}$ also depends some latent variable(s) $\boldsymbol{z}$
- $\boldsymbol{z}_n$ is akin to a latent representation or "encoding" of $\boldsymbol{x}_n$; controls what data "looks like".E.g,
  - $\boldsymbol{z}_n \in \{1, \ldots, K\}$ denotes the cluster $\boldsymbol{x}_n$ belongs to
  - $\boldsymbol{z}_n \in \mathbb{R}^K$ denotes a low-dimensional latent representation or latent "code" for $\boldsymbol{x}_n$
- Unknowns: $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$, and (θ, ϕ).

Generative Model with Latent Variables

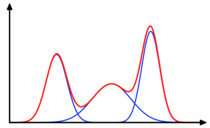
- Assume each observation $\boldsymbol{x}_n$ to be associated with a latent variable $\boldsymbol{z}_n$



- In this "latent variable model" of data generation, data $\boldsymbol{x}$ also depends some latent variable(s) $\boldsymbol{z}$
- $\boldsymbol{z}_n$ is akin to a latent representation or "encoding" of $\boldsymbol{x}_n$; controls what data "looks like". E.g,
  - $\boldsymbol{z}_n \in \{1, \ldots, K\}$ denotes the cluster $\boldsymbol{x}_n$ belongs to
  - $\boldsymbol{z}_n \in \mathbb{R}^K$ denotes a low-dimensional latent representation or latent "code" for $\boldsymbol{x}_n$
- Unknowns: $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$, and (θ, ϕ). $\boldsymbol{z}_n$'s called "local" variables; (θ, ϕ) called "global" variables

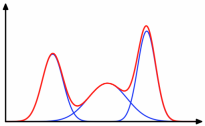
A Motivating Example: Mixture Model

- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$



A Motivating Example: Mixture Model

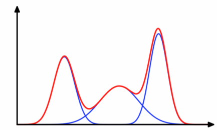
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$



- Don't know mixture component generated each $\boldsymbol{x}_n$

A Motivating Example: Mixture Model

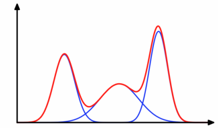
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$



- Don't know mixture component generated each $\boldsymbol{x}_n$ (o/w it is simply generative classification)

A Motivating Example: Mixture Model

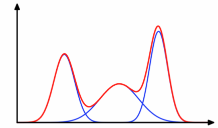
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^{N}$ was generated from a mixture of K distributions $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$



- Don't know mixture component generated each $\boldsymbol{x}_n$ (o/w it is simply generative classification)
- Assume a latent random variable $\boldsymbol{z}_n \in \{1, \ldots, K\}$ denotes which component generated $\boldsymbol{x}_n$

A Motivating Example: Mixture Model

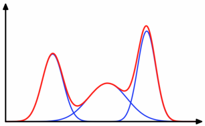
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$



- Don't know mixture component generated each $\boldsymbol{x}_n$ (o/w it is simply generative classification)
- Assume a latent random variable $\boldsymbol{z}_n \in \{1, \ldots, K\}$ denotes which component generated $\boldsymbol{x}_n$
- Here is a simple generative story for each $\boldsymbol{x}_n$, $n = 1, 2, \ldots, N$

A Motivating Example: Mixture Model

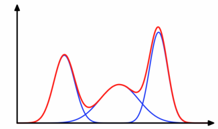
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^{N}$ was generated from a mixture of K distributions $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$



- Don't know mixture component generated each $\boldsymbol{x}_n$ (o/w it is simply generative classification)
- Assume a latent random variable $\boldsymbol{z}_n \in \{1, \ldots, K\}$ denotes which component generated $\boldsymbol{x}_n$
- Here is a simple generative story for each $\boldsymbol{x}_n$, $n = 1, 2, \ldots, N$
    - First choose a mixture component $\boldsymbol{z}_n \in \{1, 2, \ldots, K\}$ as $\boldsymbol{z}_n \sim \text{multinoulli}(\boldsymbol{z}|\phi)$

A Motivating Example: Mixture Model

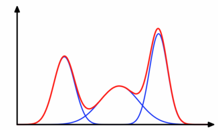
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$



- Don't know mixture component generated each $\boldsymbol{x}_n$ (o/w it is simply generative classification)
- Assume a latent random variable $\boldsymbol{z}_n \in \{1, \ldots, K\}$ denotes which component generated $\boldsymbol{x}_n$
- Here is a simple generative story for each $\boldsymbol{x}_n$, $n = 1, 2, \ldots, N$
  - First choose a mixture component $\boldsymbol{z}_n \in \{1, 2, \ldots, K\}$ as $\boldsymbol{z}_n \sim \text{multinoulli}(\boldsymbol{z}|\phi)$
  - Now generate $\boldsymbol{x}_n$ from that mixture component as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta_{z_n})$

A Motivating Example: Mixture Model

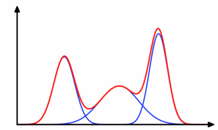
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$



- Don't know mixture component generated each $\boldsymbol{x}_n$ (o/w it is simply generative classification)
- Assume a latent random variable $\boldsymbol{z}_n \in \{1, \ldots, K\}$ denotes which component generated $\boldsymbol{x}_n$
- Here is a simple generative story for each $\boldsymbol{x}_n$, $n = 1, 2, \ldots, N$
  - First choose a mixture component $\boldsymbol{z}_n \in \{1, 2, \ldots, K\}$ as $\boldsymbol{z}_n \sim \text{multinoulli}(\boldsymbol{z}|\phi)$
  - Now generate $\boldsymbol{x}_n$ from that mixture component as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta_{\boldsymbol{z}_n})$
- Goal: Given data $\{\boldsymbol{x}_n\}_{n=1}^N$, learn the K distributions $(\theta_1, \ldots, \theta_K)$ and latent variables $\boldsymbol{z}_1 \ldots, \boldsymbol{z}_N$

A Motivating Example: Mixture Model

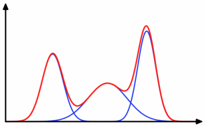
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$



- Don't know mixture component generated each $\boldsymbol{x}_n$ (o/w it is simply generative classification)
- Assume a latent random variable $\boldsymbol{z}_n \in \{1, \ldots, K\}$ denotes which component generated $\boldsymbol{x}_n$
- Here is a simple generative story for each $\boldsymbol{x}_n$, $n = 1, 2, \ldots, N$
  - First choose a mixture component $\boldsymbol{z}_n \in \{1, 2, \ldots, K\}$ as $\boldsymbol{z}_n \sim \text{multinoulli}(\boldsymbol{z}|\phi)$
  - Now generate $\boldsymbol{x}_n$ from that mixture component as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta_{\boldsymbol{z}_n})$
- Goal: Given data $\{\boldsymbol{x}_n\}_{n=1}^N$, learn the K distributions $(\theta_1, \ldots, \theta_K)$ and latent variables $\boldsymbol{z}_1 \ldots, \boldsymbol{z}_N$
- If each $p(\boldsymbol{x}|\theta_k)$ is a Gaussian $\Rightarrow$ Gaussian Mixture Model

A Motivating Example: Mixture Model

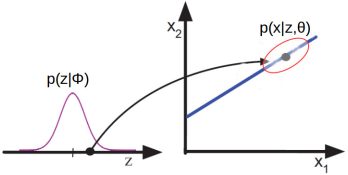
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$



- Don't know mixture component generated each $\boldsymbol{x}_n$ (o/w it is simply generative classification)
- Assume a latent random variable $\boldsymbol{z}_n \in \{1, \ldots, K\}$ denotes which component generated $\boldsymbol{x}_n$
- Here is a simple generative story for each $\boldsymbol{x}_n$, $n = 1, 2, \ldots, N$
  - First choose a mixture component $\boldsymbol{z}_n \in \{1, 2, \ldots, K\}$ as $\boldsymbol{z}_n \sim \text{multinoulli}(\boldsymbol{z}|\phi)$
  - Now generate $\boldsymbol{x}_n$ from that mixture component as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta_{\boldsymbol{z}_n})$
- Goal: Given data $\{\boldsymbol{x}_n\}_{n=1}^N$, learn the K distributions $(\theta_1, \ldots, \theta_K)$ and latent variables $\boldsymbol{z}_1 \ldots, \boldsymbol{z}_N$
- If each $p(\boldsymbol{x}|\theta_k)$ is a Gaussian $\Rightarrow$ Gaussian Mixture Model (used for probabilistic or "soft" clustering)

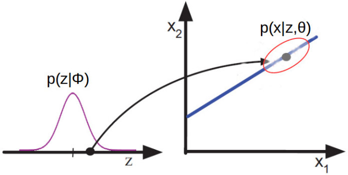
Another Motivating Example: Latent Factor Model

- Assume data $\boldsymbol{x}_n \in \mathbb{R}^D$ generated from a low-dimensional latent factor $\boldsymbol{z}_n \in \mathbb{R}^K$



Another Motivating Example: Latent Factor Model

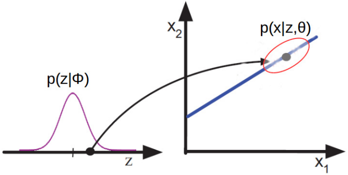
- Assume data $\boldsymbol{x}_n \in \mathbb{R}^D$ generated from a low-dimensional latent factor $\boldsymbol{z}_n \in \mathbb{R}^K$



- The $\boldsymbol{z}$ to $\boldsymbol{x}$ map can be a linear/nonlinear transformation

Another Motivating Example: Latent Factor Model

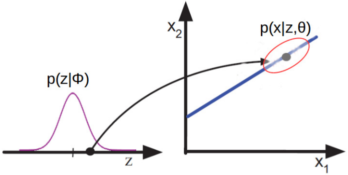
- Assume data $\boldsymbol{x}_n \in \mathbb{R}^D$ generated from a low-dimensional latent factor $\boldsymbol{z}_n \in \mathbb{R}^K$



- The $\boldsymbol{z}$ to $\boldsymbol{x}$ map can be a linear/nonlinear transformation
- Consider the following generative story for each $\boldsymbol{x}_n$, $n = 1, 2, \ldots, N$

Another Motivating Example: Latent Factor Model

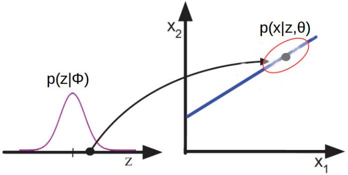
- Assume data $\boldsymbol{x}_n \in \mathbb{R}^D$ generated from a low-dimensional latent factor $\boldsymbol{z}_n \in \mathbb{R}^K$



- The $\boldsymbol{z}$ to $\boldsymbol{x}$ map can be a linear/nonlinear transformation
- Consider the following generative story for each $\boldsymbol{x}_n$, $n = 1, 2, \ldots, N$
  - First generate $\boldsymbol{z}_n$ from a K-dim distr. as $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$

Another Motivating Example: Latent Factor Model

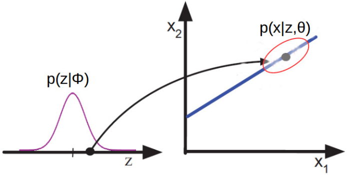
- Assume data $\boldsymbol{x}_n \in \mathbb{R}^D$ generated from a low-dimensional latent factor $\boldsymbol{z}_n \in \mathbb{R}^K$



- The $\boldsymbol{z}$ to $\boldsymbol{x}$ map can be a linear/nonlinear transformation
- Consider the following generative story for each $\boldsymbol{x}_n$, $n = 1, 2, \ldots, N$
  - First generate $\boldsymbol{z}_n$ from a K-dim distr. as $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$
  - Now generate $\boldsymbol{x}_n$ from a D-dim distr. as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\boldsymbol{z}_n, \theta)$

Another Motivating Example: Latent Factor Model

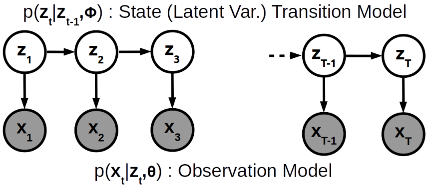
- Assume data $\boldsymbol{x}_n \in \mathbb{R}^D$ generated from a low-dimensional latent factor $\boldsymbol{z}_n \in \mathbb{R}^K$



- The $\boldsymbol{z}$ to $\boldsymbol{x}$ map can be a linear/nonlinear transformation
- Consider the following generative story for each $\boldsymbol{x}_n$, $n = 1, 2, \ldots, N$
  - First generate $\boldsymbol{z}_n$ from a K-dim distr. as $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$
  - Now generate $\boldsymbol{x}_n$ from a D-dim distr. as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\boldsymbol{z}_n, \theta)$
- If $p(\boldsymbol{z}|\phi)$ and $p(\boldsymbol{x}|\boldsymbol{z}, \theta)$ are Gaussians and $\boldsymbol{z}$ to $\boldsymbol{x}$ map linear $\Rightarrow$ factor analysis or probabilistic PCA

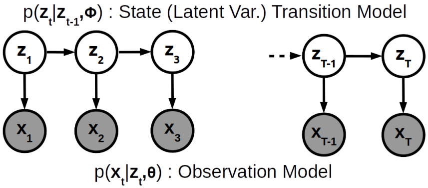
Other Examples..

- State-space latent variable models (e.g., Hidden Markov Models, Kalman Filters)

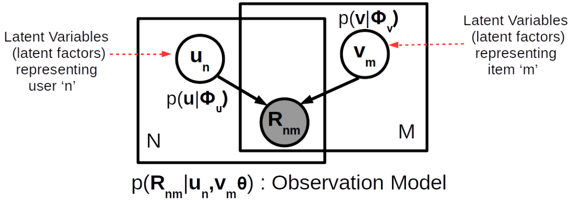


Other Examples..

- State-space latent variable models (e.g., Hidden Markov Models, Kalman Filters)

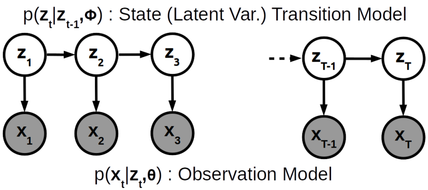


- Latent variable models for "relational data" (e.g., ratings matrix, graph, etc.)

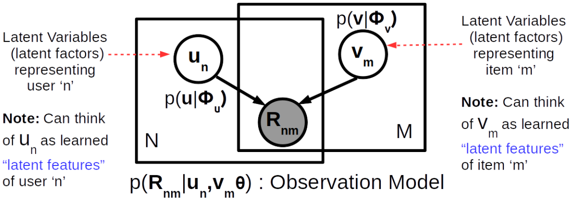


Other Examples..

- State-space latent variable models (e.g., Hidden Markov Models, Kalman Filters)

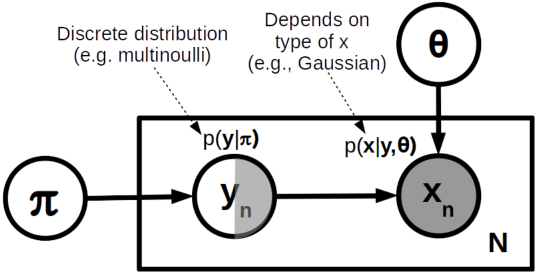


- Latent variable models for "relational data" (e.g., ratings matrix, graph, etc.)



Other Examples..

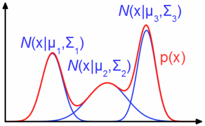
- Semi-supervised generative classification: Some training inputs can be unlabeled
  - These "missing" labels can be treated as latent variables and inferred



Latent Variable Models for Clustering and Density Estimation

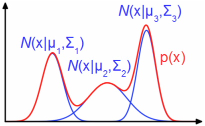
Gaussian Mixture Model (GMM)

- A generative model for data clustering. Assume data generated from a mixture of K Gaussians



Gaussian Mixture Model (GMM)

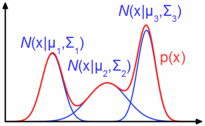
- A generative model for data clustering. Assume data generated from a mixture of K Gaussians



- Let $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$ with $\sum_{k=1}^{K} \pi_k = 1$ denote the "mixing weights" of the K Gaussians

Gaussian Mixture Model (GMM)

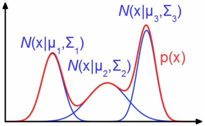
- A generative model for data clustering. Assume data generated from a mixture of K Gaussians



- Let $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$ with $\sum_{k=1}^{K} \pi_k = 1$ denote the "mixing weights" of the K Gaussians
- Intuitively $\pi_k \in (0, 1)$ is the fraction of data "contributed" by the k-th Gaussian

Gaussian Mixture Model (GMM)

- A generative model for data clustering. Assume data generated from a mixture of K Gaussians

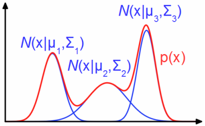


- Let $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$ with $\sum_{k=1}^{K} \pi_k = 1$ denote the "mixing weights" of the K Gaussians
- Intuitively $\pi_k \in (0, 1)$ is the fraction of data "contributed" by the k-th Gaussian
- π_k is the prior probability than $\boldsymbol{x}_n$ comes from the k-th Gaussian (i.e., cluster id $\boldsymbol{z}_n = k$)

$$p(\boldsymbol{z}_n = k|\pi) = \pi_k$$

Gaussian Mixture Model (GMM)

- A generative model for data clustering. Assume data generated from a mixture of K Gaussians



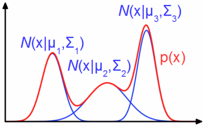
- Let $\pi = (\pi_1, \pi_2, \dots, \pi_K)$ with $\sum_{k=1}^{K} \pi_k = 1$ denote the "mixing weights" of the K Gaussians
- Intuitively $\pi_k \in (0,1)$ is the fraction of data "contributed" by the k-th Gaussian
- π_k is the prior probability than $\boldsymbol{x}_n$ comes from the k-th Gaussian (i.e., cluster id $\boldsymbol{z}_n = k$)

$$p(\boldsymbol{z}_n = k|\pi) = \pi_k$$

- Same as a multinoulli prior on $\boldsymbol{z}$, i.e., $p(\boldsymbol{z}_n|\pi) = \prod_{k=1}^{K} \pi_k^{z_{nk}}$

Gaussian Mixture Model (GMM)

- A generative model for data clustering. Assume data generated from a mixture of K Gaussians



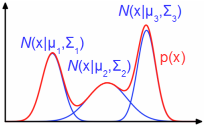
- Let $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$ with $\sum_{k=1}^{K} \pi_k = 1$ denote the "mixing weights" of the K Gaussians
- Intuitively $\pi_k \in (0, 1)$ is the fraction of data "contributed" by the k-th Gaussian
- π_k is the prior probability than $\boldsymbol{x}_n$ comes from the k-th Gaussian (i.e., cluster id $\boldsymbol{z}_n = k$)

$$p(\boldsymbol{z}_n = k|\pi) = \pi_k$$

- Same as a multinoulli prior on $\boldsymbol{z}$, i.e., $p(\boldsymbol{z}_n|\pi) = \prod_{k=1}^{K} \pi_k^{z_{nk}}$ (note $z_{nk} = 1$ if $\boldsymbol{z}_n = k$; 0 otherwise)

Gaussian Mixture Model (GMM)

- A generative model for data clustering. Assume data generated from a mixture of K Gaussians



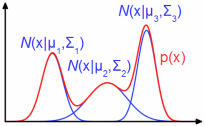
- Let $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$ with $\sum_{k=1}^K \pi_k = 1$ denote the "mixing weights" of the K Gaussians
- Intuitively $\pi_k \in (0, 1)$ is the fraction of data "contributed" by the k-th Gaussian
- π_k is the prior probability than $\boldsymbol{x}_n$ comes from the k-th Gaussian (i.e., cluster id $\boldsymbol{z}_n = k$)

$$p(\boldsymbol{z}_n = k|\pi) = \pi_k$$

- Same as a multinoulli prior on $\boldsymbol{z}$, i.e., $p(\boldsymbol{z}_n|\pi) = \prod_{k=1}^K \pi_k^{z_{nk}}$ (note $z_{nk} = 1$ if $\boldsymbol{z}_n = k$; 0 otherwise)
- Given $\boldsymbol{z}_n = k$, we generate $\boldsymbol{x}_n$ from the k-th Gaussian.

Gaussian Mixture Model (GMM)

- A generative model for data clustering. Assume data generated from a mixture of K Gaussians



- Let $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$ with $\sum_{k=1}^{K} \pi_k = 1$ denote the "mixing weights" of the K Gaussians
- Intuitively $\pi_k \in (0,1)$ is the fraction of data "contributed" by the k-th Gaussian
- π_k is the prior probability than $\boldsymbol{x}_n$ comes from the k-th Gaussian (i.e., cluster id $\boldsymbol{z}_n = k$)

$$p(\boldsymbol{z}_n = k|\pi) = \pi_k$$

- Same as a multinoulli prior on $\boldsymbol{z}$, i.e., $p(\boldsymbol{z}_n|\pi) = \prod_{k=1}^{K} \pi_k^{z_{nk}}$ (note $z_{nk} = 1$ if $\boldsymbol{z}_n = k$; 0 otherwise)
- Given $\boldsymbol{z}_n = k$, we generate $\boldsymbol{x}_n$ from the k-th Gaussian. Thus

$$p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

Gaussian Mixture Model (GMM)

- Recall that $p(\boldsymbol{z}_n = k|\pi) = \pi_k$ and the conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$

Gaussian Mixture Model (GMM)

- Recall that $p(\boldsymbol{z}_n = k|\pi) = \pi_k$ and the <u>conditional</u> distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$
- What is the <u>marginal</u> distribution of $\boldsymbol{x}_n$, i.e., $p(\boldsymbol{x}_n)$?

Gaussian Mixture Model (GMM)

- Recall that $p(\boldsymbol{z}_n = k|\pi) = \pi_k$ and the conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$
- What is the marginal distribution of $\boldsymbol{x}_n$, i.e., $p(\boldsymbol{x}_n)$?
  - To get that, we must sum over all possibilities of $\boldsymbol{z}_n$

$$p(\boldsymbol{x}_n) = \sum_{k=1}^{K} p(\boldsymbol{x}_n, \boldsymbol{z}_n = k)$$

Gaussian Mixture Model (GMM)

- Recall that $p(\boldsymbol{z}_n = k|\pi) = \pi_k$ and the conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$
- What is the marginal distribution of $\boldsymbol{x}_n$, i.e., $p(\boldsymbol{x}_n)$?
    - To get that, we must sum over all possibilities of $\boldsymbol{z}_n$

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

Gaussian Mixture Model (GMM)

- Recall that $p(\boldsymbol{z}_n = k|\pi) = \pi_k$ and the conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$
- What is the marginal distribution of $\boldsymbol{x}_n$, i.e., $p(\boldsymbol{x}_n)$?
  - To get that, we must sum over all possibilities of $\boldsymbol{z}_n$

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

  - Therefore the marginal distribution of $\boldsymbol{x}_n$ will be

Gaussian Mixture Model (GMM)

- Recall that $p(\boldsymbol{z}_n = k|\pi) = \pi_k$ and the conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$
- What is the marginal distribution of $\boldsymbol{x}_n$, i.e., $p(\boldsymbol{x}_n)$?
  - To get that, we must sum over all possibilities of $\boldsymbol{z}_n$

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

  - Therefore the marginal distribution of $\boldsymbol{x}_n$ will be

$$\boxed{p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)}$$

Gaussian Mixture Model (GMM)

- Recall that $p(\boldsymbol{z}_n = k|\pi) = \pi_k$ and the conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$
- What is the marginal distribution of $\boldsymbol{x}_n$, i.e., $p(\boldsymbol{x}_n)$?
  - To get that, we must sum over all possibilities of $\boldsymbol{z}_n$

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

  - Therefore the marginal distribution of $\boldsymbol{x}_n$ will be

$$\boxed{p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)}$$

where Θ collectively denotes all the parameters of the GMM model $(\pi, \{\mu_k, \Sigma_k\}_{k=1}^{K})$

Gaussian Mixture Model (GMM)

- Recall that $p(\boldsymbol{z}_n = k|\pi) = \pi_k$ and the conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$
- What is the marginal distribution of $\boldsymbol{x}_n$, i.e., $p(\boldsymbol{x}_n)$?
  - To get that, we must sum over all possibilities of $\boldsymbol{z}_n$

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

  - Therefore the marginal distribution of $\boldsymbol{x}_n$ will be

$$\boxed{p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)}$$

  where Θ collectively denotes all the parameters of the GMM model $(\pi, \{\mu_k, \Sigma_k\}_{k=1}^{K})$

- Note that GMM also defines probability density (mixture of Gaussians) for the inputs. Therefore it can be used as both a density estimation model as well as a clustering model

Gaussian Mixture Model (GMM)

- Recall that $p(\boldsymbol{z}_n = k|\pi) = \pi_k$ and the conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$
- What is the marginal distribution of $\boldsymbol{x}_n$, i.e., $p(\boldsymbol{x}_n)$?
  - To get that, we must sum over all possibilities of $\boldsymbol{z}_n$

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

  - Therefore the marginal distribution of $\boldsymbol{x}_n$ will be

$$\boxed{p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)}$$

  where Θ collectively denotes all the parameters of the GMM model $(\pi, \{\mu_k, \Sigma_k\}_{k=1}^{K})$

- Note that GMM also defines probability density (mixture of Gaussians) for the inputs. Therefore it can be used as both a density estimation model as well as a clustering model
- Goal: Learn the GMM parameters $\Theta = (\pi, \{\mu_k, \Sigma_k\}_{k=1}^{K})$ (and cluster assignments $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$)

Gaussian Mixture Model (GMM)

- Recall the marginal distribution of $\boldsymbol{x}_n$

$$p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

Gaussian Mixture Model (GMM)

- Recall the marginal distribution of $\boldsymbol{x}_n$

$$p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

where $\Theta = (\pi, \{\mu_k, \Sigma_k\}_{k=1}^{K})$

Gaussian Mixture Model (GMM)

- Recall the marginal distribution of $\boldsymbol{x}_n$

$$p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

  where $\Theta = (\pi, \{\mu_k, \Sigma_k\}_{k=1}^K)$

- Suppose we want to do MLE for Θ. The MLE objective will be

$$\sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\Theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

- Doing MLE on this objective is tricky due to the "$\log \sum$" term

Gaussian Mixture Model (GMM)

- Recall the marginal distribution of $\boldsymbol{x}_n$

$$p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

where $\Theta = (\pi, \{\mu_k, \Sigma_k\}_{k=1}^{K})$

- Suppose we want to do MLE for Θ. The MLE objective will be

$$\sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\Theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

- Doing MLE on this objective is tricky due to the "$\log \sum$" term
  - Parameters get coupled; no closed form solution (iterative methods needed, slow convergence)

Gaussian Mixture Model (GMM)

- Recall the marginal distribution of $\boldsymbol{x}_n$

$$p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

  where $\Theta = (\pi, \{\mu_k, \Sigma_k\}_{k=1}^{K})$

- Suppose we want to do MLE for Θ. The MLE objective will be

$$\sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\Theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

- Doing MLE on this objective is tricky due to the "$\log \sum$" term
  - Parameters get coupled; no closed form solution (iterative methods needed, slow convergence)
  - Expectation Maximization (EM) helps solve such problems in a clean and efficient way

Expectation Maximization (EM)

- A general algorithm for doing MLE/MAP in models that contain latent variables (e.g., GMM)

Expectation Maximization (EM)

- A general algorithm for doing MLE/MAP in models that contain latent variables (e.g., GMM)
- The basic (rough) idea is to do MLE/MAP not on $p(\boldsymbol{x}|\Theta)$ but on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$

Expectation Maximization (EM)

- A general algorithm for doing MLE/MAP in models that contain latent variables (e.g., GMM)
- The basic (rough) idea is to do MLE/MAP not on $p(\boldsymbol{x}|\Theta)$ but on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$
- EM thinks of not just $\boldsymbol{x}$ as the data but both $\boldsymbol{x}$ and a good "guess" of $\boldsymbol{z}$ as the data

Expectation Maximization (EM)

- A general algorithm for doing MLE/MAP in models that contain latent variables (e.g., GMM)
- The basic (rough) idea is to do MLE/MAP not on $p(\boldsymbol{x}|\Theta)$ but on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$
- EM thinks of not just $\boldsymbol{x}$ as the data but both $\boldsymbol{x}$ and a good "guess" of $\boldsymbol{z}$ as the data
- Key idea: Once $\boldsymbol{z}$ is known, the MLE/MAP becomes very simple

Expectation Maximization (EM)

- A general algorithm for doing MLE/MAP in models that contain latent variables (e.g., GMM)
- The basic (rough) idea is to do MLE/MAP not on $p(\boldsymbol{x}|\Theta)$ but on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$
- EM thinks of not just $\boldsymbol{x}$ as the data but both $\boldsymbol{x}$ and a good "guess" of $\boldsymbol{z}$ as the data
- Key idea: Once $\boldsymbol{z}$ is known, the MLE/MAP becomes very simple
    - E.g., for GMM, once we know the "labels" $\boldsymbol{z}$, it reduces to estimating K Gaussians independently

Expectation Maximization (EM)

- A general algorithm for doing MLE/MAP in models that contain latent variables (e.g., GMM)
- The basic (rough) idea is to do MLE/MAP not on $p(\boldsymbol{x}|\Theta)$ but on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$
- EM thinks of not just $\boldsymbol{x}$ as the data but both $\boldsymbol{x}$ and a good "guess" of $\boldsymbol{z}$ as the data
- Key idea: Once $\boldsymbol{z}$ is known, the MLE/MAP becomes very simple
  - E.g., for GMM, once we know the "labels" $\boldsymbol{z}$, it reduces to estimating K Gaussians independently
- EM alternates between the following two steps until convergence

Expectation Maximization (EM)

- A general algorithm for doing MLE/MAP in models that contain latent variables (e.g., GMM)
- The basic (rough) idea is to do MLE/MAP not on $p(\boldsymbol{x}|\Theta)$ but on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$
- EM thinks of not just $\boldsymbol{x}$ as the data but both $\boldsymbol{x}$ and a good "guess" of $\boldsymbol{z}$ as the data
- Key idea: Once $\boldsymbol{z}$ is known, the MLE/MAP becomes very simple
    - E.g., for GMM, once we know the "labels" $\boldsymbol{z}$, it reduces to estimating K Gaussians independently
- EM alternates between the following two steps until convergence
    - Given the current estimate of Θ, make a good "guess" for $\boldsymbol{z}$ for each $\boldsymbol{x}$

Expectation Maximization (EM)

- A general algorithm for doing MLE/MAP in models that contain latent variables (e.g., GMM)
- The basic (rough) idea is to do MLE/MAP not on $p(\boldsymbol{x}|\Theta)$ but on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$
- EM thinks of not just $\boldsymbol{x}$ as the data but both $\boldsymbol{x}$ and a good "guess" of $\boldsymbol{z}$ as the data
- Key idea: Once $\boldsymbol{z}$ is known, the MLE/MAP becomes very simple
  - E.g., for GMM, once we know the "labels" $\boldsymbol{z}$, it reduces to estimating K Gaussians independently
- EM alternates between the following two steps until convergence
  - Given the current estimate of Θ, make a good "guess" for $\boldsymbol{z}$ for each $\boldsymbol{x}$
  - Use these guesses of $\boldsymbol{z}$ to do MLE/MAP on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$ to "refine" Θ

Expectation Maximization (EM)

- A general algorithm for doing MLE/MAP in models that contain latent variables (e.g., GMM)
- The basic (rough) idea is to do MLE/MAP not on $p(\boldsymbol{x}|\Theta)$ but on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$
- EM thinks of not just $\boldsymbol{x}$ as the data but both $\boldsymbol{x}$ and a good "guess" of $\boldsymbol{z}$ as the data
- Key idea: Once $\boldsymbol{z}$ is known, the MLE/MAP becomes very simple
  - E.g., for GMM, once we know the "labels" $\boldsymbol{z}$, it reduces to <u>estimating K Gaussians independently</u>
- EM alternates between the following two steps until convergence
  - Given the current estimate of Θ, make a good "guess" for $\boldsymbol{z}$ for each $\boldsymbol{x}$
  - Use these guesses of $\boldsymbol{z}$ to do MLE/MAP on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$ to "refine" Θ
- Important: EM leads to very easy param. updates if $p(\boldsymbol{x}|\boldsymbol{z})$ and $p(\boldsymbol{z})$ are exp. family distributions

Expectation Maximization (EM)

- A general algorithm for doing MLE/MAP in models that contain latent variables (e.g., GMM)
- The basic (rough) idea is to do MLE/MAP not on $p(\boldsymbol{x}|\Theta)$ but on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$
- EM thinks of not just $\boldsymbol{x}$ as the data but both $\boldsymbol{x}$ and a good "guess" of $\boldsymbol{z}$ as the data
- Key idea: Once $\boldsymbol{z}$ is known, the MLE/MAP becomes very simple
  - E.g., for GMM, once we know the "labels" $\boldsymbol{z}$, it reduces to <u>estimating K Gaussians independently</u>
- EM alternates between the following two steps until convergence
  - Given the current estimate of Θ, make a good "guess" for $\boldsymbol{z}$ for each $\boldsymbol{x}$
  - Use these guesses of $\boldsymbol{z}$ to do MLE/MAP on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$ to "refine" Θ
- Important: EM leads to very easy param. updates if $p(\boldsymbol{x}|\boldsymbol{z})$ and $p(\boldsymbol{z})$ are exp. family distributions
- EM is guaranteed to converge to a local optima

Expectation Maximization (EM)

- A general algorithm for doing MLE/MAP in models that contain latent variables (e.g., GMM)
- The basic (rough) idea is to do MLE/MAP not on $p(\boldsymbol{x}|\Theta)$ but on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$
- EM thinks of not just $\boldsymbol{x}$ as the data but both $\boldsymbol{x}$ and a good "guess" of $\boldsymbol{z}$ as the data
- Key idea: Once $\boldsymbol{z}$ is known, the MLE/MAP becomes very simple
  - E.g., for GMM, once we know the "labels" $\boldsymbol{z}$, it reduces to estimating K Gaussians independently
- EM alternates between the following two steps until convergence
  - Given the current estimate of Θ, make a good "guess" for $\boldsymbol{z}$ for each $\boldsymbol{x}$
  - Use these guesses of $\boldsymbol{z}$ to do MLE/MAP on $p(\boldsymbol{x}, \boldsymbol{z}|\Theta)$ to "refine" Θ
- Important: EM leads to very easy param. updates if $p(\boldsymbol{x}|\boldsymbol{z})$ and $p(\boldsymbol{z})$ are exp. family distributions
- EM is guaranteed to converge to a local optima
- More details in the next class..