ECE 57000

Unsupervised Learning and Density Estimation

Chaoyue Liu

Fall 2024

Unsupervised Learning

Dataset: each sample contains only “input” **x**, but has no label.

topics include:

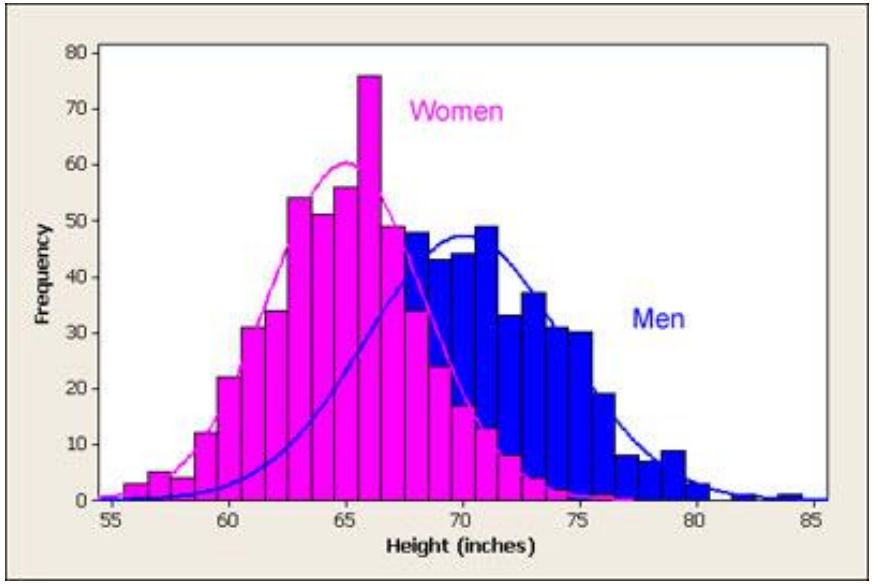
- density estimation
- clustering
- dimensionality reduction
- generative models
- …

Density estimation

Density estimation finds a density (PDF/PMF) that represents the data (or empirical distribution) well

- Assume there is a ground-truth (unknown) distribution $P(\mathbf{x})$, and the dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{n}$ is obtained by randomly drawn samples i.i.d. from $P(\mathbf{x})$

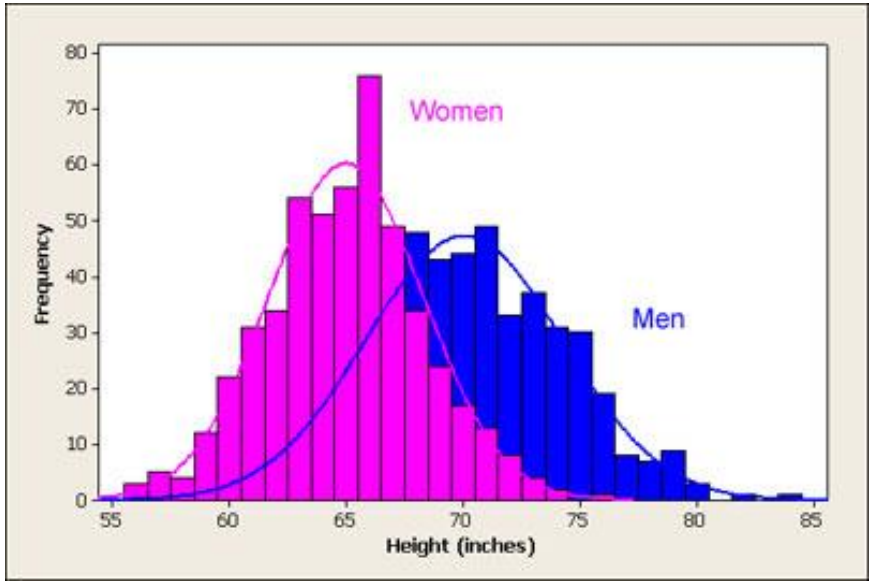
- **Goal**: based on $\mathcal{D}$, find a density/distribution function $\hat{P}(\mathbf{x})$ so that $\hat{P}(\mathbf{x})$ is as close to $P(\mathbf{x})$ as possible



Histogram

Histogram is the most basic density estimation method

- Setup bin size v_i (typically $v_i = v$) and locations
- Count number of samples that fall in each bin a_i
- Assign $p_i(\mathbf{x}) = \frac{a_i}{v_i}$ to all $\mathbf{x}$ within i-th bin
- Normalize the function to be a density (i.e., integration is 1)

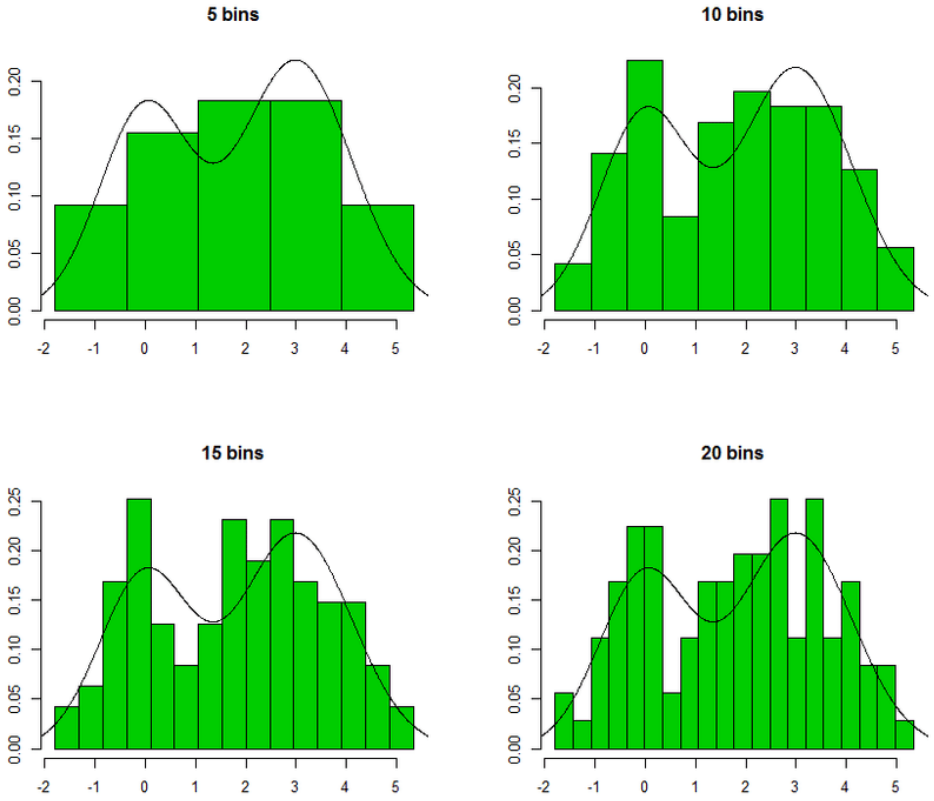


$\hat{P}(\mathbf{x})$: piecewise constant functions

Histogram

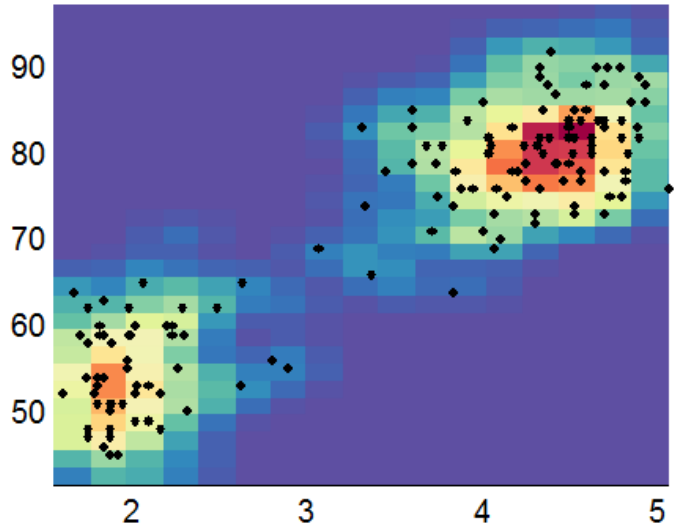
How to select the number of bins?

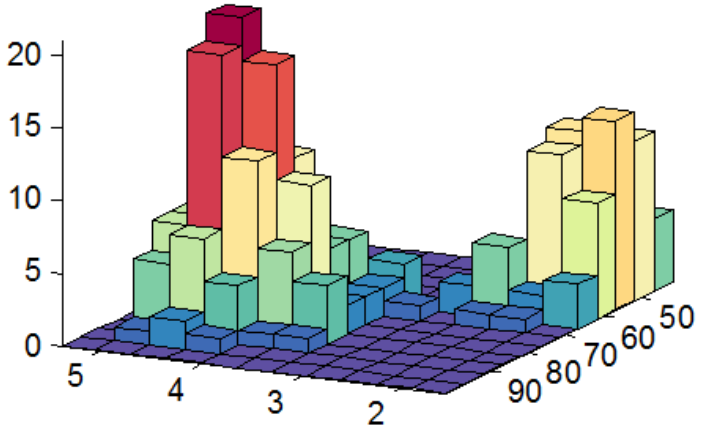
- Too few bins will underfit
- Too many bins will overfit



Histogram

2D Histograms can be created





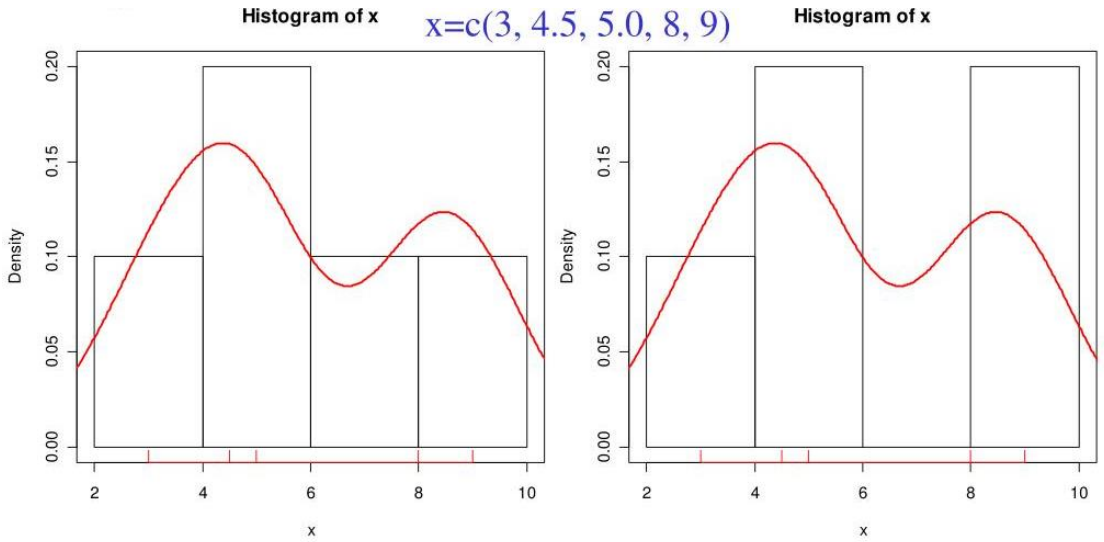
Curse of dimensionality:

In high dimensional case ($d \gg 1$), number of bins grows exponentially: $\left(\frac{range}{bin_{size}}\right)^{d}$

Histogram

Drawback:

- Estimation function $\hat{P}(\mathbf{x})$ is not smooth (on the bin edges)



- hist(x,right=**T**,freq=F), R-default
- (a,b] right closed (left-open)
- hist(x,right=**F**,freq=F)
- [a,b) left closed (right-open)

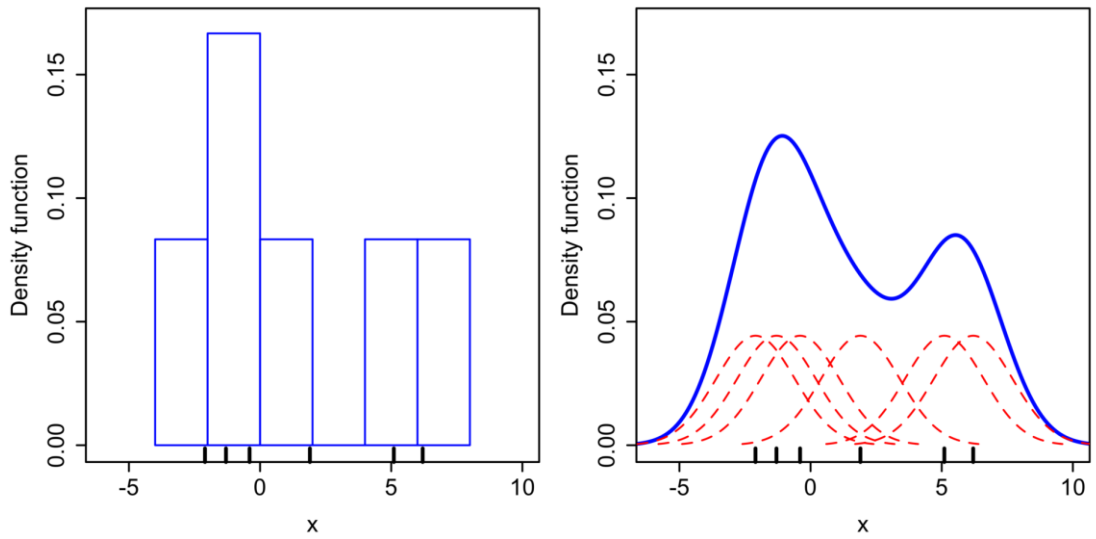
Area=1

Kernel density

Kernel densities overcome this drawback by placing a Gaussian density at each point

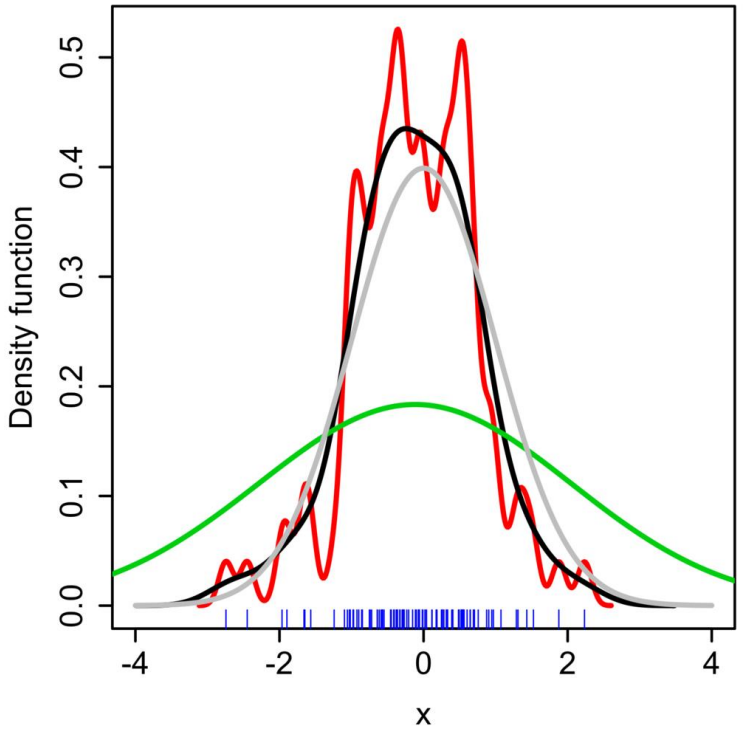
- Kernel density has the following form:

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} p_{\text{base}}(x - x_i) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{N}(x - x_i, \sigma)$$



Kernel density

Similar to number of bins, the key parameter for kernel densities is the “bandwidth” or σ parameter



k-nearest neighbor (k-NN)

(**Note**: different from the k-NN model that we discussed in supervised learning)

Goal: compute $\hat{P}(\mathbf{x})$ for all $\mathbf{x}$.

Steps:

- For a given $\mathbf{x}$, among all the training samples $\{\mathbf{x}_i\}_{i=1}^n$, find the k-th nearest neighbor to $\mathbf{x}$
- let $r_k(\mathbf{x})$ be the distance from x to its k-th nearest neighbor
- Let $v_k(\mathbf{x})$ be the volume of the ball with radius $r_k(\mathbf{x})$: $v_k(\mathbf{x}) = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} \cdot \left(r_k(\mathbf{x})\right)^d$
- $\hat{P}(\mathbf{x}) = C \cdot \frac{k}{v_k(\mathbf{x})}$, where C is a normalization factor

Histogram, Kernel density, *k*-NN are non-parametric methods

Parametric density estimation assumes a **density model class** parameterized by θ

- Assumption: Bernoulli density

$$\theta = [p], \qquad p \in [0,1]$$

  - Example: toss a (biased) coin
- Assumption: Gaussian density

$$\theta = [\mu, \sigma^2], \qquad \mu \in \mathbb{R}, \sigma^2 \in \mathbb{R}_+$$

- Assumption: Gaussian mixture model

$$\theta = \left\{\pi_i, \mu_i, \sigma_i^2\right\}_{i=1}^{K}, \qquad \pi_i \in (0,1), \mu_i \in \mathbb{R}, \sigma_i^2 \in \mathbb{R}_+$$

Q: How to determine which model (i.e., parameter setting) in the model class is the best?

- Need to find a "distance" to measure the difference between two density functions
- Minimize the "distance"

Kullback-Leibler Divergence (KL)

- **KL divergence** for discrete variables

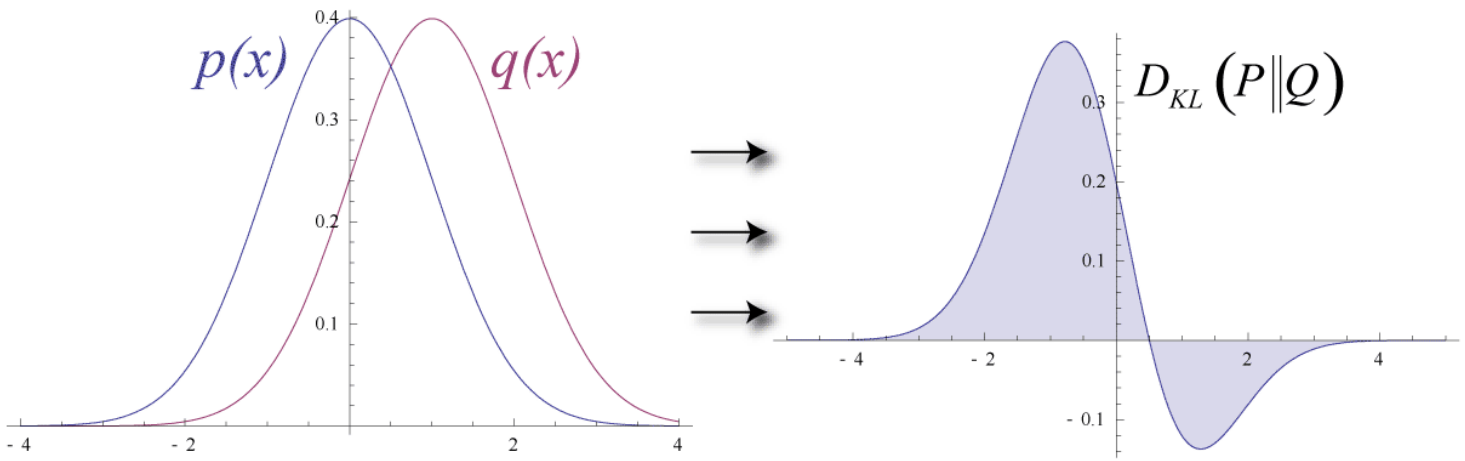
$$KL(P(\cdot), Q(\cdot)) = \mathbb{E}_{x\sim P}\left[\log\frac{P(x)}{Q(x)}\right] = \sum_{x} P(x)\log\frac{P(x)}{Q(x)}$$

- **KL divergence** for continuous variables

$$KL(p(\cdot), q(\cdot)) = \mathbb{E}_{X\sim p}\left[\log\frac{p(x)}{q(x)}\right] = \int_{x} p(x)\log\frac{p(x)}{q(x)}\,dx$$

- **KL divergence** for continuous variables

$$KL(p(\cdot), q(\cdot)) = \mathbb{E}_{X\sim p}\left[\log \frac{p(x)}{q(x)}\right] = \int_x p(x) \log \frac{p(x)}{q(x)} dx$$



Original Gaussian PDF's

KL Area to be Integrated

- KL divergence is not a distance!

$$KL(p(\cdot), q(\cdot)) = \mathbb{E}_{x\sim p}\left[\log\frac{p(x)}{q(x)}\right] = \int_x p(x)\log\frac{p(x)}{q(x)}\,dx$$

Not symmetric!

$$KL\big(p(\cdot), q(\cdot)\big) \neq KL\big(q(\cdot), p(\cdot)\big)$$

Non-negative property

$$KL\big(p(\cdot), q(\cdot)\big) \geq 0$$

Equal distribution property:

$$KL\big(p(\cdot), q(\cdot)\big) = 0 \Leftrightarrow p(\cdot) = q(\cdot)$$

One use of KL divergence is to estimate distribution parameters only from samples

- $P(\mathbf{x})$: the **real/true** distribution of the data
  - $P(\mathbf{x})$ is ***unknown***
  - We only have samples $\{\mathbf{x}_i\}_{i=1}^{n}$ from $P(\mathbf{x})$
- $\hat{P}(\mathbf{x};\theta)$: an **estimate** of the true distribution
  - Parametrized by θ
- We want to find $\hat{P}(\mathbf{x};\theta)$that is closest to $P(\mathbf{x})$

$$\theta^* = \arg\min_{\theta} KL(\, P(\cdot), \hat{P}(\cdot;\theta))$$

Wait, but we don't know $P(\mathbf{x})$, how do we do this?

Two main ideas for simplification

- Constants with respect to (w.r.t.) θ can be ignored
- Full expectation replaced by empirical expectation

$$
\begin{aligned}
&\arg\min_{\theta} KL(\,P(\cdot), \hat{P}(\cdot;\theta)) \\
&= \arg\min_{\theta} \mathbb{E}_{\mathbf{x}\sim P}\left[\log \frac{P(\mathbf{x})}{\hat{P}(\mathbf{x};\theta)}\right] \\
&= \arg\min_{\theta} -\mathbb{E}_{\mathbf{x}\sim P}\left[\log \hat{P}(\mathbf{x};\theta)\right] + \mathbb{E}_{\mathbf{x}\sim P}[\log P(\mathbf{x})] \\
&= \arg\min_{\theta} -\mathbb{E}_{\mathbf{x}\sim P}\left[\log \hat{P}(\mathbf{x};\theta)\right] + C \\
&\approx \arg\min_{\theta} -\widehat{\mathbb{E}}_{\mathbf{x}\sim P}\left[\log \hat{P}(\mathbf{x};\theta)\right] \\
&= \arg\min_{\theta} -\frac{1}{n}\sum_{i=1}^{n} \log \hat{P}(\mathbf{x}_i;\theta)
\end{aligned}
$$

Maximum likelihood estimation (MLE) is another way to estimate distribution parameters from samples

- **Likelihood function** how likely (or probable) a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$ is under a distribution with parameters θ

$$\mathcal{L}(\theta; \mathcal{D}) = \hat{P}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n; \theta)$$

- If we ***assume*** samples (or observations) of dataset are i.i.d., then

$$\mathcal{L}(\theta; \mathcal{D}) = \prod_{i=1}^{n} \hat{P}(\mathbf{x}_i; \theta)$$

- Often simplified to the **log-likelihood function**

$$\ell(\theta; \mathcal{D}) = \log \mathcal{L}(\theta; \mathcal{D}) = \sum_{i=1}^{n} \log \hat{P}(\mathbf{x}_i; \theta)$$

Maximum likelihood estimation (MLE) is another way to estimate distribution parameters from samples

- Optimize the following

$$\theta^* = \arg\max_\theta \ell(\theta; \mathcal{D}) = \arg\max_\theta \sum_{i=1}^{n} \log \hat{P}(\mathbf{x}_i; \theta)$$

- Equivalent to

$$\theta^* = \arg\min_\theta -\frac{1}{n} \sum_{i=1}^{n} \log \hat{P}(\mathbf{x}_i; \theta)$$

- Wait, doesn't that look familiar?
- **MLE equivalent to minimum KL divergence!**

Gaussian Density

- Univariate: (μ is mean and $\sigma^2$is variance)

$$\hat{P}(x) = \frac{1}{\sqrt{2\pi}\sqrt{\sigma^2}} \exp\left\{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right\}$$

- Multivariate (μ is mean and Σ is covariance)

$$\hat{P}(\mathbf{x}) = \frac{1}{\left(\sqrt{2\pi}\right)^d \sqrt{\det \Sigma}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\right\}$$

- $\Theta = \Sigma^{-1}$ is called the **precision matrix** (or **inverse covariance**)
- Σ (and Θ) must be positive definite $\Sigma > 0$

Gaussian Density

- Univariate: (μ is mean and $\sigma^2$is variance)

$$\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n} \log \hat{P}(x_i;\theta)$$

$$= \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\frac{(x_i-\mu)^2}{\sigma^2} + \log\sigma + const$$

$$\hat{P}(x) = \frac{1}{\sqrt{2\pi}\sqrt{\sigma^2}} \exp\left\{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right\}$$

$$0 = \frac{\partial \mathcal{L}}{\partial \mu} = \frac{1}{n}\sum_{i=1}^{n} \frac{\mu - x_i}{\sigma^2} \quad \Rightarrow \quad \mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$0 = \frac{\partial \mathcal{L}}{\partial \sigma} = \frac{1}{n}\sum_{i=1}^{n} -(x_i-\mu)^2 \cdot \frac{1}{\sigma^3} + \frac{1}{\sigma} \Rightarrow \quad \sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i-\mu)^2$$

Gaussian Density

- Similarly for **multivariate** Gaussian:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$$

$$\hat{\Sigma}_{\text{MLE}} = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T$$

Gaussian mixture model

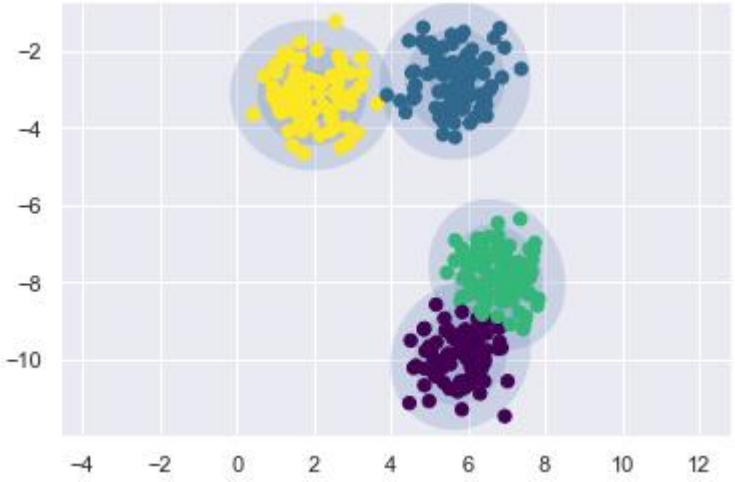
Data may have multiple modes

- Can be approximated by a superposition of several Gaussian distributions

$$\hat{P}(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)$$

$$0 \leq \pi_k \leq 1, \forall k = 1,2,\cdots,K$$

$$\sum_{k=1}^{K} \pi_k = 1$$



$$\mathcal{L}(\theta; \mathcal{D}) = -\frac{1}{n}\sum_{i=1}^{n} \log \hat{P}(\mathbf{x}_i; \theta) = -\frac{1}{n}\sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)$$

No closed-form solution

Gaussian mixture model

How do we sample from a GMM?

1. Select from the mode (i.e., which Gaussian to sample from)
   - Latent variable $z \in \{1,2,\cdots,K\}$, probability $\mathrm{P}(z = k) = \pi_k$
2. Sample from the selected mode
   - Data $\mathbf{x} \sim \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)$

- Observed data $\{\mathbf{x}_i\}_{i=1}^n$ is incomplete
- $\{\mathbf{x}_i, z_i\}_{i=1}^n$ is the complete data set

If we know the latent variable z_i for all i (i.e., we have the complete data), then the optimization problem can be solved

EM algorithm

Iterate over the following two steps:

- **Expectation step**:
  - Compute responsibility that point $\mathbf{x}_i$ belongs to mode k (temporarily fixing π_k, μ_k, Σ_k),

$$r_{ik} = \frac{\pi_k \cdot \mathcal{N}(\mathbf{x}_i \mid \mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_i \mid \mu_k, \Sigma_k)}$$

- **Maximization step**:
  - Find π_k, μ_k, Σ_k that maximize the log likelihood (now fixing r_{ik})

$$\pi_k = \frac{1}{N}\sum_{i=1}^{N} r_{ik} = \frac{N_k}{N}, \text{with } N_k := \sum_{i=1}^{N} r_{ik}$$

$$\mu_k = \frac{1}{N_k}\sum_{i=1}^{N} r_{ik}\mathbf{x}_i$$

$$\Sigma_k = \frac{1}{N_k}\sum_{i=1}^{N} r_{ik}(\mathbf{x}_i - \mu_k)\,(\mathbf{x}_i - \mu_k)^T$$