Advanced Machine Learning
Generative Model

Yu Wang
Assistant Professor
Department of Computer Science
University of Oregon

Data Distribution

Dog

Cat

Data Distribution

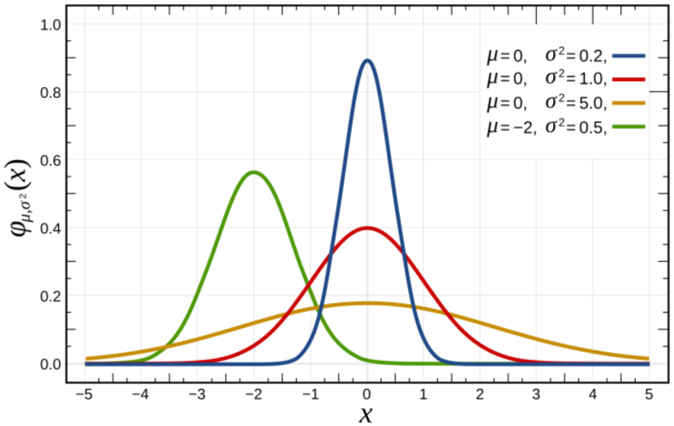
Dog

Cat

1. **There is no concrete image/shape of the dog, everyone can come up with one of your own choice**

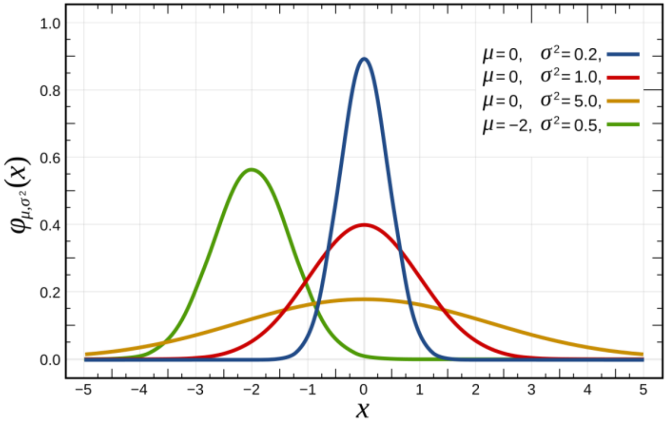
2. **But somehow dog and cat image distributions are different**

When you draw an image, you are actually sampling from a probability distribution!



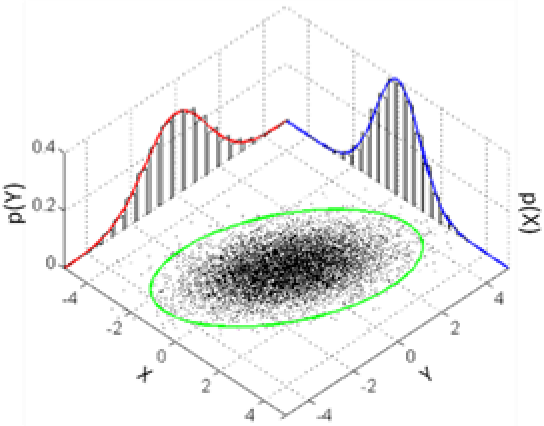
Data Distribution

1D Gaussian Distribution



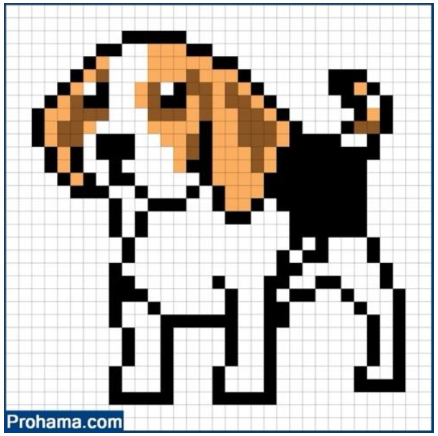
$\mathbb{R}$

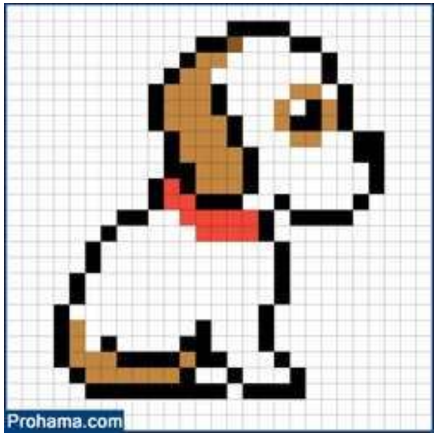
2D Gaussian Distribution



$\mathbb{R}^2$

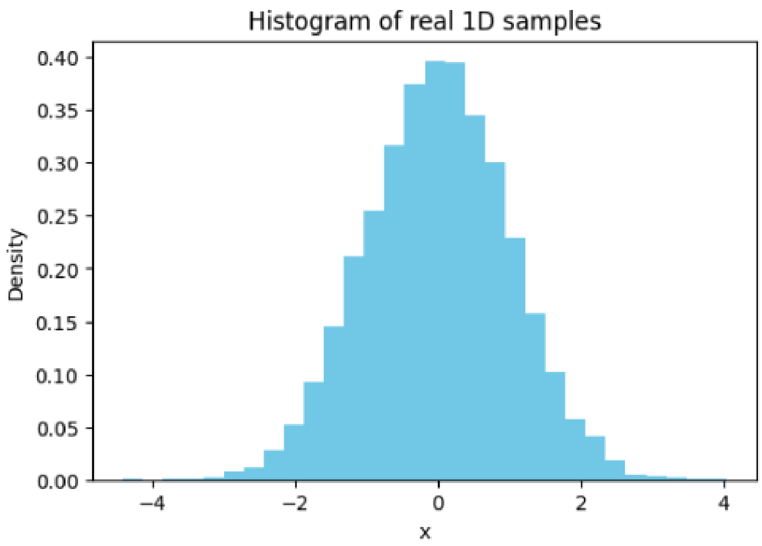
$\mathbb{R}^{256\times256}$

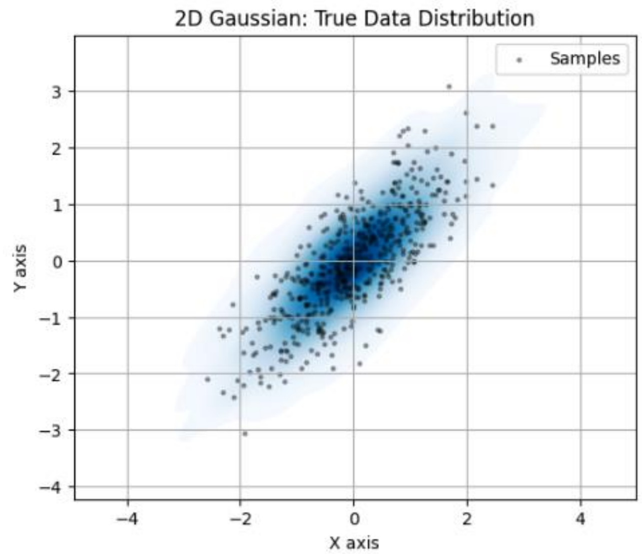




$\mathbb{R}^{256\times256}$

Data Distribution






```python
import numpy as np
import matplotlib.pyplot as plt

# Generate real 1D data samples from a Gaussian distribution N(mean, std^2)
mean1d, std1d = 0.0, 1.0
real_samples_1d = np.random.normal(mean1d, std1d, size=10000)

# Plot histogram of real samples
plt.figure(figsize=(6,4))
plt.hist(real_samples_1d, bins=30, density=True, color='skyblue')
plt.title("Histogram of real 1D samples")
plt.xlabel("x"); plt.ylabel("Density")
plt.show()
```

```python
# Generate real 2D data (Gaussian with correlation)
mean2d = [0.0, 0.0]
cov2d = [[1.0, 0.8],
         [0.8, 1.0]]  # covariance matrix with correlation 0.8
real_samples_2d = np.random.multivariate_normal(mean2d, cov2d, size=5000)
```

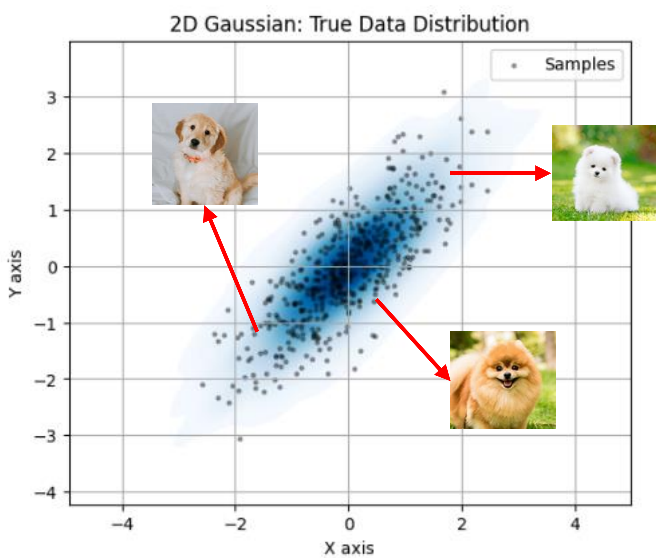
```python
import seaborn as sns

# Unpack real samples into x and y
x_real = real_samples_2d[:, 0]
y_real = real_samples_2d[:, 1]

# Plot: scatter with density contour
plt.figure(figsize=(6, 5))
sns.kdeplot(x=x_real, y=y_real, fill=True, cmap="Blues", thresh=0.01, levels=100)
plt.scatter(x_real[:500], y_real[:500], s=5, color="black", alpha=0.3, label="Samples")
plt.title("2D Gaussian: True Data Distribution")
plt.xlabel("X axis"); plt.ylabel("Y axis")
plt.legend()
plt.grid(True)
plt.axis("equal")
plt.show()
```

Estimating Data Distribution

Estimate Data Distribution



Generative Model

Probability distribution of the **objective** based on the **observed data**

$$\{x_i\}_{i=1}^{N} \xrightarrow[\text{Good Model}]{} P(x) \xrightarrow[\text{Good Data}]{} x$$

- **Machine Learning Methods**
  - Gaussian Kernel Density Estimation
  - Gaussian Mixture Models

Using existing function to estimate what you do not know that can best fit your observation

- **Deep Learning Methods**
  - Auto-Encoder (AE)
  - Variational AE (VAE)
  - Generative Adversarial Network
  - Diffusion Model

Using learnable function to estimate what you do not know that can best fit your observation

Generative Model

Probability distribution of the **objective** based on the **observed data**

$$\{x_i\}_{i=1}^{N} \xrightarrow{\text{Good Model}} P(x) \xrightarrow{\text{Good Data}} x$$

- **Machine Learning Methods**
  - Gaussian Kernel Density Estimation
  - Gaussian Mixture Models

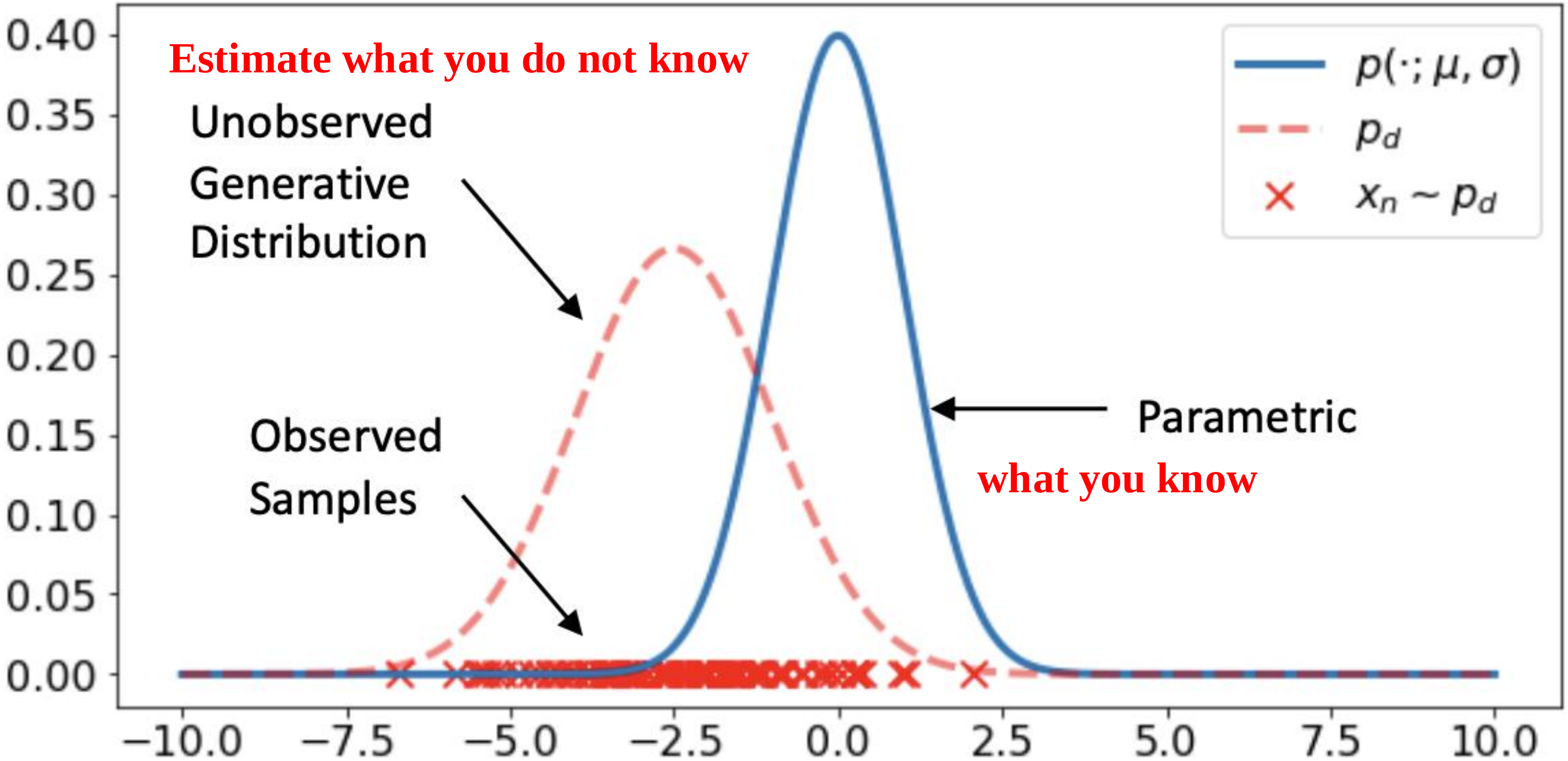
Using existing function to estimate what you do not know that can best fit your observation

- **Deep Learning Methods**
  - Auto-Encoder (AE)
  - Variational AE (VAE)
  - Generative Adversarial Network
  - Diffusion Model

Using learnable function to estimate what you do not know that can best fit your observation

Gaussian Kernel Density Estimation

Using existing function to estimate what you do not know that can best fit your observation

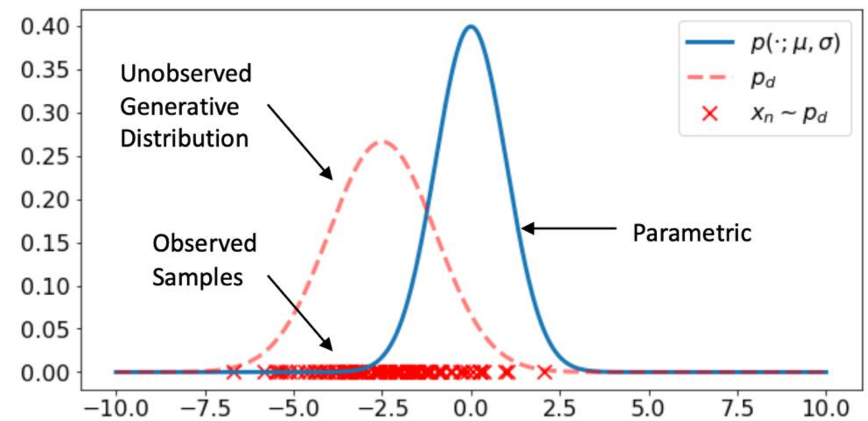


Gaussian Kernel Density Estimation

Using existing function to estimate what you do not know that can best fit your observation

What you know is Gaussian

$$p(x;\mu,\sigma) = N(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



What you observe is a set of i.i.d samples from a Gaussian

$$\{x_i | x_i \sim P_d\}_{i=1}^{N}$$

How can we estimate the gaussian distribution?

What is μ, σ ?

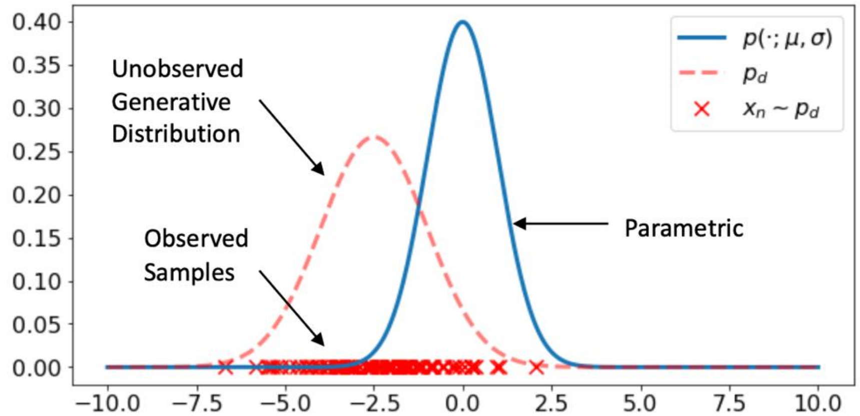
Gaussian Kernel Density Estimation

Using existing function to estimate what you do not know that can best fit your observation

$$\{x_i | x_i \sim P_d\}_{i=1}^{N}$$

$$p(x;\mu,\sigma) = N(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

What is μ, σ ?



1- Maximum Likelihood:

$$\underset{\mu,\sigma}{\operatorname{argmax}}\, p(x_1, \dots, x_N; \mu, \sigma) = \prod_{n=1}^{N} p(x_n; \mu, \sigma)$$

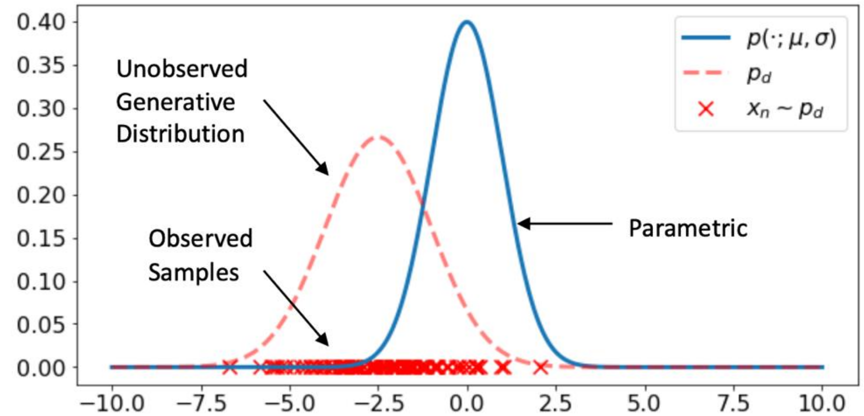
Gaussian Kernel Density Estimation

Using existing function to estimate what you do not know that can best fit your observation

$$\{x_i | x_i \sim P_d\}_{i=1}^{N}$$

$$p(x;\mu,\sigma) = N(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

What is μ, σ ?



2 - Maximum Log-Likelihood:

$$\underset{\mu,\sigma}{\operatorname{argmax}} \log\left(\prod_{n=1}^{N} p(x_n;\mu,\sigma)\right) = \sum_{n=1}^{N} \log(p(x_n;\mu,\sigma))$$

Gaussian Kernel Density Estimation

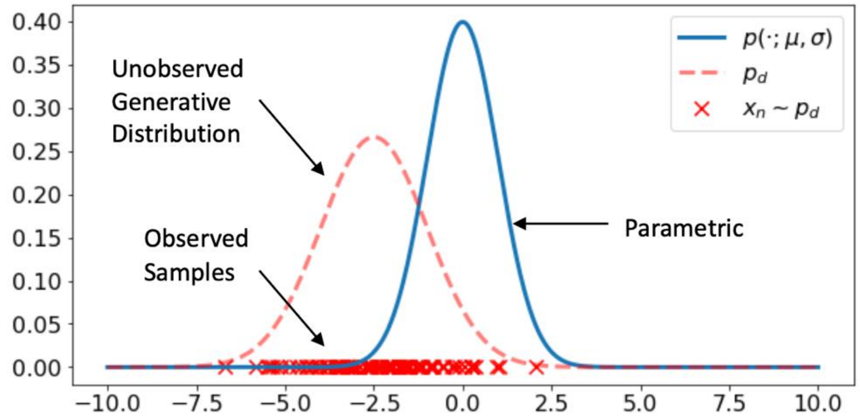
$$\{x_i | x_i \sim P_d\}_{i=1}^{N}$$

$$\downarrow$$

$$p(x;\mu,\sigma) = N(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\downarrow$$

What is μ, σ ?



3- Minimizing Negative Log-Likelihood:

$$\underbrace{\underset{\mu,\sigma}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{\log(2\pi\sigma^2)}{2} + \frac{(x_n-\mu)^2}{2\sigma^2}}_{\text{L}} \longrightarrow \begin{cases} \dfrac{\partial L}{\partial \mu} = 0 \Rightarrow \mu_* = \dfrac{1}{N}\displaystyle\sum_{n=1}^{N} x_n \\ \dfrac{\partial L}{\partial \sigma} = 0 \Rightarrow \sigma_*^2 = \dfrac{1}{N}\displaystyle\sum_{n=1}^{N} (x_n - \mu^*)^2 \end{cases}$$

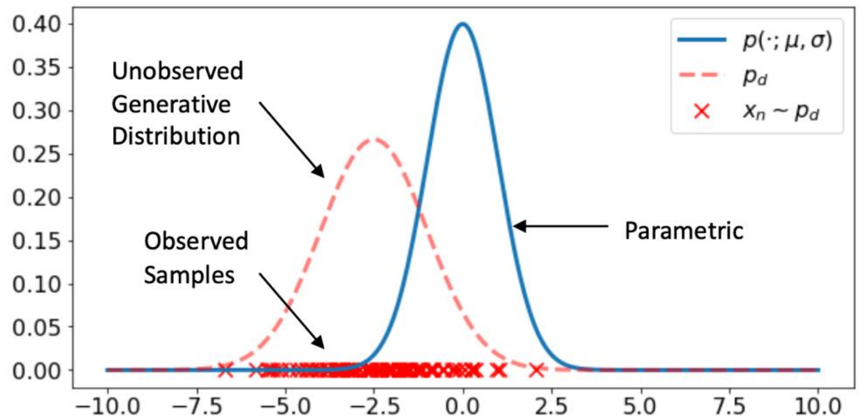
Gaussian Kernel Density Estimation

2 - Maximum Log-Likelihood:

$$\underset{\mu,\sigma}{\operatorname{argmax}} \log\left(\prod_{n=1}^{N} p(x_n;\mu,\sigma)\right) = \sum_{n=1}^{N} \log(p(x_n;\mu,\sigma))$$

Monte-Carlo
Approximation

$$\int_{\mathbb{R}} p_d(x)\log(p(x;\mu,\sigma))\,dx \approx \frac{1}{N}\sum_{n=1}^{N}\log(p(x_n;\mu,\sigma))$$



$$\underset{\mu,\sigma}{\operatorname{argmax}} \int_{\mathbb{R}} p_d(x)\log(p(x;\mu,\sigma))\,dx = \underset{\mu,\sigma}{\operatorname{argmax}} \int_{\mathbb{R}} p_d(x)\log(p(x;\mu,\sigma))\,dx - \int_{\mathbb{R}} p_d(x)\log(p_d(x))\,dx$$

$$= \underset{\mu,\sigma}{\operatorname{argmax}} \int_{\mathbb{R}} p_d(x)\log\left(\frac{p(x;\mu,\sigma)}{p_d(x)}\right)dx = \underset{\mu,\sigma}{argmin} \int_{\mathbb{R}} p_d(x)\log\left(\frac{p_d(x)}{p(x;\mu,\sigma)}\right)dx$$

$$= \underset{\mu,\sigma}{\operatorname{argmin}}\, D_{KL}(p_d || p(\cdot;\mu,\sigma))$$

Maximize Log-Likelihood of parametric distribution = Minimize KL Divergence between the data distribution and the parametric distribution

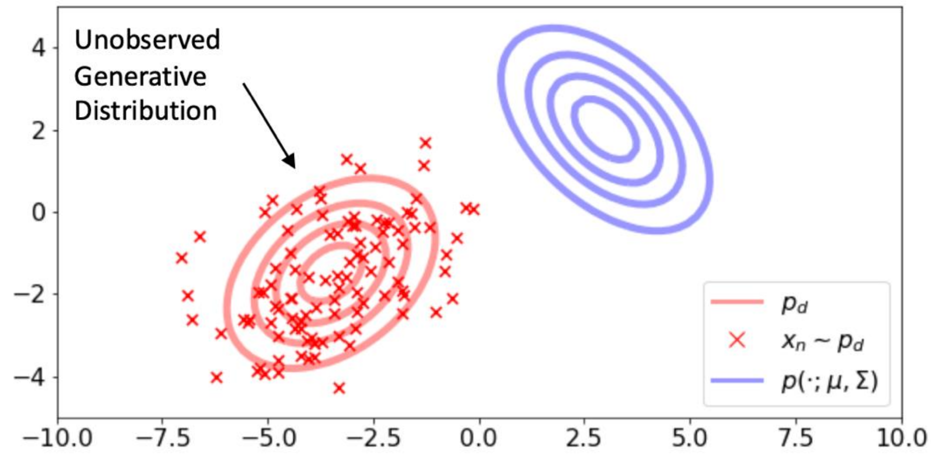
Gaussian Kernel Density Estimation - Example

DEMO

Gaussian Kernel Density Estimation

How about 2D Gaussian Dimension?

$$p(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} e^{-\frac{(x-\mu)^T \Sigma^{-1}(x-\mu)}{2}}$$

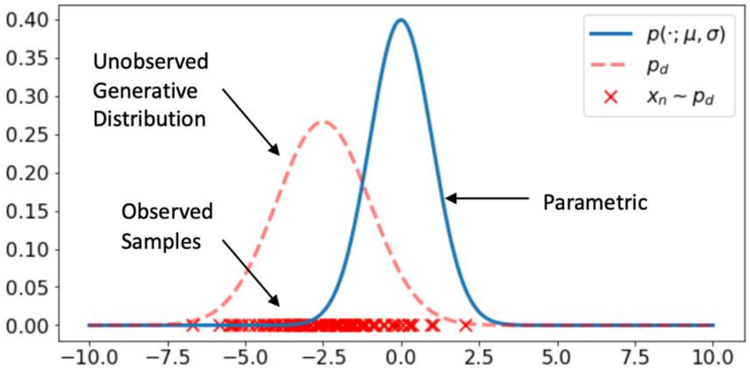


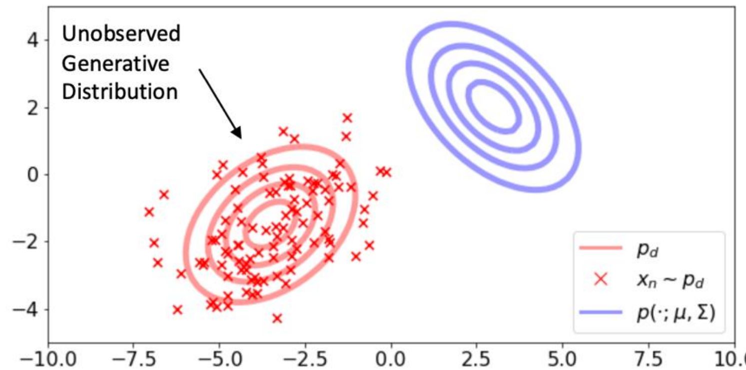
$$\frac{\partial L}{\partial \mu} = 0 \Rightarrow \mu_* = \frac{1}{N}\sum_{n=1}^{N} x_n$$

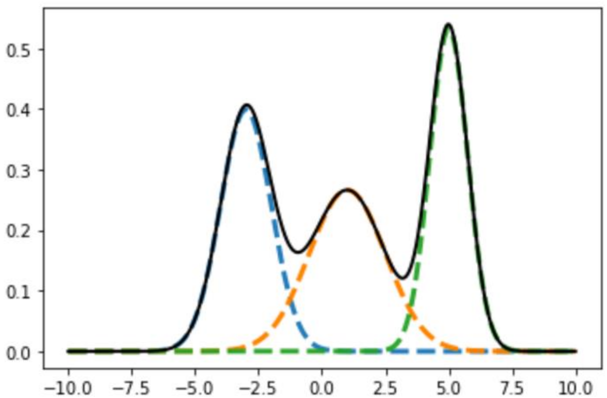
$$\frac{\partial L}{\partial \Sigma} = 0 \Rightarrow \Sigma_* = \frac{1}{N}\sum_{n=1}^{N} (x_n - \mu^*)(x_n - \mu^*)^T$$

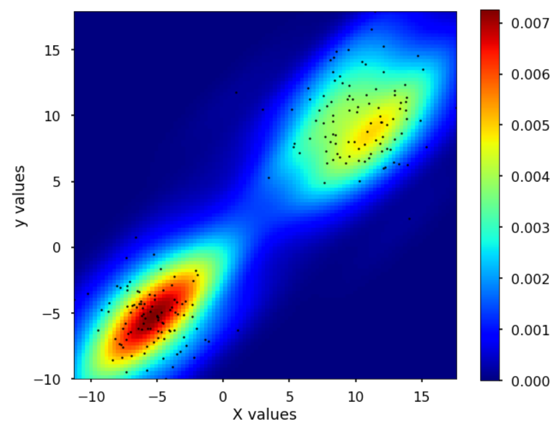
Any problem with such estimation?

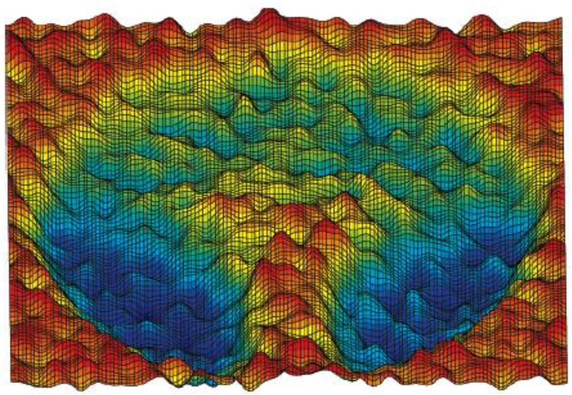
Gaussian Mixture Models





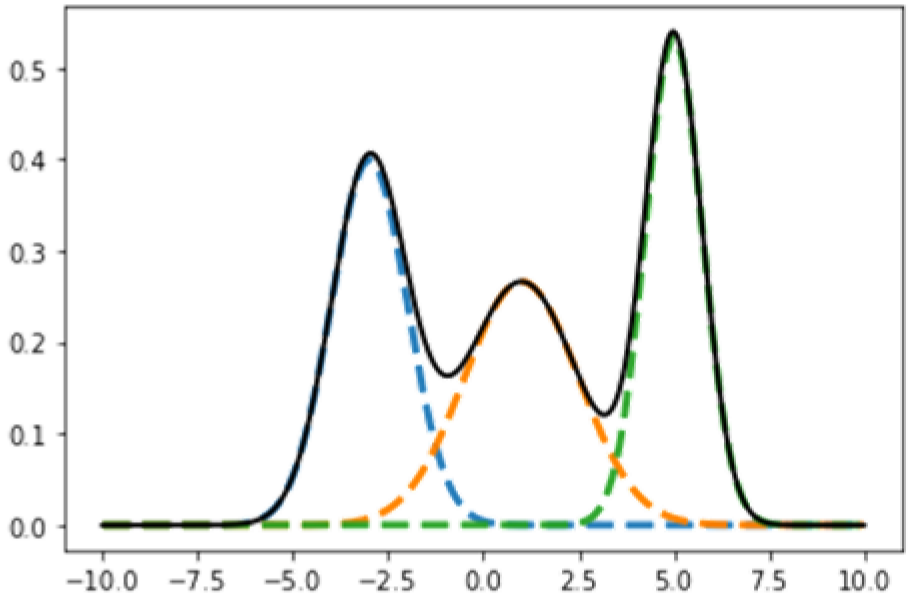






If using one existing function to estimate is not enough, then let’s try more!

Gaussian Mixture Models



Assume we have N i.i.d samples from a mixture of Gaussians distribution, $\{x_n \sim p_d\}_{n=1}^{N}$.

- **How do we estimate the parameters of the mixture from the observed samples?**

$$p\left(x; [(\alpha_k, \mu_k, \sigma_k)]_{k=1}^{K}\right) = \sum_{k=1}^{K} \alpha_k N(x; \mu_k, \sigma_k) = \sum_{k=1}^{K} \frac{\alpha_k}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

Where $\alpha_k \geq 0$, $\sum_{k=1}^{K} \alpha_k = 1$, and $\sigma_k \geq 0$.

Gaussian Mixture Models

$$p(x; [(\alpha_k, \mu_k, \sigma_k)]_{k=1}^{K}) = \sum_{k=1}^{K} \alpha_k N(x; \mu_k, \sigma_k) = \sum_{k=1}^{K} \frac{\alpha_k}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

Where $\alpha_k \geq 0$, $\sum_{k=1}^{K} \alpha_k = 1$, and $\sigma_k \geq 0$.

$$\underset{\mu,\sigma}{\operatorname{argmax}} \log\left(\prod_{n=1}^{N} p(x_n; \mu, \sigma)\right) = \sum_{n=1}^{N} \log(p(x_n; \mu, \sigma))$$

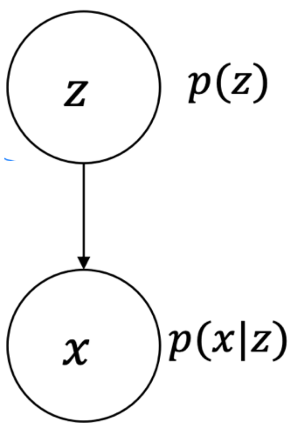
$$\sum_{n=1}^{N} Log(\sum_{k=1}^{K} p(x_n; u, \sigma)) = \sum_{n=1}^{N} Log(\sum_{k=1}^{K} \frac{\alpha_k}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x_n-\mu_k)^2}{2\sigma_k^2}})$$

Hard to optimize

Gaussian Mixture Models

If, for each sample, we knew which Gaussian it is sampled from the problem would have been solved! Meaning that we could have solved K maximum log-likelihoods to estimate parameters of each Gaussian independent of the others!

$z = [z_1, \dots, z_K], z_k \in \{0,1\}$



$$p(x|z_k = 1) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

$$p(x) = \sum_{k=1}^{K} p(x|z_k = 1)p(z_k = 1) = \sum_{k=1}^{K} \frac{\alpha_k}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

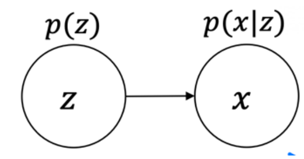
Gaussian Mixture Models

$$p(x) = \sum_z p(z,x)$$

$$p(x) = \int p(z,x)\, dz$$

$$\frac{d}{dx}\log(f(x)) = \frac{1}{f(x)} \cdot f'(x)$$

$$\frac{d}{d\theta}\log p(x) = \frac{d}{d\theta}\log\left(\sum_z p(z,x)\right) = \frac{\frac{d}{d\theta}\sum_z p(z,x)}{\sum_{z'} p(z',x)} = \frac{\sum_z \frac{d}{d\theta} p(z,x)}{\sum_{z'} p(z',x)}$$



$$\frac{\sum_z p(z,x)\frac{\frac{d}{d\theta}p(z,x)}{p(z,x)}}{\sum_{z'} p(z',x)} = \sum_z \left(\frac{p(z,x)}{\sum_{z'} p(z',x)}\right)\frac{d}{d\theta}\log(p(z,x)) = \sum_z p(z|x)\frac{d}{d\theta}\log(p(z,x))$$

$$\sum_z p(z|x)\frac{d}{d\theta}\log(p(x|z)p(z)) = \sum_z p(z|x)\frac{d}{d\theta}\log(p(x|z)) + \sum_z p(z|x)\frac{d}{d\theta}\log(p(z))$$

$p(z|x)$ **Soft assignment of data x to each Gaussian**

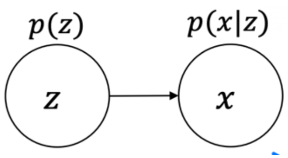
$\log(p(z))$ **Factuality of the latent distribution z, does not depend on θ**

Gaussian Mixture Models

$$p(x) = \sum_z p(z, x) \qquad \frac{d}{dx}\log(f(x)) = \frac{1}{f(x)} \cdot f'(x)$$

$$p(x) = \int_z p(z, x)\, dz$$

$$\frac{d}{d\theta}\log p(x) = \frac{d}{d\theta}\log\left(\sum_z p(z,x)\right) = \frac{\frac{d}{d\theta}\sum_z p(z,x)}{\sum_{z'} p(z',x)} = \frac{\sum_z \frac{d}{d\theta} p(z,x)}{\sum_{z'} p(z',x)}$$



$$\frac{\sum_z p(z,x) \frac{\frac{d}{d\theta}p(z,x)}{p(z,x)}}{\sum_{z'} p(z',x)} = \sum_z \left(\frac{p(z,x)}{\sum_{z'} p(z',x)}\right) \frac{d}{d\theta}\log(p(z,x)) = \sum_z p(z|x) \frac{d}{d\theta}\log(p(z,x))$$

$$\sum_z p(z|x) \frac{d}{d\theta}\log(p(x|z)p(z)) = \boxed{\sum_z p(z|x) \frac{d}{d\theta}\log(p(x|z))} + \sum_z p(z|x) \frac{d}{d\theta}\log(p(z))$$

$p(z|x)$ **Soft assignment of data x to each Gaussian**

$\log(p(z))$ **Factuality of the latent distribution z, does not depend on θ**

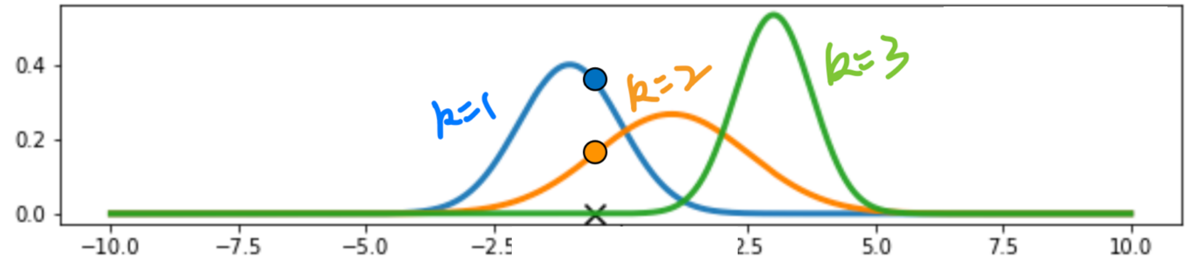
Gaussian Mixture Models – EM algorithms

$$\sum_{z} p(z|x) \frac{d}{d\theta} \log(p(x|z))$$

Linear Combine N independent Gaussian

- **E**xpectation Step: for fixed parameters $[(\alpha_k, \mu_k, \sigma_k)]_{k=1}^{K}$ compute $r_n^k = p(z_k = 1|x_n)$ for each sample

$$r_n^k = \frac{\alpha_k N(x_n; \mu_k, \sigma_k)}{\sum_i \alpha_i N(x_n; \mu_i, \sigma_i)}$$



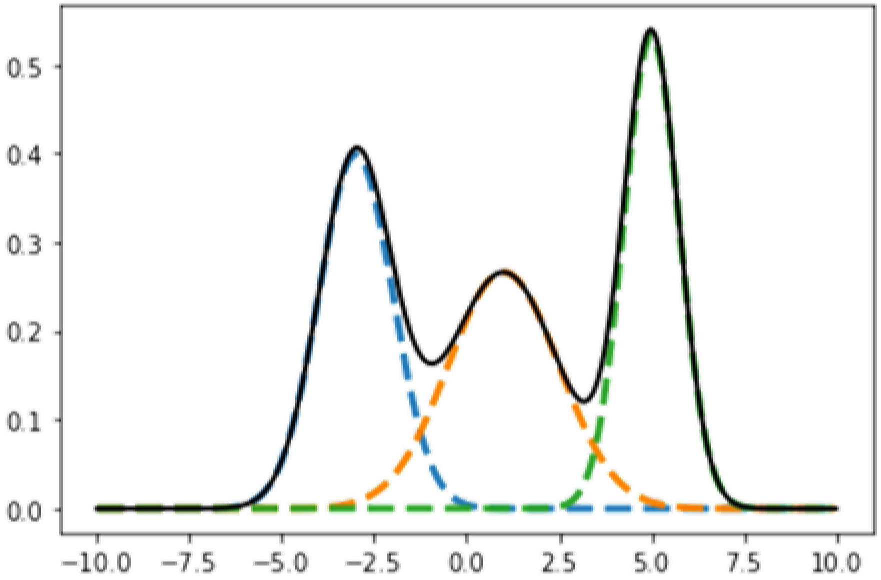
- **M**aximization Step: for fixed r_n^k solve the maximum log-likelihood to obtain optimal parameters:
  - Means: $\mu_k = \frac{1}{N_k} \sum_n r_n^k x_n$
  - Variances: $\sigma_k^2 = \frac{1}{N_k} \sum_n r_n^k (x_n - \mu_k)^2$
  - **Covariances: $\boldsymbol{\Sigma_k = \frac{1}{N_k} \sum_n r_n^k (x_n - \mu_k)(x_n - \mu_k)^T}$**

Gaussian Mixture Models – EM algorithm Example

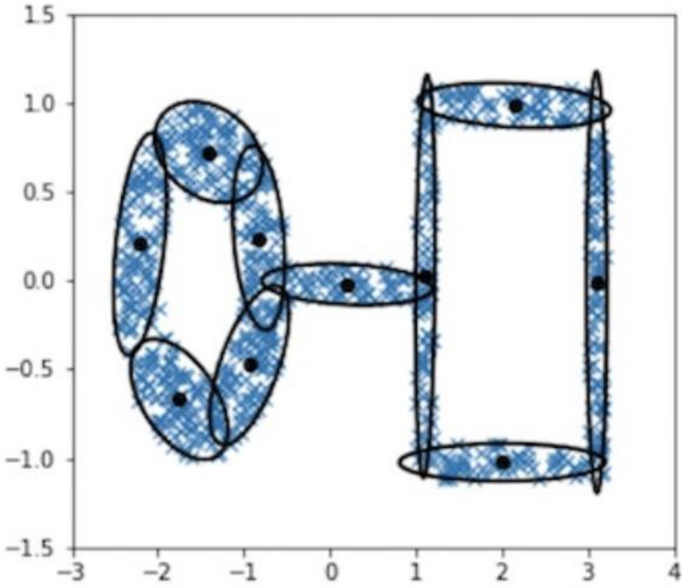
DEMO

Problems

Using existing function to estimate what you do not know that can best fit your observation



Three bumps but I just give you two different gaussian

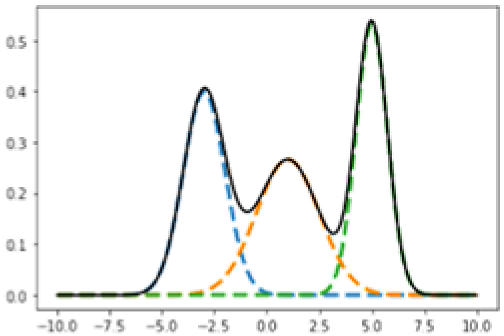


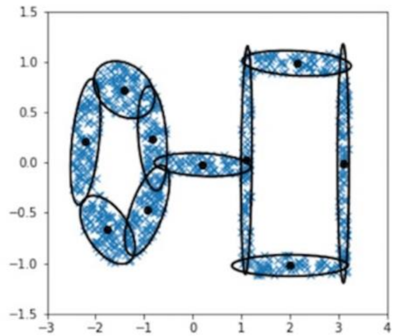
I just give you two different gaussian.

All you know for modeling what you do not know is fixed. But how do you know those fixed things is able to model the unknown thing?

Problems

Using existing function to estimate what you do not know that can best fit your observation





$\mathbb{R}^1, \mathbb{R}^2$

$\mathbb{R}^{256\times256}$

What you have is some low-dimensional data
But what you want to model is some high-dimensional data, how it could be?