Introduction to Generative Models

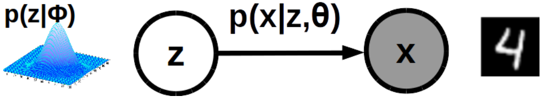
Piyush Rai

Machine Learning (CS771A)

Sept 23, 2016

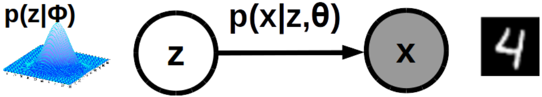
Generative Model

- Defines a probabilistic way that could have “generated” the data



Generative Model

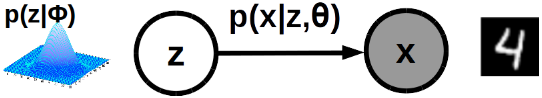
- Defines a probabilistic way that could have "generated" the data



- Each observation $\boldsymbol{x}_n$ is assumed to be associated with a latent variable $\boldsymbol{z}_n$ (we can think of $\boldsymbol{z}_n$ as a compact/compressed "encoding" of $\boldsymbol{x}_n$)

Generative Model

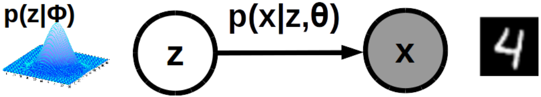
- Defines a probabilistic way that could have "generated" the data



- Each observation $\boldsymbol{x}_n$ is assumed to be associated with a latent variable $\boldsymbol{z}_n$ (we can think of $\boldsymbol{z}_n$ as a compact/compressed "encoding" of $\boldsymbol{x}_n$)
- $\boldsymbol{z}_n$ is assumed to be a random variable with some prior distribution $p(\boldsymbol{z}|\phi)$

Generative Model

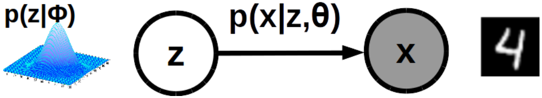
- Defines a probabilistic way that could have "generated" the data



- Each observation $\boldsymbol{x}_n$ is assumed to be associated with a latent variable $\boldsymbol{z}_n$ (we can think of $\boldsymbol{z}_n$ as a compact/compressed "encoding" of $\boldsymbol{x}_n$)
- $\boldsymbol{z}_n$ is assumed to be a random variable with some prior distribution $p(\boldsymbol{z}|\phi)$
- Assume another data distribution $p(\boldsymbol{x}|\boldsymbol{z}, \theta)$ that can "generate" $\boldsymbol{x}$ given $\boldsymbol{z}$

Generative Model

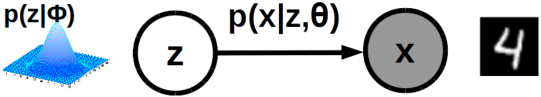
- Defines a probabilistic way that could have "generated" the data



- Each observation $\boldsymbol{x}_n$ is assumed to be associated with a latent variable $\boldsymbol{z}_n$ (we can think of $\boldsymbol{z}_n$ as a compact/compressed "encoding" of $\boldsymbol{x}_n$)
- $\boldsymbol{z}_n$ is assumed to be a random variable with some prior distribution $p(\boldsymbol{z}|\phi)$
- Assume another data distribution $p(\boldsymbol{x}|\boldsymbol{z},\theta)$ that can "generate" $\boldsymbol{x}$ given $\boldsymbol{z}$
- What $\boldsymbol{x}$ and $\boldsymbol{z}$ "look like", and the form of the distributions $p(\boldsymbol{z}|\phi), p(\boldsymbol{x}|\boldsymbol{z},\theta)$ will be problem-specific (we will soon look at some examples)

Generative Model

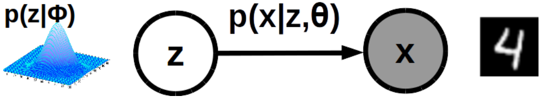
- Defines a probabilistic way that could have "generated" the data



- Each observation $\boldsymbol{x}_n$ is assumed to be associated with a latent variable $\boldsymbol{z}_n$ (we can think of $\boldsymbol{z}_n$ as a compact/compressed "encoding" of $\boldsymbol{x}_n$)
- $\boldsymbol{z}_n$ is assumed to be a random variable with some prior distribution $p(\boldsymbol{z}|\phi)$
- Assume another data distribution $p(\boldsymbol{x}|\boldsymbol{z},\theta)$ that can "generate" $\boldsymbol{x}$ given $\boldsymbol{z}$
- What $\boldsymbol{x}$ and $\boldsymbol{z}$ "look like", and the form of the distributions $p(\boldsymbol{z}|\phi), p(\boldsymbol{x}|\boldsymbol{z},\theta)$ will be problem-specific (we will soon look at some examples)
- $\{\theta,\phi\}$ are the unknown model parameters

Generative Model

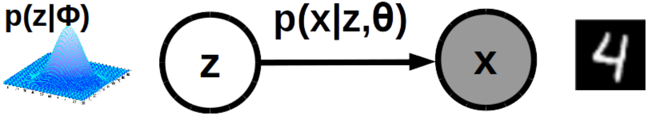
- Defines a probabilistic way that could have "generated" the data



- Each observation $\boldsymbol{x}_n$ is assumed to be associated with a latent variable $\boldsymbol{z}_n$ (we can think of $\boldsymbol{z}_n$ as a compact/compressed "encoding" of $\boldsymbol{x}_n$)
- $\boldsymbol{z}_n$ is assumed to be a random variable with some prior distribution $p(\boldsymbol{z}|\phi)$
- Assume another data distribution $p(\boldsymbol{x}|\boldsymbol{z},\theta)$ that can "generate" $\boldsymbol{x}$ given $\boldsymbol{z}$
- What $\boldsymbol{x}$ and $\boldsymbol{z}$ "look like", and the form of the distributions $p(\boldsymbol{z}|\phi), p(\boldsymbol{x}|\boldsymbol{z},\theta)$ will be problem-specific (we will soon look at some examples)
- $\{\theta, \phi\}$ are the unknown model parameters
- The goal will be to learn $\{\theta, \phi\}$ and $\boldsymbol{z}_n$'s, given the observed data

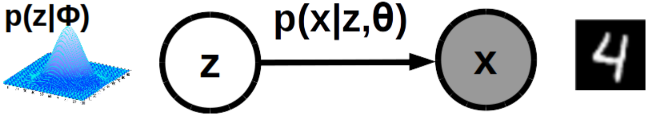
"Generative Story" of Data

- Generative models can be described using a "generative story" for the data



"Generative Story" of Data

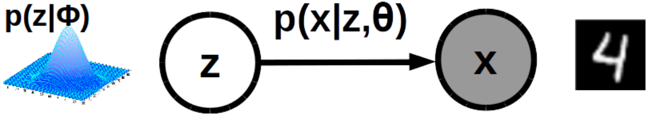
- Generative models can be described using a "generative story" for the data



- The "generative story" of each observation $\boldsymbol{x}_n$, $\forall n$

"Generative Story" of Data

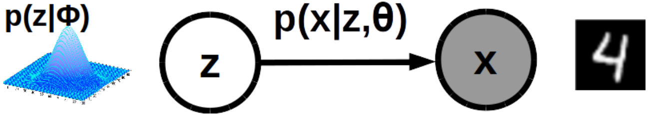
- Generative models can be described using a "generative story" for the data



- The "generative story" of each observation $\boldsymbol{x}_n$, $\forall n$
  - First draw a random latent variable $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$ from the prior on $\boldsymbol{z}$

"Generative Story" of Data

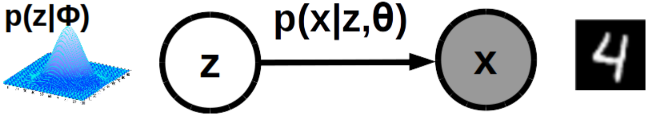
- Generative models can be described using a "generative story" for the data



- The "generative story" of each observation $\boldsymbol{x}_n$, $\forall n$
  - First draw a random latent variable $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$ from the prior on $\boldsymbol{z}$
  - Given $\boldsymbol{z}_n$, now generate $\boldsymbol{x}_n$ as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta, \boldsymbol{z}_n)$ from the data distribution

"Generative Story" of Data

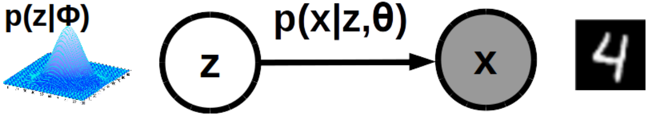
- Generative models can be described using a "generative story" for the data



- The "generative story" of each observation $\boldsymbol{x}_n$, $\forall n$
  - First draw a random latent variable $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$ from the prior on $\boldsymbol{z}$
  - Given $\boldsymbol{z}_n$, now generate $\boldsymbol{x}_n$ as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta, \boldsymbol{z}_n)$ from the data distribution
- Such models usually have two types of variables: "local" and "global"

“Generative Story” of Data

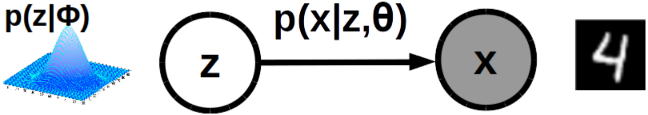
- Generative models can be described using a “generative story” for the data



- The “generative story” of each observation $\boldsymbol{x}_n$, $\forall n$
  - First draw a random latent variable $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$ from the prior on $\boldsymbol{z}$
  - Given $\boldsymbol{z}_n$, now generate $\boldsymbol{x}_n$ as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta, \boldsymbol{z}_n)$ from the data distribution
- Such models usually have two types of variables: “local” and “global”
  - Each $\boldsymbol{z}_n$ is a “local” variable (specific to the data point $\boldsymbol{x}_n$)

"Generative Story" of Data

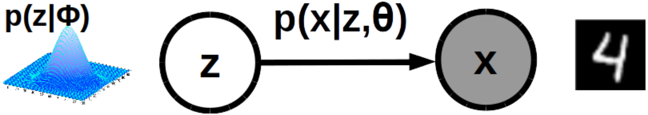
- Generative models can be described using a "generative story" for the data



- The "generative story" of each observation $\boldsymbol{x}_n$, $\forall n$
  - First draw a random latent variable $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$ from the prior on $\boldsymbol{z}$
  - Given $\boldsymbol{z}_n$, now generate $\boldsymbol{x}_n$ as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta, \boldsymbol{z}_n)$ from the data distribution
- Such models usually have two types of variables: "local" and "global"
  - Each $\boldsymbol{z}_n$ is a "local" variable (specific to the data point $\boldsymbol{x}_n$)
  - (ϕ, θ) are global variables (shared by all the data points)

"Generative Story" of Data

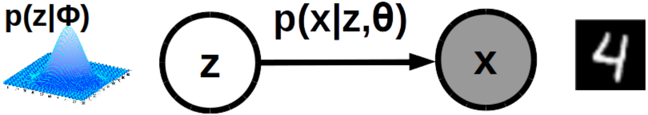
- Generative models can be described using a "generative story" for the data



- The "generative story" of each observation $\boldsymbol{x}_n$, $\forall n$
  - First draw a random latent variable $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$ from the prior on $\boldsymbol{z}$
  - Given $\boldsymbol{z}_n$, now generate $\boldsymbol{x}_n$ as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta, \boldsymbol{z}_n)$ from the data distribution
- Such models usually have two types of variables: "local" and "global"
  - Each $\boldsymbol{z}_n$ is a "local" variable (specific to the data point $\boldsymbol{x}_n$)
  - (ϕ, θ) are global variables (shared by all the data points)
  - We may be interested in learning the global vars, or local vars, or both

“Generative Story” of Data

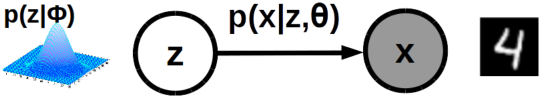
- Generative models can be described using a “generative story” for the data



- The “generative story” of each observation $\boldsymbol{x}_n$, $\forall n$
  - First draw a random latent variable $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$ from the prior on $\boldsymbol{z}$
  - Given $\boldsymbol{z}_n$, now generate $\boldsymbol{x}_n$ as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta, \boldsymbol{z}_n)$ from the data distribution
- Such models usually have two types of variables: “local” and “global”
  - Each $\boldsymbol{z}_n$ is a “local” variable (specific to the data point $\boldsymbol{x}_n$)
  - (ϕ, θ) are global variables (shared by all the data points)
  - We may be interested in learning the global vars, or local vars, or both
  - Usually it’s possible to infer the global vars from local vars (or vice-versa)

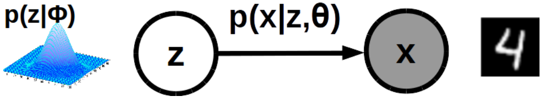
Why Generative Models?

- A proper, probabilistic way to think about the data generation process



Why Generative Models?

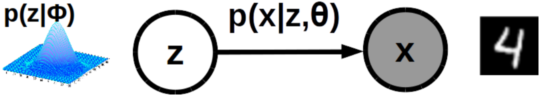
- A proper, probabilistic way to think about the data generation process



- Allows modeling different types of data (real, binary, count, etc.) by changing the data distribution $p(\boldsymbol{x}|\theta, \boldsymbol{z})$ appropriately

Why Generative Models?

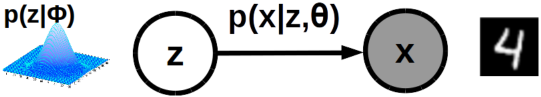
- A proper, probabilistic way to think about the data generation process



- Allows modeling different types of data (real, binary, count, etc.) by changing the data distribution $p(\boldsymbol{x}|\theta, \boldsymbol{z})$ appropriately
- Can synthesize or "hallucinate" new data using an already learned model
  - Generate a "random" $\boldsymbol{z}$ from $p(\boldsymbol{z}|\phi)$ and generate $\boldsymbol{x}$ from $p(\boldsymbol{x}|\theta, \boldsymbol{z})$

Why Generative Models?

- A proper, probabilistic way to think about the data generation process



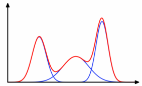
- Allows modeling different types of data (real, binary, count, etc.) by changing the data distribution $p(\boldsymbol{x}|\theta, \boldsymbol{z})$ appropriately
- Can synthesize or "hallucinate" new data using an already learned model
  - Generate a "random" $\boldsymbol{z}$ from $p(\boldsymbol{z}|\phi)$ and generate $\boldsymbol{x}$ from $p(\boldsymbol{x}|\theta, \boldsymbol{z})$

- Allows handling missing data (by treating missing data also as latent variable)

Hallucinated faces pic courtesy: `http://torch.ch/blog/2015/11/13/gan.html`

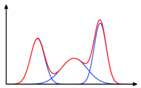
Some "Canonical" Generative Models

- Mixture model (used in clustering and probability density estimation)

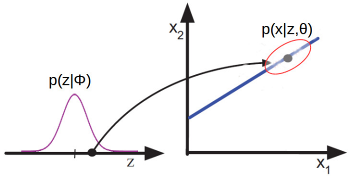


Some "Canonical" Generative Models

- Mixture model (used in clustering and probability density estimation)

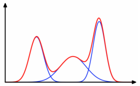


- Latent factor model (used in dimensionality reduction)

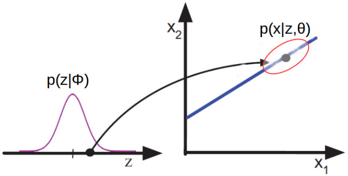


Some "Canonical" Generative Models

- Mixture model (used in clustering and probability density estimation)



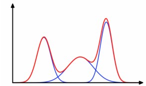
- Latent factor model (used in dimensionality reduction)



- Can even combine these (e.g., mixture of latent factor models)

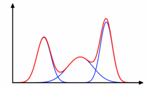
Example: Mixture Model

- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions



Example: Mixture Model

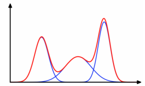
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions



- Suppose these K distributions are $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$

Example: Mixture Model

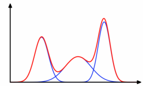
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions



- Suppose these K distributions are $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$
- Don't know which of the K distributions each $\boldsymbol{x}_n$ was generated from

Example: Mixture Model

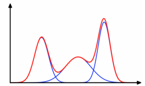
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions



- Suppose these K distributions are $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$
- Don't know which of the K distributions each $\boldsymbol{x}_n$ was generated from
- Consider the following generative story for each $\boldsymbol{x}_n$, $n = 1, 2, \ldots, N$

Example: Mixture Model

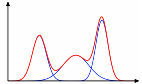
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions



- Suppose these K distributions are $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$
- Don't know which of the K distributions each $\boldsymbol{x}_n$ was generated from
- Consider the following 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 $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$

Example: Mixture Model

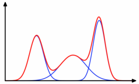
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions



- Suppose these K distributions are $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$
- Don't know which of the K distributions each $\boldsymbol{x}_n$ was generated from
- Consider the following 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 $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$
  - Now generate $\boldsymbol{x}_n$ from the mixture component no. $\boldsymbol{z}_n$ as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta_{\boldsymbol{z}_n})$

Example: Mixture Model

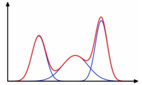
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions



- Suppose these K distributions are $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$
- Don't know which of the K distributions each $\boldsymbol{x}_n$ was generated from
- Consider the following 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 $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$
  - Now generate $\boldsymbol{x}_n$ from the mixture component no. $\boldsymbol{z}_n$ as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta_{\boldsymbol{z}_n})$
- In a mixture model, $\boldsymbol{z}$ is discrete so $p(\boldsymbol{z}|\phi)$ is a multinomial distribution

Example: Mixture Model

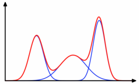
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions



- Suppose these K distributions are $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$
- Don't know which of the K distributions each $\boldsymbol{x}_n$ was generated from
- Consider the following 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 $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$
  - Now generate $\boldsymbol{x}_n$ from the mixture component no. $\boldsymbol{z}_n$ as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta_{\boldsymbol{z}_n})$
- In a mixture model, $\boldsymbol{z}$ is discrete so $p(\boldsymbol{z}|\phi)$ is a multinomial distribution
- The data distribution $p(\boldsymbol{x}|\theta_{\boldsymbol{z}_n})$ depends on the type of data being modeled

Example: Mixture Model

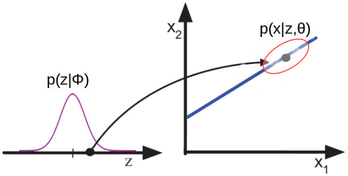
- Assume data $\{\boldsymbol{x}_n\}_{n=1}^N$ was generated from a mixture of K distributions



- Suppose these K distributions are $p(\boldsymbol{x}|\theta_1), \ldots, p(\boldsymbol{x}|\theta_K)$
- Don't know which of the K distributions each $\boldsymbol{x}_n$ was generated from
- Consider the following 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 $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$
  - Now generate $\boldsymbol{x}_n$ from the mixture component no. $\boldsymbol{z}_n$ as $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\theta_{\boldsymbol{z}_n})$
- In a mixture model, $\boldsymbol{z}$ is discrete so $p(\boldsymbol{z}|\phi)$ is a multinomial distribution
- The data distribution $p(\boldsymbol{x}|\theta_{\boldsymbol{z}_n})$ depends on the type of data being modeled
- Mixture models can model complex distributions as superposition of simpler distributions (can be used for density estimation, as well as clustering).

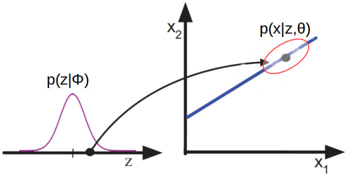
Example: Latent Factor Model

- Assume each D-dim $\boldsymbol{x}_n$ generated from a K-dim latent factor $\boldsymbol{z}_n$ ($K \ll D$)



Example: Latent Factor Model

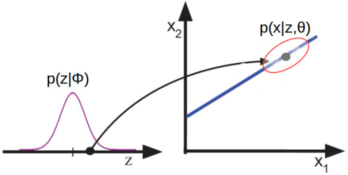
- Assume each D-dim $\boldsymbol{x}_n$ generated from a K-dim latent factor $\boldsymbol{z}_n$ ($K \ll D$)



- Consider the following generative story for each $\boldsymbol{x}_n$, $n = 1, 2, \ldots, N$

Example: Latent Factor Model

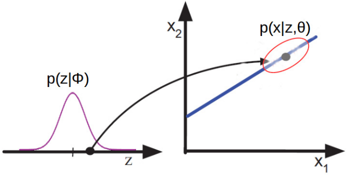
- Assume each D-dim $\boldsymbol{x}_n$ generated from a K-dim latent factor $\boldsymbol{z}_n$ ($K \ll D$)



- 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 $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\phi)$

Example: Latent Factor Model

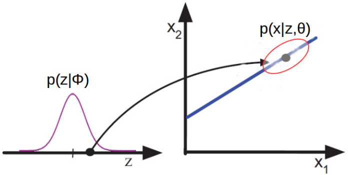
- Assume each D-dim $\boldsymbol{x}_n$ generated from a K-dim latent factor $\boldsymbol{z}_n$ $(K \ll D)$



- 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 $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\boldsymbol{z}_n, \theta)$

Example: Latent Factor Model

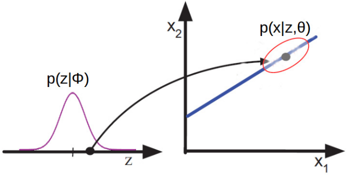
- Assume each D-dim $\boldsymbol{x}_n$ generated from a K-dim latent factor $\boldsymbol{z}_n$ ($K \ll D$)



- 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 $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\boldsymbol{z}_n, \theta)$
- When $p(\boldsymbol{z}|\phi)$ and $p(\boldsymbol{x}|\boldsymbol{z}_n, \theta)$ are Gaussian distributions, this basic generative model is called factor analysis or probabilistic PCA

Example: Latent Factor Model

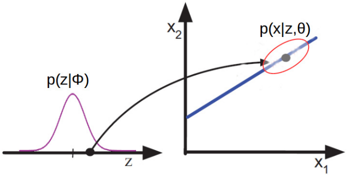
- Assume each D-dim $\boldsymbol{x}_n$ generated from a K-dim latent factor $\boldsymbol{z}_n$ ($K \ll D$)



- 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 $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\boldsymbol{z}_n, \theta)$
- When $p(\boldsymbol{z}|\phi)$ and $p(\boldsymbol{x}|\boldsymbol{z}_n, \theta)$ are Gaussian distributions, this basic generative model is called factor analysis or probabilistic PCA
- The choice of $p(\boldsymbol{z}|\phi)$ and $p(\boldsymbol{x}|\boldsymbol{z}_n, \theta)$ in general will be problem dependent

Example: Latent Factor Model

- Assume each D-dim $\boldsymbol{x}_n$ generated from a K-dim latent factor $\boldsymbol{z}_n$ ($K \ll D$)



- 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 $\boldsymbol{x}_n \sim p(\boldsymbol{x}|\boldsymbol{z}_n, \theta)$
- When $p(\boldsymbol{z}|\phi)$ and $p(\boldsymbol{x}|\boldsymbol{z}_n, \theta)$ are Gaussian distributions, this basic generative model is called factor analysis or probabilistic PCA
- The choice of $p(\boldsymbol{z}|\phi)$ and $p(\boldsymbol{x}|\boldsymbol{z}_n, \theta)$ in general will be problem dependent
- Many recent advances in generative models (e.g., deep generative models, generative adversarial networks, etc) are based on these basic principles

Going Forward..

- We will look at, in more detail, some specific generative models

Going Forward..

- We will look at, in more detail, some specific generative models
    - Gaussian mixture model (for clustering and density estimation)

Going Forward..

- We will look at, in more detail, some specific generative models
  - Gaussian mixture model (for clustering and density estimation)
  - Factor Analysis and Probabilistic PCA (for dimensionality reduction)

Going Forward..

- We will look at, in more detail, some specific generative models
  - Gaussian mixture model (for clustering and density estimation)
  - Factor Analysis and Probabilistic PCA (for dimensionality reduction)
- We will also look at how to do parameter estimation in such models

Going Forward..

- We will look at, in more detail, some specific generative models
    - Gaussian mixture model (for clustering and density estimation)
    - Factor Analysis and Probabilistic PCA (for dimensionality reduction)
- We will also look at how to do parameter estimation in such models
    - One common approach is to perform MLE/MAP

Going Forward..

- We will look at, in more detail, some specific generative models
  - Gaussian mixture model (for clustering and density estimation)
  - Factor Analysis and Probabilistic PCA (for dimensionality reduction)
- We will also look at how to do parameter estimation in such models
  - One common approach is to perform MLE/MAP
  - However, presence of latent variables $\boldsymbol{z}$ makes MLE/MAP hard

Going Forward..

- We will look at, in more detail, some specific generative models
  - Gaussian mixture model (for clustering and density estimation)
  - Factor Analysis and Probabilistic PCA (for dimensionality reduction)
- We will also look at how to do parameter estimation in such models
  - One common approach is to perform MLE/MAP
  - However, presence of latent variables $\boldsymbol{z}$ makes MLE/MAP hard
  - Reason: Since $\boldsymbol{z}$ is a random variable, we must sum over all possible values of $\boldsymbol{z}$ when doing MLE/MAP for the model parameters θ, ϕ

$$\log p(\boldsymbol{x}|\theta, \phi) = \log \sum_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{z}, \theta) p(\boldsymbol{z}|\phi) \quad \text{(Log can't go inside the summation!)}$$

Going Forward..

- We will look at, in more detail, some specific generative models
  - Gaussian mixture model (for clustering and density estimation)
  - Factor Analysis and Probabilistic PCA (for dimensionality reduction)
- We will also look at how to do parameter estimation in such models
  - One common approach is to perform MLE/MAP
  - However, presence of latent variables $\boldsymbol{z}$ makes MLE/MAP hard
  - Reason: Since $\boldsymbol{z}$ is a random variable, we must sum over all possible values of $\boldsymbol{z}$ when doing MLE/MAP for the model parameters θ, ϕ

$$\log p(\boldsymbol{x}|\theta, \phi) = \log \sum_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{z}, \theta)p(\boldsymbol{z}|\phi) \quad \text{(Log can't go inside the summation!)}$$

  - Expectation Maximization (EM) algorithm gives a way to solve the problem

Going Forward..

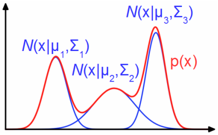
- We will look at, in more detail, some specific generative models
  - Gaussian mixture model (for clustering and density estimation)
  - Factor Analysis and Probabilistic PCA (for dimensionality reduction)
- We will also look at how to do parameter estimation in such models
  - One common approach is to perform MLE/MAP
  - However, presence of latent variables $\boldsymbol{z}$ makes MLE/MAP hard
  - Reason: Since $\boldsymbol{z}$ is a random variable, we must sum over all possible values of $\boldsymbol{z}$ when doing MLE/MAP for the model parameters θ, ϕ

$$\log p(\boldsymbol{x}|\theta,\phi) = \log \sum_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{z},\theta)p(\boldsymbol{z}|\phi) \quad \text{(Log can't go inside the summation!)}$$

  - Expectation Maximization (EM) algorithm gives a way to solve the problem
  - Basic idea in EM: Instead of summing over all possibilities of $\boldsymbol{z}$, make a "guess" $\tilde{\boldsymbol{z}}$ and maximize $\log p(\boldsymbol{x}, \tilde{\boldsymbol{z}}|\theta,\phi)$ w.r.t. θ, ϕ to learn θ, ϕ. Use these values of θ, ϕ to refine your guess $\tilde{\boldsymbol{z}}$ and repeat until convergence.

Gaussian Mixture Model (GMM)

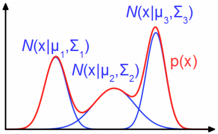
- Assume the data is generated from a mixture of K Gaussians



- Each Gaussian represents a "cluster" in the data

Gaussian Mixture Model (GMM)

- Assume the data is generated from a mixture of K Gaussians

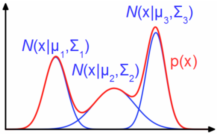


- Each Gaussian represents a "cluster" in the data
- The distribution $p(\boldsymbol{x})$ will be a weighted a mixture of K Gaussians

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

Gaussian Mixture Model (GMM)

- Assume the data is generated from a mixture of K Gaussians



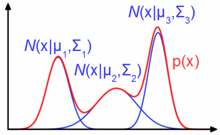
- Each Gaussian represents a "cluster" in the data
- The distribution $p(\boldsymbol{x})$ will be a weighted a mixture of K Gaussians

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

where π_k's are the **mixing weights**: $\sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0$ (intuitively, $\pi_k = p(\boldsymbol{z} = k)$ is the fraction of data generated by the k-th distribution)

Gaussian Mixture Model (GMM)

- Assume the data is generated from a mixture of K Gaussians



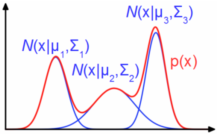
- Each Gaussian represents a "cluster" in the data
- The distribution $p(\boldsymbol{x})$ will be a weighted a mixture of K Gaussians

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

  where π_k's are the **mixing weights**: $\sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0$ (intuitively, $\pi_k = p(\boldsymbol{z} = k)$ is the fraction of data generated by the k-th distribution)
- The goal is to learn the params $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ of these K Gaussians, the mixing weights $\{\pi_k\}_{k=1}^{K}$, and/or the cluster assignment $\boldsymbol{z}_n$ of each $\boldsymbol{x}_n$

Gaussian Mixture Model (GMM)

- Assume the data is generated from a mixture of K Gaussians



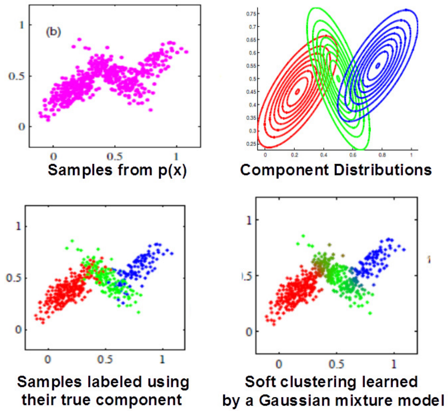
- Each Gaussian represents a "cluster" in the data
- The distribution $p(\boldsymbol{x})$ will be a weighted a mixture of K Gaussians

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

  where π_k's are the **mixing weights**: $\sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0$ (intuitively, $\pi_k = p(\boldsymbol{z} = k)$ is the fraction of data generated by the k-th distribution)
- The goal is to learn the params $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ of these K Gaussians, the mixing weights $\{\pi_k\}_{k=1}^{K}$, and/or the cluster assignment $\boldsymbol{z}_n$ of each $\boldsymbol{x}_n$
- GMM in many ways improves over K-means clustering

GMM Clustering: Pictorially

Some synthetically generated data (top-left) generated from a mixture of 3 overlapping Gaussians (top-right).



Notice the "mixed" colored points in the overlapping regions in the final clustering

Next Class

- GMM in more detail. Extensions of GMM.
- Parameter estimation in GMM
- The Expectation Maximization (EM) algorithm