CZ4041/CE4041: Machine Learning

Lesson 11: Density Estimation

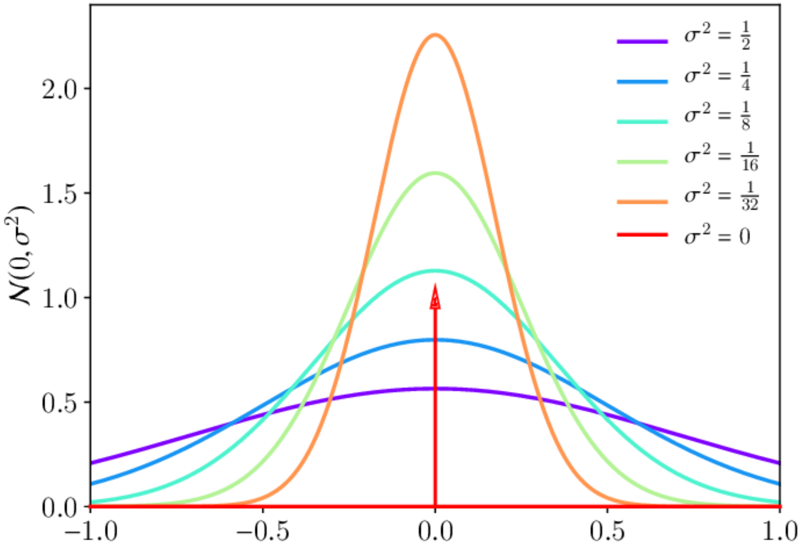
Li Boyang, Albert

School of Computer Science and Engineering, NTU, Singapore

Acknowledgements: some figures are adapted from the lecture notes of Jason J. Corso (SUNY @ Buffalo). Slides are modified from the version prepared by Dr. Sinno Pan.

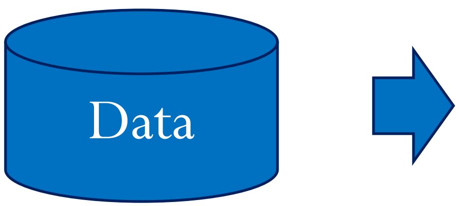
Purpose of Machine Learning

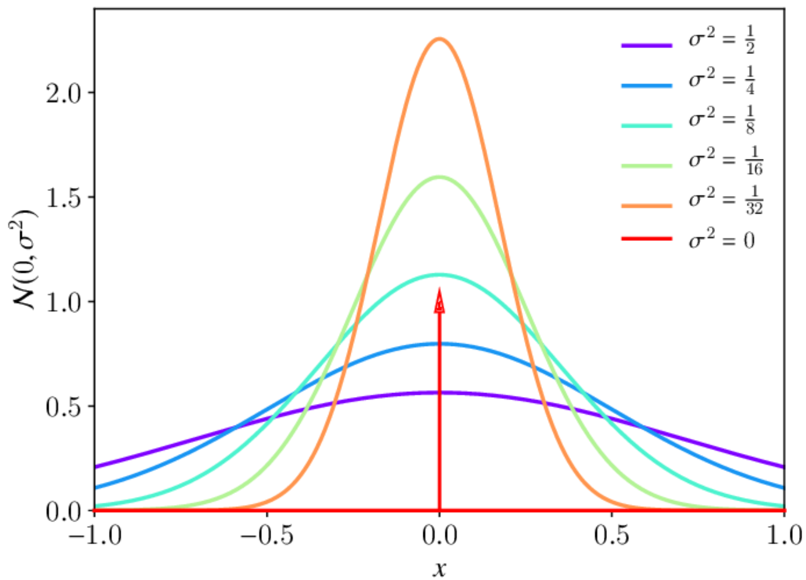
- Model Uncertainty
  - Germany will probably beat Japan, but what are the odds? 60/40, 70/30, or 80/20?
  - I’m willing to bet more money if the odds are in my favor.
  - Often translates to: what is the shape of the probability distribution?



Purpose of Machine Learning

- Model Uncertainty
  - Density Estimation (Week 11)





Why Estimating Distributions

- Recall Naïve Bayes Classifiers

$$y_i = \arg\max_c P(\boldsymbol{x}|y=c)P(y=c)$$

$$= \arg\max_c \underbrace{\prod_{i=1}^{d} P(x_i|y=c)}_{\substack{\text{A naïve, simplifying assumption,} \\ \text{which we may remove if we can} \\ \text{estimate the distribution properly.}}} P(y=c)$$

Discrete vs Continuous Probability Distributions

Discrete Probability Distribution

- Usually a finite number of outcomes
- A 6-sided die ⇒ 6 possible outcomes
- The distribution can be described with six numbers
  - Non-negative
  - Sum up to 1

Outcome	**1**	**2**	**3**	**4**	**5**	**6**
Probability	0.1	0.2	0.1	0.3	0.05	0.25

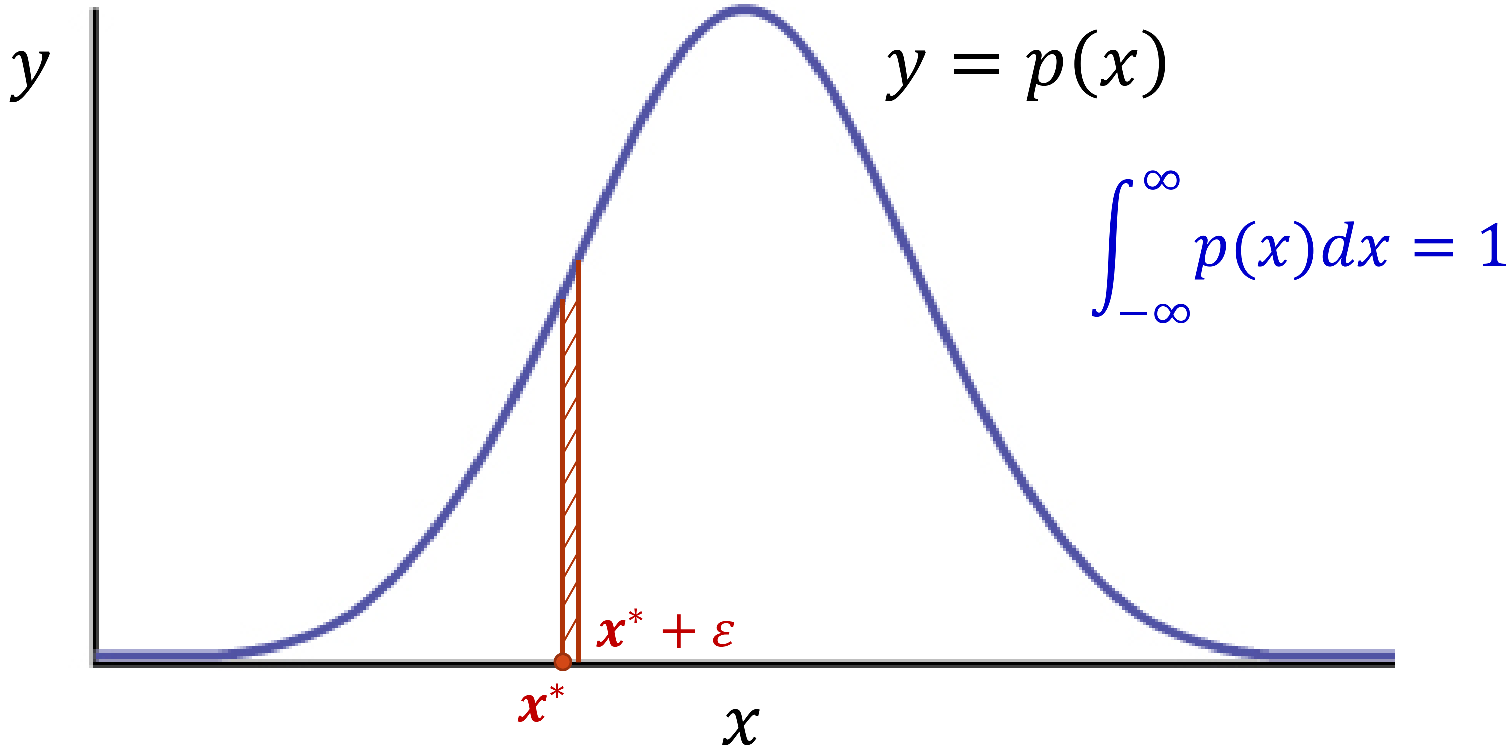
Discrete vs Continuous Probability Distributions

Continuous Probability Distribution

- The outcome is a real number in a range
  - For example, $[0,1]$, $(-\infty, \infty)$
- An infinite number of outcomes between any two real numbers.
- This is really tricky.
- There is a formal branch of mathematics (measure theory) that deal with this, though we will not introduce it here.

Probability Density Function

Key message: Probability is area under the curve



$$P(\boldsymbol{x}^*) = \int_{\boldsymbol{x}^*}^{\boldsymbol{x}^*+\varepsilon} p(x)dx \approx p(\boldsymbol{x}^*) \times \varepsilon$$

Density Estimation

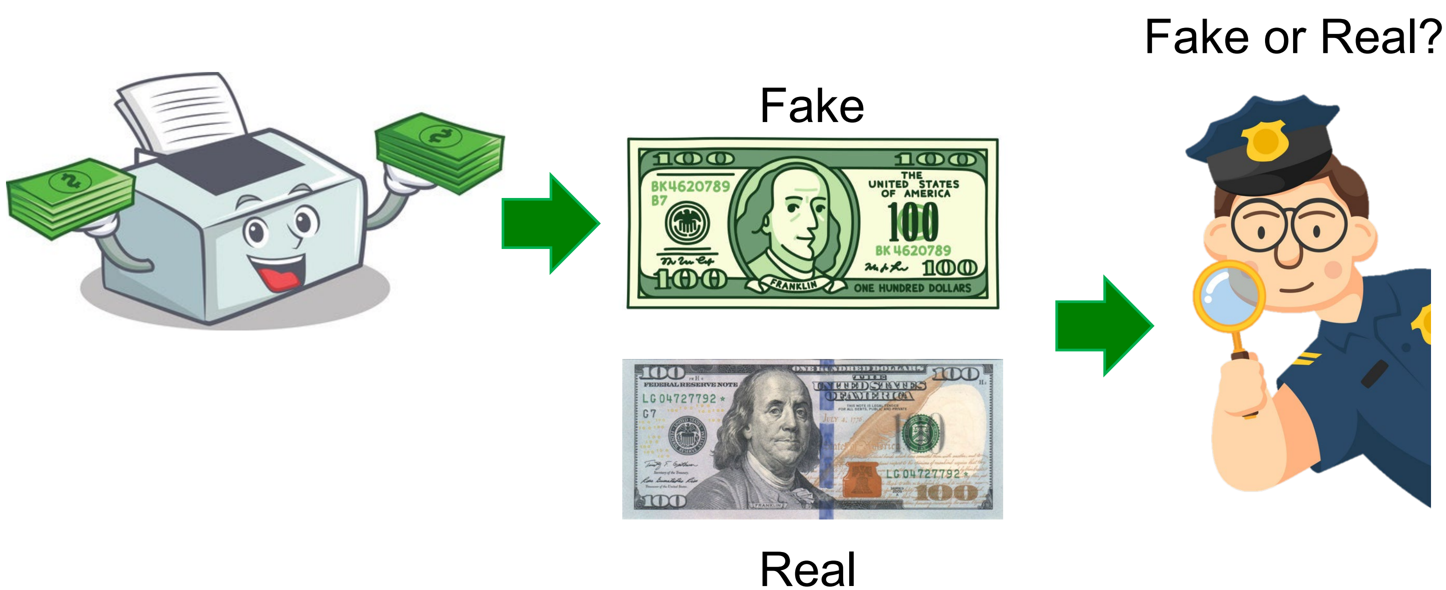
- Density estimation aims to estimate an unobservable underlying probability density function based on observed data
- Denote by $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ the set of observed data points, drawn from an unknown $p(\boldsymbol{x})$,

$$\boldsymbol{x}_i \sim p(\boldsymbol{x}), \text{for } i = 1, 2, \dots, N$$

The goal is to estimate the probability density function $p(\boldsymbol{x})$

State-of-the-Art Density Estimation

- Generative Adversarial Networks
  - The competition drives the counterfeiter to learn the distribution of real data.



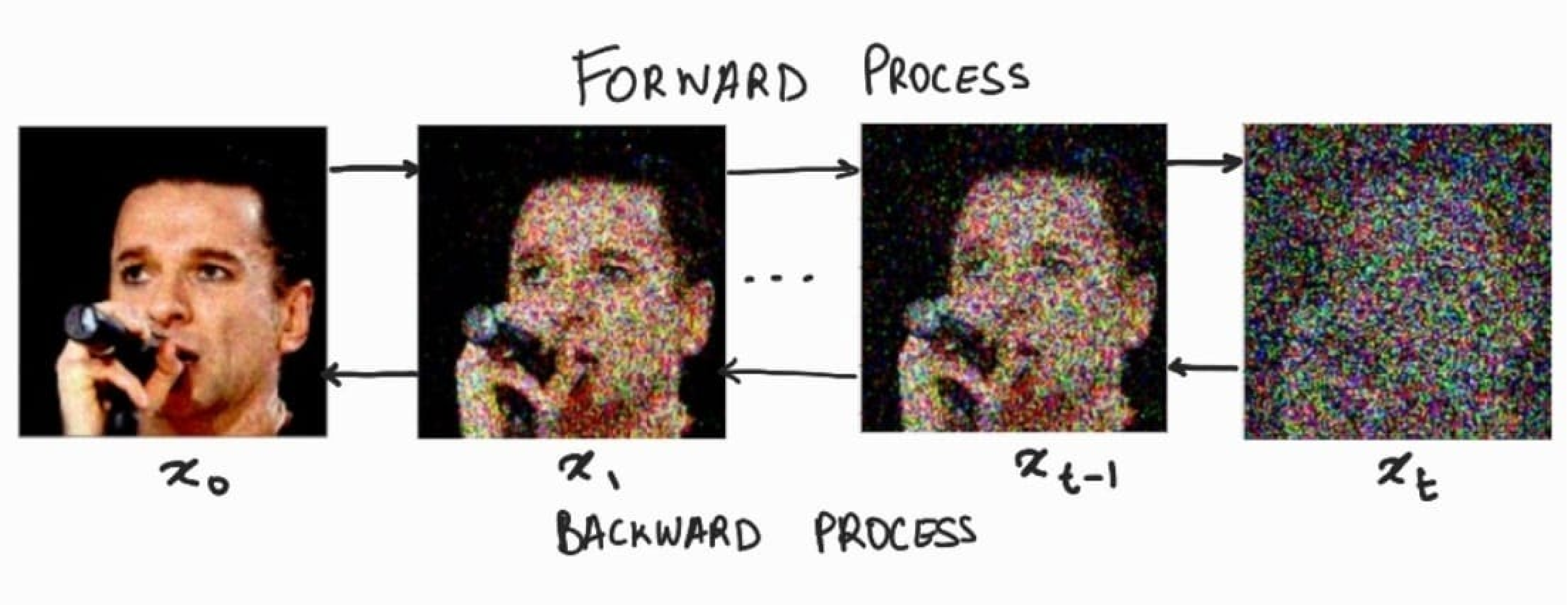
State-of-the-Art Density Estimation

- Generative Adversarial Networks
  - The competition drives the counterfeiter to learn the distribution of real data.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming Transformers for High-Resolution Image Synthesis, CVPR 2021.

State-of-the-Art Density Estimation

- Diffusion Models
  - Learn a sequence of transformations that create images from pure noise



https://medium.com/@vedantjumle/image-generation-with-diffusion-models-using-keras-and-tensorflow-9f60aae72ac

State-of-the-Art Density Estimation

- DALL·E v2, 2022
  - Sampling from the conditional distribution

    P(image | text)

An astronaut lounging in a tropical resort in a vaporwave style

Teddy bears mixing sparkling chemicals as mad scientists as digital art

State-of-the-Art Density Estimation

- Midjourney V4, 2022

A penguin in Venice

the adorable lambs sing rock opera, serious surreal, comic illustration, detailed, centered composition, uncropped, happy vine mood.

Robotics & AI

Stability AI, the startup behind Stable Diffusion, raises $101M

Kyle Wiggers @kyle_l_wiggers / 1:01 AM GMT+8 • October 18, 2022 Comment

Stable diffusion is an image generating AI, similar to DALL-E and Midjourney.

Why study? Whatever they teach in NTU has nothing to do with the real world.

Density Estimation Approaches

- Parametric density estimation
  - Assume a form for $p(\boldsymbol{x};\boldsymbol{\theta})$, defined up to parameters, $\boldsymbol{\theta}$
    - E.g., Guassian distribution $\mathcal{N}(\mu,\sigma^2)$, $\boldsymbol{\theta}=\{\mu,\sigma^2\}$
  - Estimate $\boldsymbol{\theta}$ from the observed data points
    - Maximum Likelihood Estimation
- Nonparametric density estimation

The General Principle

- The observed data points $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ are assumed to be a sample of N random variables independent and identically distributed (i.i.d.)
- Identically distributed: for any $\boldsymbol{x}_i \in \mathcal{D}$, it is sampled from the same probability distribution
- Independent: all the data points $\boldsymbol{x}_i \in \mathcal{D}$ are independent events

Parametric Density Estimation

- Assume that $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ are drawn from some known probability density family, $P(\boldsymbol{x}|\boldsymbol{\theta})$, defined up to parameters, $\boldsymbol{\theta}$

$$\boldsymbol{x}_i \sim P(\boldsymbol{x}|\boldsymbol{\theta})$$

- We seek $\boldsymbol{\theta}$ that makes $\boldsymbol{x}_i$ as likely as possible under $P(\boldsymbol{x}|\boldsymbol{\theta})$
- Approach: maximum likelihood estimation

Maximum Likelihood Estimation

- Likelihood of parameter $\boldsymbol{\theta}$ given sample $\mathcal{D}$:

$$l(\mathcal{D}; \boldsymbol{\theta}) \triangleq P(\mathcal{D}|\boldsymbol{\theta})$$

- As $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ are i.i.d., the above likelihood is the product of the likelihoods of the individual data points

$$l(\mathcal{D}; \boldsymbol{\theta}) \triangleq P(\mathcal{D}|\boldsymbol{\theta}) = \prod_{i=1}^{N} P(\boldsymbol{x}_i|\boldsymbol{\theta})$$

- In MLE, we aim to find $\boldsymbol{\theta}$ that makes $\mathcal{D}$ the most likely to be drawn from. Mathematically, we aim to search for $\widehat{\boldsymbol{\theta}}$ such that

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} l(\mathcal{D}; \boldsymbol{\theta})$$

MLE (cont.)

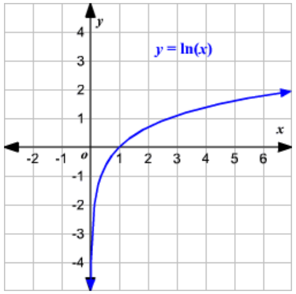
- Typically, we maximize the log-likelihood:

$$\mathcal{L}(\mathcal{D};\boldsymbol{\theta}) \triangleq \ln l(\mathcal{D};\boldsymbol{\theta})$$

- Why?
  1. The $\ln(\cdot)$ function converts the product into a sum

$$\ln l(\mathcal{D};\boldsymbol{\theta}) = \ln P(\mathcal{D}|\boldsymbol{\theta}) = \ln\left(\prod_{i=1}^{N} P(\boldsymbol{x}_i|\boldsymbol{\theta})\right) = \sum_{i=1}^{N} \ln P(\boldsymbol{x}_i|\boldsymbol{\theta})$$

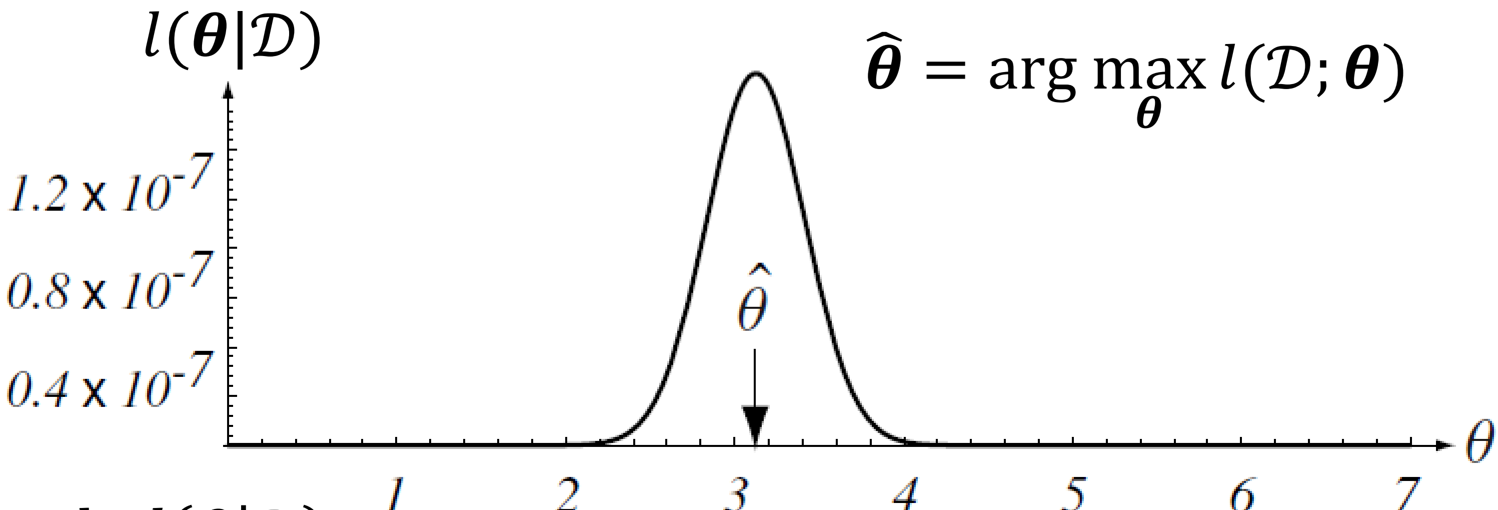
  2. The $\ln(\cdot)$ function is a strictly increasing function, one can maximize the likelihood without changing the value where it takes its maximum



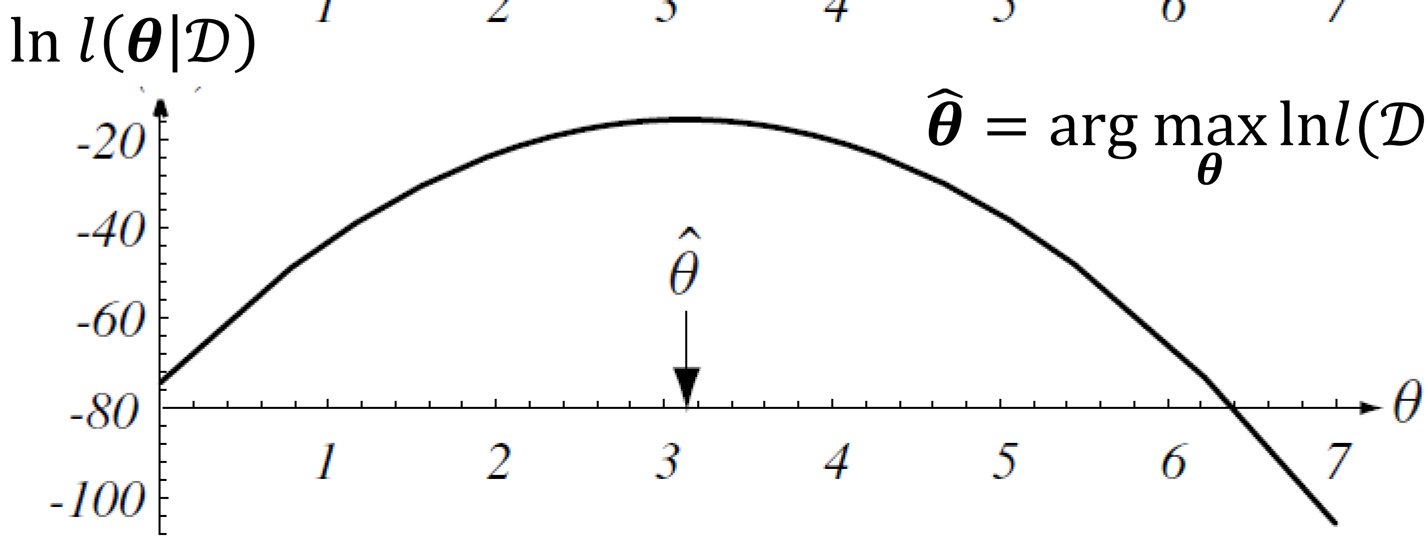
$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} l(\mathcal{D};\boldsymbol{\theta}) \Leftrightarrow \widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \ln l(\mathcal{D};\boldsymbol{\theta})$$

MLE Illustration

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} l(\mathcal{D};\boldsymbol{\theta})$$



$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \ln l(\mathcal{D};\boldsymbol{\theta})$$



Solution of MLE

- Suppose $\boldsymbol{\theta}$ contains p parameters,

$$\boldsymbol{\theta} = [\theta_1\ \theta_2\ \ \dots\ \theta_p]^T$$

- $\max_{\boldsymbol{\theta}} \ln l(\mathcal{D}; \boldsymbol{\theta})$ is an unconstrained optimization problem.
  To solve it, we first set the derivative of $\ln l(\mathcal{D}; \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ to zero

$$\boxed{\nabla_{\boldsymbol{\theta}}} \ln l(\mathcal{D}; \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \left(\sum_{i=1}^{N} \ln P(\boldsymbol{x}_i | \boldsymbol{\theta}) \right) = \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} \ln P(\boldsymbol{x}_i | \theta) = \mathbf{0},$$

Gradient operator

$$\nabla_{\boldsymbol{\theta}} = \left[\frac{\partial}{\partial \theta_1}\ \frac{\partial}{\partial \theta_2}\ \cdots\ \frac{\partial}{\partial \theta_p} \right]^T$$

- We then obtain a solution $\widehat{\boldsymbol{\theta}}$ by solving the above system of equations

Univariate Gaussian

- Suppose $\mathcal{D} = \{x_1, x_2, \dots, x_N\}$. Each data point x_i is a scalar, and is drawn from a Guassian distribution with unknown μ and σ^2:

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

$$\mu = \boxed{\mathbb{E}}[x] = \int P(x; \mu, \sigma^2) x dx$$

$$\sigma^2 = \mathrm{Var}(x) = \mathbb{E}[(x - \mathbb{E}[x])(x - \mathbb{E}[x])^T]$$

Expectation, considering all possible values (infinite)

Univariate Gaussian (cont.)

- The log-likelihood is

$$\ln l(\mathcal{D};\boldsymbol{\theta}) = \sum_{i=1}^{N} \ln P(x_i|\boldsymbol{\theta})$$

$$p(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{(\boldsymbol{x}-\mu)^2}{2\sigma^2}\right)}$$

$$= \sum_{i=1}^{N} \cancel{\ln\left(\exp\right.}\left.\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)\right) - \sum_{i=1}^{N} \ln(\sqrt{2\pi\sigma^2})$$

$$= -\frac{\sum_{i=1}^{N}(x_i-\mu)^2}{2\sigma^2} - \frac{N}{2}\ln(2\pi\sigma^2)$$

$$= -\frac{\sum_{i=1}^{N}(x_i-\mu)^2}{2\sigma^2} - \frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln\sigma^2$$

Univariate Gaussian (cont.)

- The log-likelihood:

$$\ln l(\mathcal{D};\boldsymbol{\theta}) = -\frac{\sum_{i=1}^{N}(x_i-\mu)^2}{2\sigma^2} - \frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln\sigma^2$$

- The derivative of the log-likelihood:

$$\nabla_{\boldsymbol{\theta}}\ln l(\mathcal{D};\boldsymbol{\theta}) = \sum_{i=1}^{N}\nabla_{\boldsymbol{\theta}}\ln P(x_i|\theta) = \begin{bmatrix} \sum_{i=1}^{N}\nabla_{\mu}\ln P(x_i|\theta) \\ \sum_{i=1}^{N}\nabla_{\sigma^2}\ln P(x_i|\theta) \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sigma^2}\sum_{i=1}^{N}(x_i-\mu) \\ \frac{\sum_{i=1}^{N}(x_i-\mu)^2}{2(\sigma^2)^2} - \frac{N}{2\sigma^2} \end{bmatrix}$$

Univariate Gaussian (cont.)

- By setting the derivative to be zero:

$$\begin{cases} \dfrac{1}{\sigma^2}\displaystyle\sum_{i=1}^{N}(x_i-\mu)=0 \\ \dfrac{\sum_{i=1}^{N}(x_i-\mu)^2}{2(\sigma^2)^2}-\dfrac{N}{2\sigma^2}=0 \end{cases}$$

- We have

$$\begin{cases} \hat{\mu}=\dfrac{1}{N}\displaystyle\sum_{i=1}^{N}x_i \\ \hat{\sigma}^2=\dfrac{1}{N}\displaystyle\sum_{i=1}^{N}(x_i-\hat{\mu})^2 \end{cases}$$

Unbiased Estimator

- An estimator is a rule for estimating a quantity based on observations.
- The MLE estimator for the Gaussian mean is $\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$
- An estimator is unbiased if its expectation is the same as the true quantity.

Unbiased estimation

$$\mathbb{E}[\hat{\mu}] = \boxed{\mu}$$

True

$$\mathbb{E}[\hat{\sigma}^2] = \frac{N-1}{N}\boxed{\sigma^2}$$

Biased estimation

To correct bias

$$\tilde{\sigma}^2 = \frac{N}{N-1}\hat{\sigma}^2$$

$$= \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \hat{\mu})^2$$

Wait… Expectation of an Estimator?

- How can you talk about an expectation without a probability distribution?
- Imagine you want to estimate the average height of Singaporeans.
- Conceptually simple: measure everyone!
- Actually feasible: measure 100 randomly selected Singaporeans.
- However, the estimate will change depending on who are selected.
- These different estimators form a distribution.
- The calculation of this distribution is out of the scope of this course.

Multivariate Gaussian

- Suppose $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$, and each data point $\boldsymbol{x}_i$ has d dimensions, and is drawn from a Guassian distribution with unknown $\boxed{\boldsymbol{\mu}}$ and $\boxed{\boldsymbol{\Sigma}}$:

d-dimensional mean vector

$d \times d$ covariance matrix

$$P(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right)$$

$|\boldsymbol{\Sigma}|$: the determinant of $\boldsymbol{\Sigma}$

$$\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{x}] = \int P(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \boldsymbol{x} d\boldsymbol{x}$$

$$\boldsymbol{\Sigma} = \mathbb{E}[(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])^T]$$

Covariance Matrix $\boldsymbol{\Sigma}$

Variance of the first dimension

Covariance of the first and second dimensions

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boxed{\mathrm{Var}(x^{(1)})} & \boxed{\mathrm{Cov}(x^{(1)}, x^{(2)})} & \dots & \mathrm{Cov}(x^{(1)}, x^{(d)}) \\ \mathrm{Cov}(x^{(2)}, x^{(1)}) & \mathrm{Var}(x^{(2)}) & \dots & \mathrm{Cov}(x^{(2)}, x^{(d)}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(x^{(d)}, x^{(1)}) & \mathrm{Cov}(x^{(d)}, x^{(2)}) & \dots & \mathrm{Var}(x^{(d)}) \end{bmatrix}$$

$$\mathrm{Var}\left(x^{(i)}\right) = \mathbb{E}\left[\left(x^{(i)} - \mathbb{E}\left[x^{(i)}\right]\right)^2\right]$$

$$\begin{aligned} \mathrm{Cov}(x^{(i)}, x^{(j)}) &= \mathbb{E}\left[\left(x^{(i)} - \mathbb{E}\left[x^{(i)}\right]\right)\left(x^{(j)} - \mathbb{E}\left[x^{(j)}\right]\right)\right] \\ &= \int \left(x^{(i)} - \mathbb{E}\left[x^{(i)}\right]\right)\left(x^{(j)} - \mathbb{E}\left[x^{(j)}\right]\right) P\left(x^{(i)}\right) P\left(x^{(j)}\right) dx^{(i)} dx^{(j)} \end{aligned}$$

Multivariate Gaussian (cont.)

- The log-likelihood of multivariate Gaussian is

$$\ln l(\mathcal{D};\boldsymbol{\theta}) = \sum_{i=1}^{N} \ln P(\boldsymbol{x}_i;\boldsymbol{\mu},\boldsymbol{\Sigma}) = -\frac{Nd}{2}\ln(2\pi) - \frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{x}_i-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})$$

- The derivative of the log-likelihood w.r.t. $\boldsymbol{\mu}$ is

$$\frac{\partial}{\partial\boldsymbol{\mu}}\ln l(\mathcal{D};\boldsymbol{\theta}) = \sum_{i=1}^{N}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})$$

- By setting the derivative to zero, we have

$$\widehat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i$$

Unbiased estimation:

$$\mathbb{E}[\widehat{\boldsymbol{\mu}}] = \boldsymbol{\mu}$$

Multivariate Gaussian (cont.)

- By computing the derivative of the log-likelihood w.r.t. $\boldsymbol{\Sigma}$ (each Σ_{ij}, i.e., the entry of the i-th row and the j-th column), and setting to be zero, we have

$$\widehat{\boldsymbol{\Sigma}}_{\mathrm{MLE}} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T$$

MLE estimator

- The above estimation is biased!

$$\mathbb{E}[\widehat{\boldsymbol{\Sigma}}] = \frac{N-1}{N}\boldsymbol{\Sigma}$$

- We can correct the bias by timing $\widehat{\boldsymbol{\Sigma}}$ with $\frac{N}{N-1}$:

$$\widetilde{\boldsymbol{\Sigma}}_{\mathbf{Unbiased}} = \frac{1}{N-1}\sum_{i=1}^{N}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T$$

Unbiased estimator (not MLE)

Summary: Estimating Multivariate Gaussian

- Suppose $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$. Each data point $\boldsymbol{x}_i$ has d dimensions, and is drawn from a Guassian distribution with unknown $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$:

$$P(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- The unbiased estimators for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are

$$\widehat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i$$

$$\widetilde{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T$$

Note: in this module, whenever you are asked to estimate the mean vector and the covariance matrix of data samples, use the unbiased estimators

Summary: MLE for General Distributions

- Suppose $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$. Each data point $\boldsymbol{x}_i$ is of d dimensions, and drawn from a distribution with unknown parameter $\boldsymbol{\theta}$

$$\boldsymbol{x}_i \sim P(\boldsymbol{x}|\boldsymbol{\theta})$$

- Step 1: Compute the likelihood of $\boldsymbol{\theta}$ given sample $\mathcal{D}$:

$$l(\mathcal{D}; \boldsymbol{\theta}) \triangleq P(\mathcal{D}|\boldsymbol{\theta}) = \prod_{i=1}^{N} P(\boldsymbol{x}_i|\boldsymbol{\theta})$$

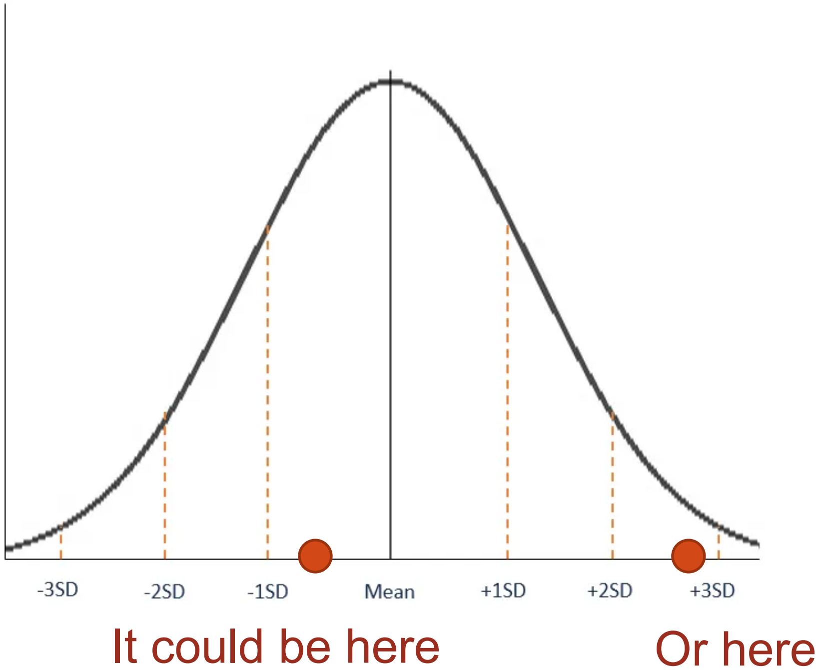
- Step 2: Compute the log-likelihood of $\boldsymbol{\theta}$ given $\mathcal{D}$: $\ln l(\mathcal{D}; \boldsymbol{\theta})$
- Step 3: Compute the derivative of $\ln l(\boldsymbol{\theta}|\mathcal{D})$ w.r.t. $\boldsymbol{\theta}$ and set it to zero:

$$\nabla_{\boldsymbol{\theta}} \ln l(\boldsymbol{\theta}|\mathcal{D}) = \mathbf{0}$$

- Step 4: Solve the above system of equations to obtain the maximum likelihood estimate $\widehat{\boldsymbol{\theta}}$

Is MLE always the best choice?

- What if you only have a single data point $\boldsymbol{x_1}$?
- The MLE $\boldsymbol{\mu} = \boldsymbol{x_1}$
- Estimating the entire distribution based on a single data point seems unwise.



The Bayesian Approach

- We may introduce a prior distribution $P(\boldsymbol{\theta})$ that represents our belief about θ in the absence of any data.

$$\underbrace{P(\boldsymbol{\theta}|\mathcal{D})}_{\text{Posterior}} = \frac{\overbrace{P(\mathcal{D}|\boldsymbol{\theta})}^{\text{Likelihood}}\overbrace{P(\boldsymbol{\theta})}^{\text{Prior}}}{P(\mathcal{D})}$$

$$= \frac{P(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{\underbrace{\int P(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta})d\boldsymbol{\theta}}_{\text{Normalization constant}}}$$

The Bayesian Approach

- If the likelihood is overly concentrated, the prior can provide some smoothing effect.

$$\underbrace{P(\boldsymbol{\theta}|\mathcal{D})}_{\text{Posterior}} = \frac{\overbrace{P(\mathcal{D}|\boldsymbol{\theta})}^{\text{Likelihood}}\overbrace{P(\boldsymbol{\theta})}^{\text{Prior}}}{P(\mathcal{D})}$$

$$= \frac{P(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{\underbrace{\int P(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta})d\boldsymbol{\theta}}_{\text{Normalization constant}}}$$

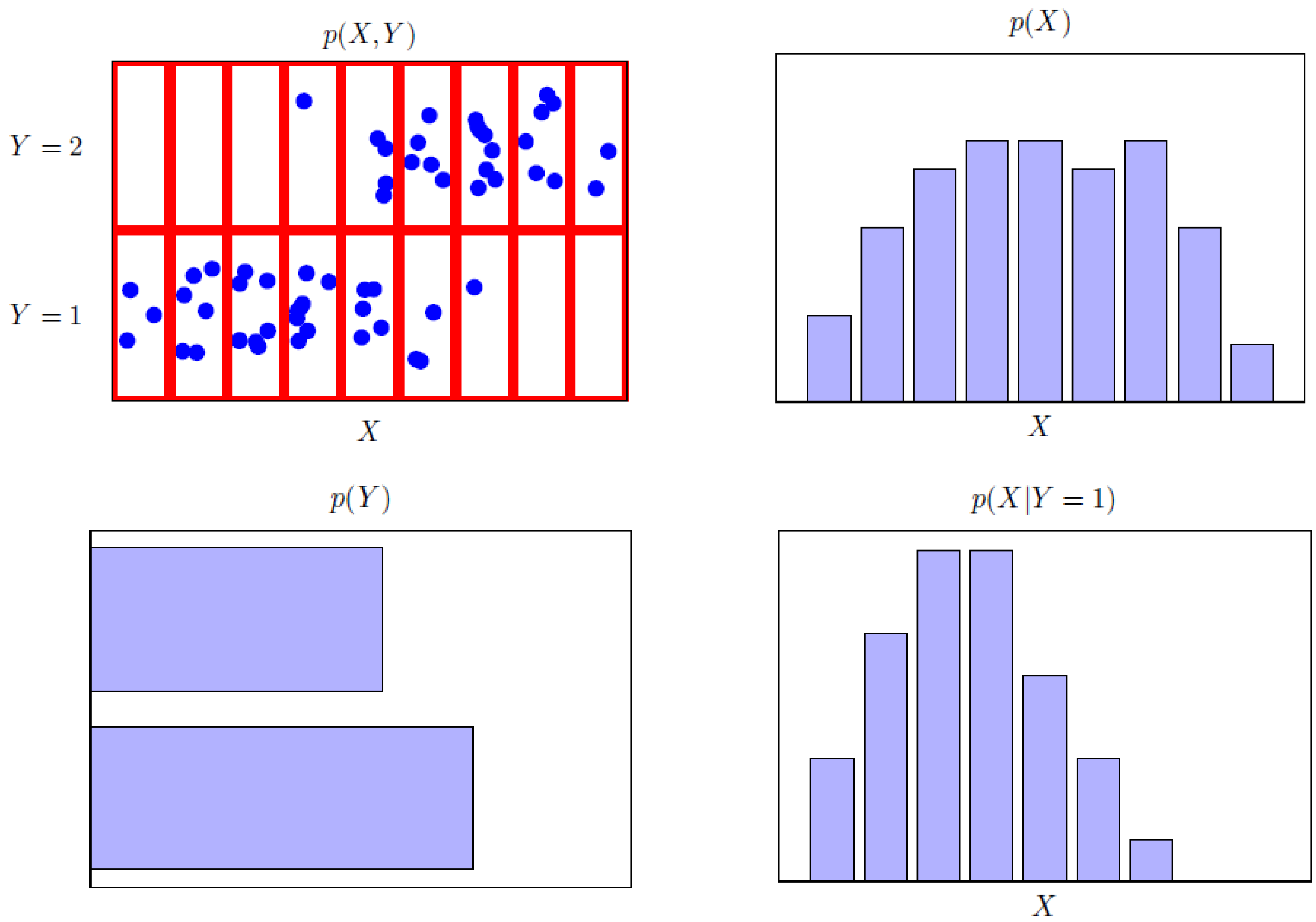
Density Estimation Approaches

- Parametric density estimation
- Nonparametric density estimation
  - Without assuming any forms for the underlying density
  - Assume that similar inputs have similar outputs: if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar, then $P(\boldsymbol{x}_i)$ and $P(\boldsymbol{x}_j)$ are similar
  - Approaches
    - Histogram Estimator
    - Naïve Estimator / Parzen Windows / Kernel Estimator
    - K-NN Estimator

Non-Parametric Density Estimation

- Assume that $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ are drawn from some unknown probability density $P(\boldsymbol{x})$
- To learn the estimator $\hat{P}(\boldsymbol{x})$ for $P(\boldsymbol{x})$
- We first focus on the univariate case, i.e., x_i is scalar
- Note that the introduced approaches can be generalized to the multivariate case easily

Histogram Estimator



Histogram Estimator (cont.)

- Simply partition x into distinct bins of a fixed width Δ
- Count the number N_t of data points falling into bin t
- Turn this count into a normalized probability density via dividing by the total number of observed data points N and by the width Δ of the bins:

$$p_t = \frac{N_t}{N\boxed{\Delta}}$$

Why divide by Δ?

- The model for the density $p(\boldsymbol{x})$ is constant over the width of each bin: find the bin where $\boldsymbol{x}$ is in (e.g., bin t), then

$$\hat{P}(\boldsymbol{x}) = \frac{\#\{\boldsymbol{x}_i \mid \boldsymbol{x}_i \text{ in the same bin as } \boldsymbol{x}\}}{N\Delta} = P_t$$

Histogram Estimator (cont.)

- For a bin t, given a density function, the probability that a data instance falling into the bin t is

$$P_t = \int_{\Delta} p(\boldsymbol{x})d\boldsymbol{x} \ = p_t(\boldsymbol{x})\Delta$$

- On the other hand,

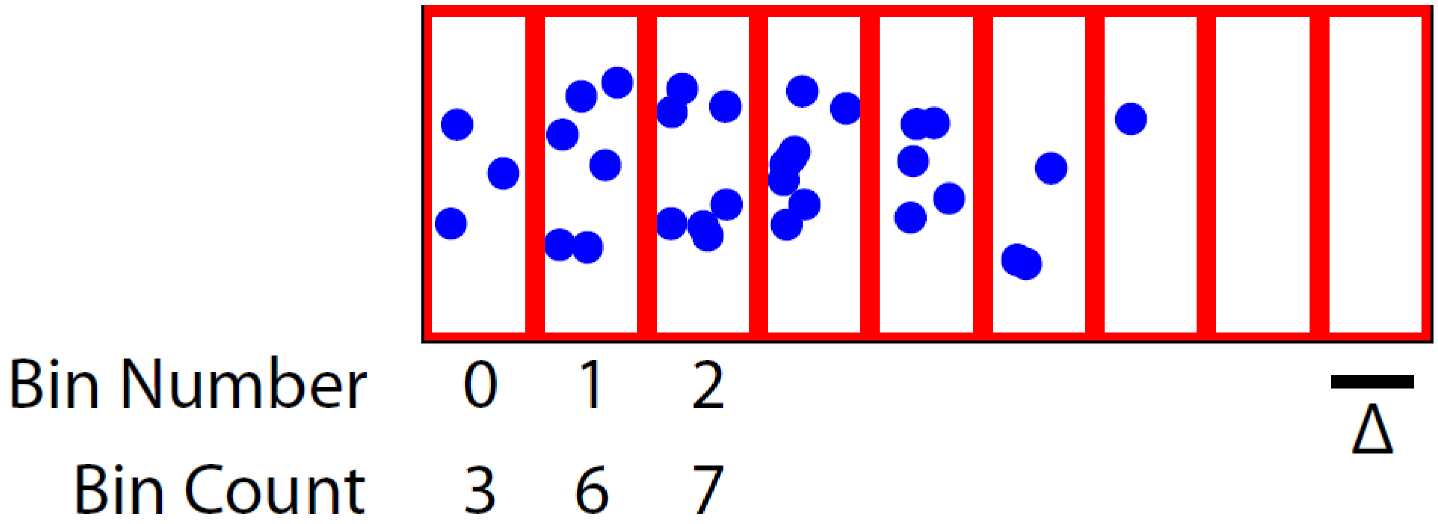
$$P_t = \frac{\#\{\boldsymbol{x}_i \mid \boldsymbol{x}_i \text{ in bin } t\}}{N} = \frac{N_t}{N}$$

- Therefore:

$$P_t(\boldsymbol{x})\Delta = \frac{N_t}{N} \Rightarrow P_t(\boldsymbol{x}) = \frac{N_t}{N\Delta}$$

Histogram Estimator (cont.)

- Note: different bins may have different widths Δ_t in general, but in practice, we use the same width Δ

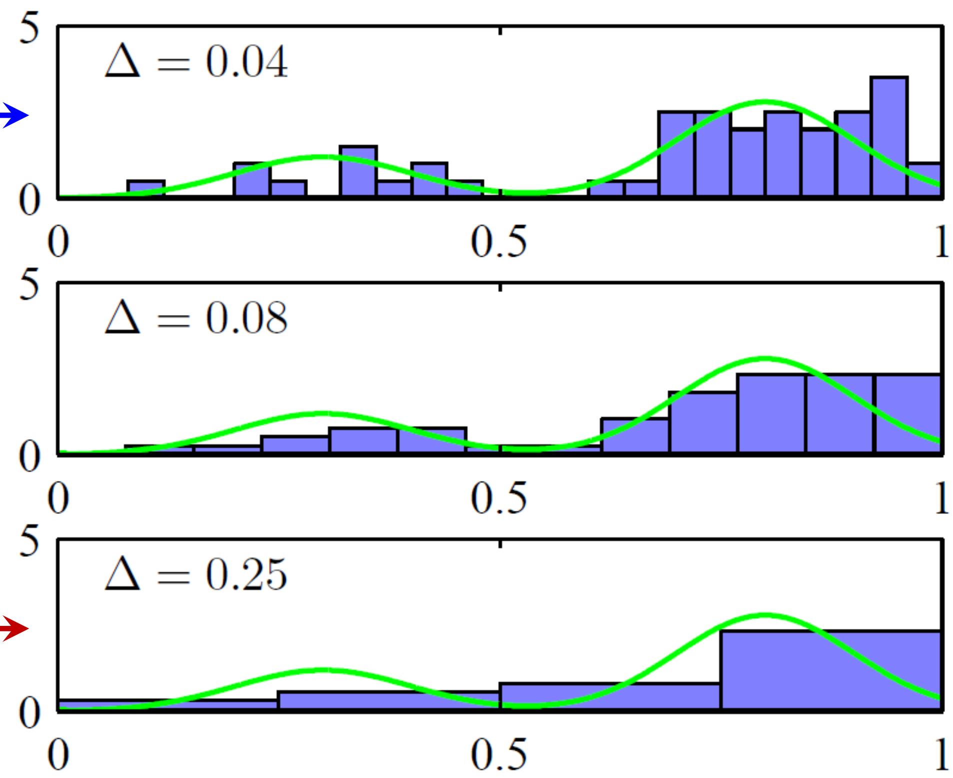


Histogram Estimator (cont.)

- Histogram density as a function of bin width Δ

The green curve is the underlying true density from which the data points were drawn

When Δ is very small, the resulting density is quite spiky and hallucinates a lot of structure not present in the true density



When Δ is very big, the resulting density is quite smooth and consequently fails to capture the bimodality of the true density

Analysis on Histogram Estimator

- Advantages:
  - Simple to evaluate and simple to use
  - One can throw away $\mathcal{D}$ once the histogram is computed
  - Can be updated incrementally
- Disadvantages:
  - The estimated density has discontinuities due to the bin edges rather than any property of the underlying density
  - Scales poorly to multivariate cases: we would have m^d *bins* (hypercubes) if we divided each feature (dimension) in a d-dimensional space into m bins

Naïve Estimator: An Alterative

- In Histogram Estimator, besides Δ, we have to choose an origin x_0 as well, the bins are the intervals defined as

$$[\boldsymbol{x}_0 + \boxed{m}\Delta, \boldsymbol{x}_0 + (m+1)\Delta)$$

m is an integer

- The Naïve Estimator does not need to set an origin

$$\hat{p}(\boldsymbol{x}) = \frac{\#\{\boldsymbol{x}_i \mid \boldsymbol{x}_i \text{ in the same bin as } \boldsymbol{x}\}}{N\Delta}$$

Given $\boldsymbol{x}$, use $\boldsymbol{x}$ as a center to create a bin with a length of Δ

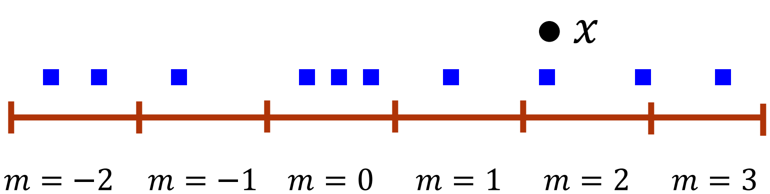
$$[\xleftrightarrow{\frac{\Delta}{2}} \boldsymbol{x} \xleftrightarrow{\frac{\Delta}{2}}]$$

$$\hat{p}(\boldsymbol{x}) = \frac{\#\left\{\boldsymbol{x}_i \mid \boxed{\boldsymbol{x} - \frac{\Delta}{2} \le \boldsymbol{x}_i < \boldsymbol{x} + \frac{\Delta}{2}}\right\}}{N\Delta}$$

Histogram v.s. Naïve Estimator

Histogram Estimator

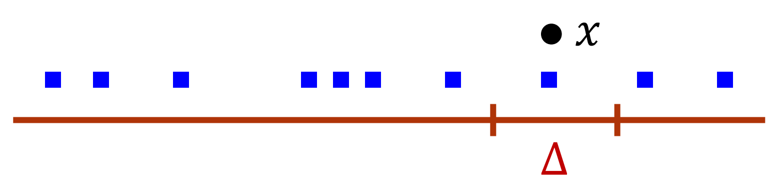
$$p(x) = \frac{2}{10 \times \Delta}$$



Count: 2 1 3 1 2 1

Naïve Estimator

$$p(x) = \frac{1}{10 \times \Delta}$$



1

Naïve Estimator: An Alterative (cont.)

$$\hat{P}(\boldsymbol{x}) = \frac{\#\{\boldsymbol{x}_i \mid \boldsymbol{x} - \frac{\Delta}{2} \le \boldsymbol{x}_i < \boldsymbol{x} + \frac{\Delta}{2}\}}{N\Delta}$$

The Naïve Estimator can also be written as

$$\hat{P}(\boldsymbol{x}) = \frac{1}{N\Delta}\sum_{i=1}^{N} \boxed{w}\left(\frac{\boldsymbol{x}_i - \boldsymbol{x}}{\Delta}\right)$$

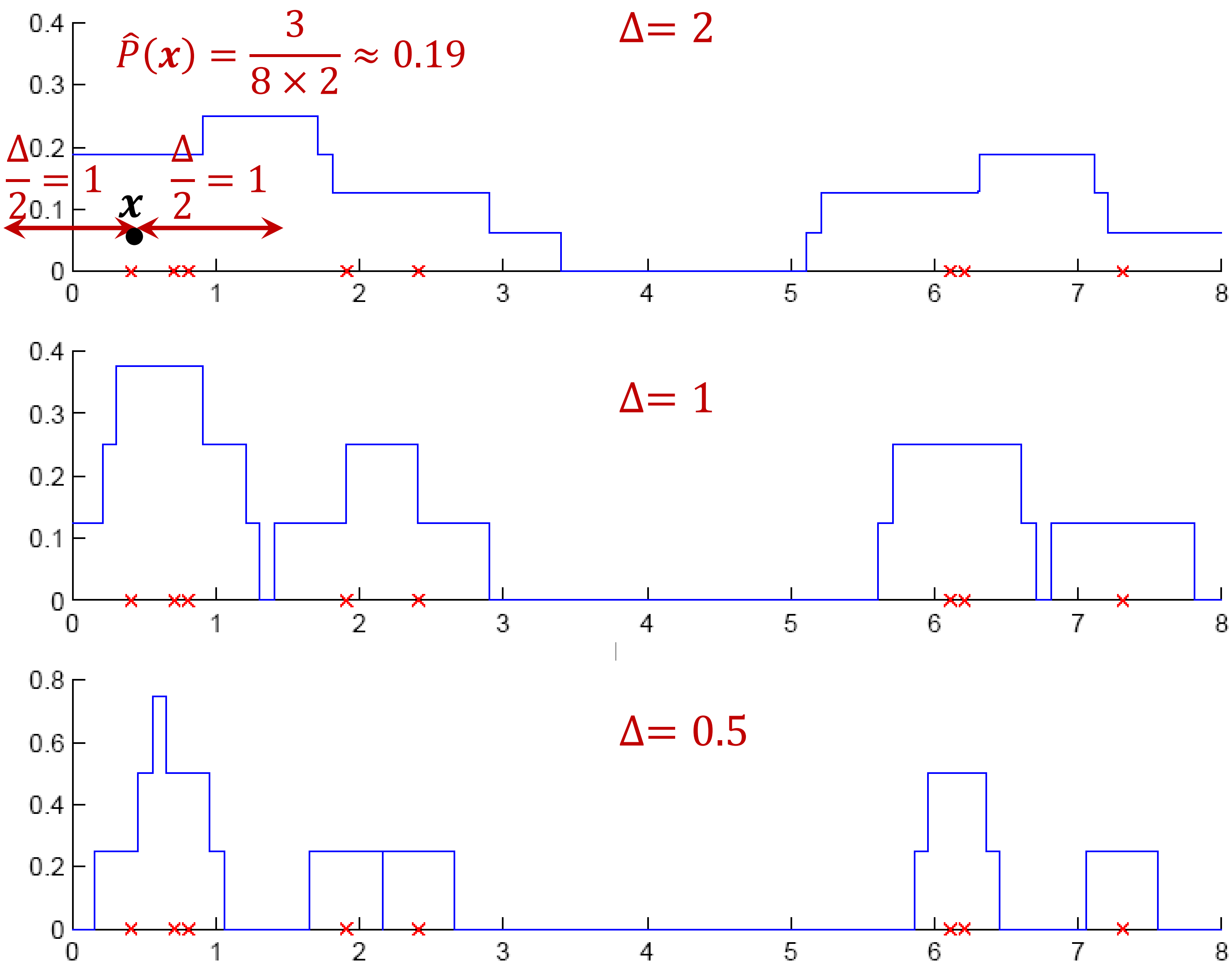
If x_i is in the bin with width Δ centered at x, then the count is increased by 1

$$u = \frac{\boldsymbol{x}_i - \boldsymbol{x}}{\Delta}$$

Windowing function $w(u) = \begin{cases} 1 & \text{if } -\frac{1}{2} \le \boxed{u} < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$

Parzen windows

Naïve Estimator



Generalization to Multivariate

- Suppose the observed data points $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ are d-dimensional
- In a d-dimensional space, we define $\mathcal{R}$ is a d-dimensional hypercube with h being the length of each edge. Then the volume of the hypercube is given by

$$V = h^d$$

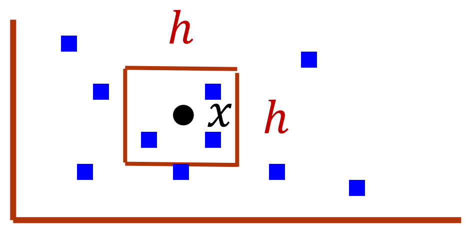
- The windowing function can be defined as

$$w(\boxed{\boldsymbol{u}}) = \begin{cases} 1 & \text{if } -\frac{1}{2} \le u_j < \frac{1}{2} \text{ for all } j \in \{1,2,\dots,d\} \\ 0 & \text{otherwise} \end{cases}$$

$$\boldsymbol{u} = \frac{\boldsymbol{x}_i - \boldsymbol{x}}{h}$$

Defines a hypercube of length h centered at $\boldsymbol{x}$, if $\boldsymbol{x}_i$ falls in the cube, then the count is increased by 1

E.g., $d = 2$



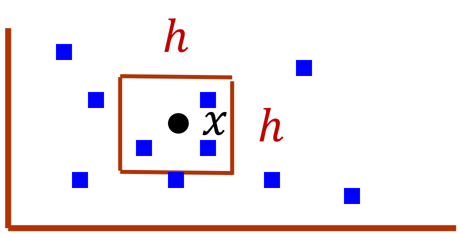
Generalization to Multivariate (cont.)

- Hence, $w\left(\frac{\boldsymbol{x}_i-\boldsymbol{x}}{h}\right)$ is equal to unity if $\boldsymbol{x}_i$ falls within the hypercube of volume V centered at $\boldsymbol{x}$, and is zero otherwise
- The density estimator can be written as

$$\hat{P}(\boldsymbol{x}) = \frac{\#\{\boldsymbol{x}_i \mid \boldsymbol{x}_i \text{ in the same hypercube as } \boldsymbol{x}\}}{NV}$$

$$\hat{P}(\boldsymbol{x}) = \frac{1}{NV}\sum_{i=1}^{N} w\left(\frac{\boldsymbol{x}_i-\boldsymbol{x}}{h}\right)$$

Parzen window function can be a kernel function – Kernel estimator (Appendix, optional)



$$\hat{P}(\boldsymbol{x}) = \frac{3}{10\boxed{h^2}}$$

K-Nearest Neighbor Estimator

- Recall

$$P(\boldsymbol{x}) = \frac{\overbrace{\boxed{\#\{\boldsymbol{x}_i \mid \boldsymbol{x}_i \text{ in the same hypercube as } \boldsymbol{x}\}}}^{K_{\boldsymbol{x}}}}{VN}$$

- In the previous approaches, V (or Δ for univariate) is fixed for different queries $\boldsymbol{x}$'s
- The K-NN Estimator *adapts* the amount of smoothing to the *local* density of data, and the degree of smoothing is controlled by K, the number of neighbors

Consider K nearest neighbors of $\boldsymbol{x}$

$$P(\boldsymbol{x}) = \frac{\boxed{K}}{N\boxed{V_{\boldsymbol{x}}}}$$

The volume of the space centered at $\boldsymbol{x}$ that exactly contains K nearest neighbors of $\boldsymbol{x}$

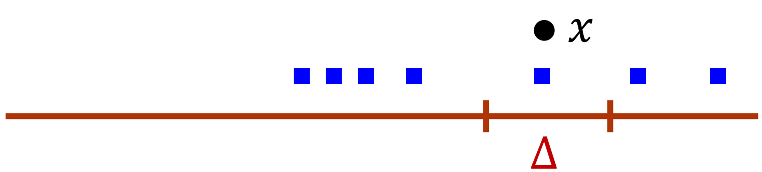
- Note that the number of nearest neighbors is typically much smaller than the number of training data points, i.e., $K \ll N$

Naïve Estimator v.s. K-NN Estimator

Univariate Case

Naïve Estimator

$$\hat{P}(\boldsymbol{x}) = \frac{\#\{\boldsymbol{x}_i \mid \boldsymbol{x} - \frac{\Delta}{2} \le \boldsymbol{x}_i < \boldsymbol{x} + \frac{\Delta}{2}\}}{N\Delta}$$

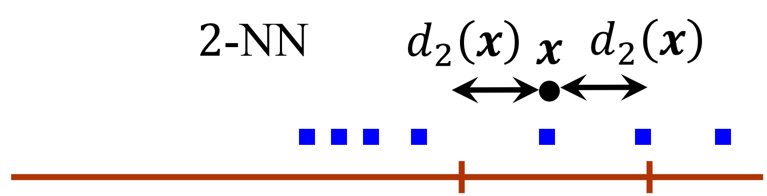


$$P(\boldsymbol{x}) = \frac{1}{7 \times \Delta}$$

- Fix Δ, and check how many data points fall in the bin

K-NN Estimator

$$\hat{P}(\boldsymbol{x}) = \frac{K}{N\left(2d_K(\boldsymbol{x})\right)}$$



$$P(\boldsymbol{x}) = \frac{2}{7 \times (2 \times d_2(\boldsymbol{x}))}$$

- Fix K, the number of observed data points to fall in the bin, and compute the bin size

K-Nearest Neighbor Estimator (cont.)

- With a predefined K, for a particular data point $\boldsymbol{x}$,
  1. Compute distance between $\boldsymbol{x}$ and all the observed data, e.g., Euclidean distance $\|\boldsymbol{x} - \boldsymbol{x}_i\|_2$
  2. Sort the observed data points based on the distances in ascending order:

$$\boxed{d_1(\boldsymbol{x})} \leq d_2(\boldsymbol{x}) \leq \cdots \boxed{d_j(\boldsymbol{x})} \leq \cdots \leq d_N(\boldsymbol{x})$$

$d_1(\boldsymbol{x})$ is the distance of $\boldsymbol{x}$ to the nearest observed instance

$d_j(\boldsymbol{x})$ is the distance of $\boldsymbol{x}$ to the j-th nearest observed instance

  3. The K-NN density estimate is

$$\hat{P}(\boldsymbol{x}) = \frac{K}{N \boxed{V_{d_K(\boldsymbol{x})}}}$$

A ball in d-dimensional Euclidean space

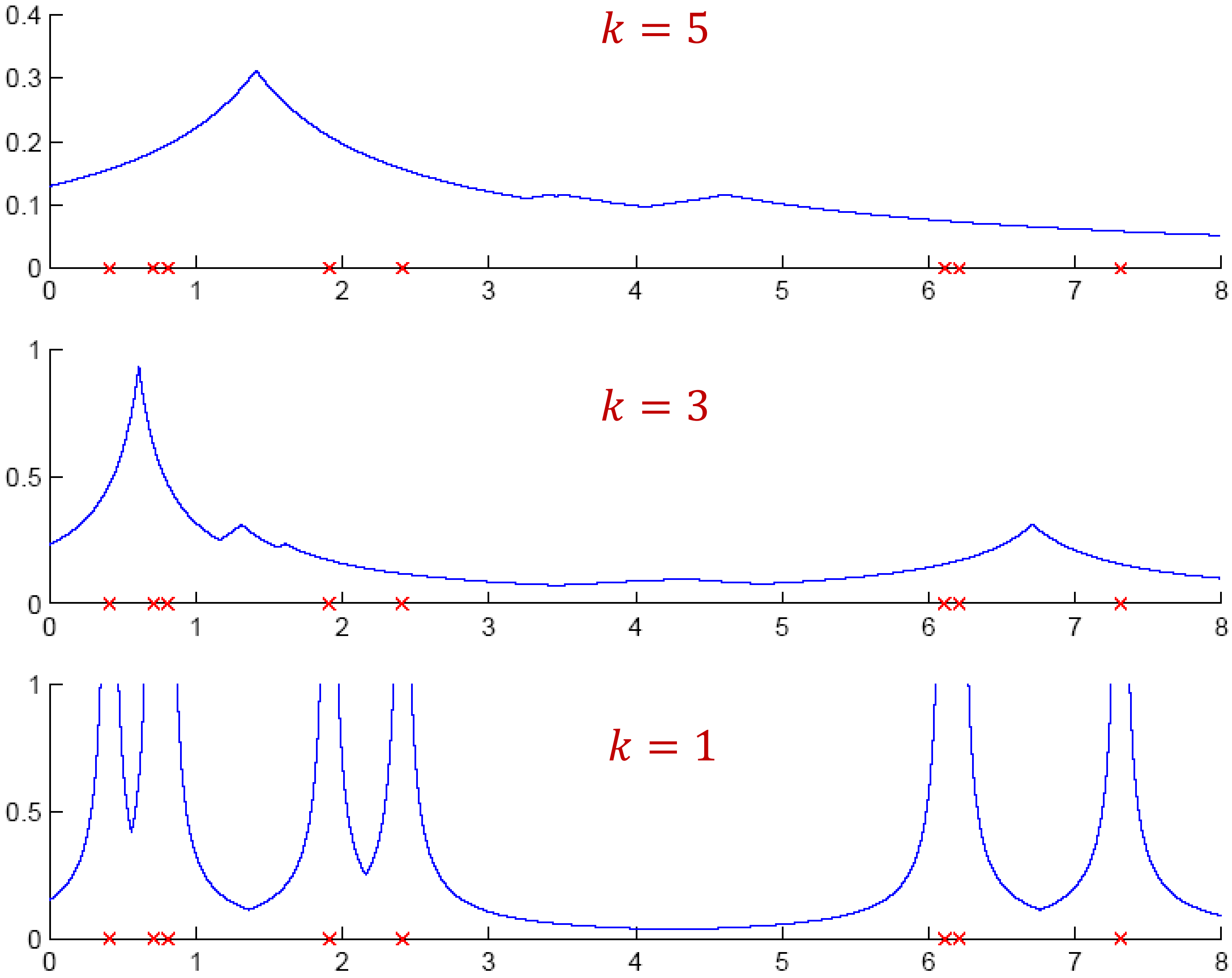
The volume of the $\boxed{d\text{-ball}}$ of the radius $d_K(\boldsymbol{x})$ centered at $\boldsymbol{x}$. And $d_K(\boldsymbol{x})$ is the distance of $\boldsymbol{x}$ to the K-th nearest observed instance

Look-up table for Volume of an n-ball

https://en.wikipedia.org/wiki/Volume_of_an_n-ball

Dimension	Volume of a ball of radius R	Radius of a ball of volume V
0	1	(all 0-balls have volume 1)
1	$2R$	$\frac{V}{2} = 0.5 \times V$
2	$\pi R^2 \approx 3.142 \times R^2$	$\frac{V^{\frac{1}{2}}}{\sqrt{\pi}} \approx 0.564 \times V^{\frac{1}{2}}$
3	$\frac{4\pi}{3} R^3 \approx 4.189 \times R^3$	$\left(\frac{3V}{4\pi}\right)^{\frac{1}{3}} \approx 0.620 \times V^{\frac{1}{3}}$
4	$\frac{\pi^2}{2} R^4 \approx 4.935 \times R^4$	$\frac{(2V)^{\frac{1}{4}}}{\sqrt{\pi}} \approx 0.671 \times V^{\frac{1}{4}}$
5	$\frac{8\pi^2}{15} R^5 \approx 5.264 \times R^5$	$\left(\frac{15V}{8\pi^2}\right)^{\frac{1}{5}} \approx 0.717 \times V^{\frac{1}{5}}$
6	$\frac{\pi^3}{6} R^6 \approx 5.168 \times R^6$	$\frac{(6V)^{\frac{1}{6}}}{\sqrt{\pi}} \approx 0.761 \times V^{\frac{1}{6}}$
7	$\frac{16\pi^3}{105} R^7 \approx 4.725 \times R^7$	$\left(\frac{105V}{16\pi^3}\right)^{\frac{1}{7}} \approx 0.801 \times V^{\frac{1}{7}}$
8	$\frac{\pi^4}{24} R^8 \approx 4.059 \times R^8$	$\frac{(24V)^{\frac{1}{8}}}{\sqrt{\pi}} \approx 0.839 \times V^{\frac{1}{8}}$
9	$\frac{32\pi^4}{945} R^9 \approx 3.299 \times R^9$	$\left(\frac{945V}{32\pi^4}\right)^{\frac{1}{9}} \approx 0.876 \times V^{\frac{1}{9}}$
10	$\frac{\pi^5}{120} R^{10} \approx 2.550 \times R^{10}$	$\frac{(120V)^{\frac{1}{10}}}{\sqrt{\pi}} \approx 0.911 \times V^{\frac{1}{10}}$
11	$\frac{64\pi^5}{10395} R^{11} \approx 1.884 \times R^{11}$	$\left(\frac{10395V}{64\pi^5}\right)^{\frac{1}{11}} \approx 0.944 \times V^{\frac{1}{11}}$
12	$\frac{\pi^6}{720} R^{12} \approx 1.335 \times R^{12}$	$\frac{(720V)^{\frac{1}{12}}}{\sqrt{\pi}} \approx 0.976 \times V^{\frac{1}{12}}$
13	$\frac{128\pi^6}{135135} R^{13} \approx 0.911 \times R^{13}$	$\left(\frac{135135V}{128\pi^6}\right)^{\frac{1}{13}} \approx 1.007 \times V^{\frac{1}{13}}$
14	$\frac{\pi^7}{5040} R^{14} \approx 0.599 \times R^{14}$	$\frac{(5040V)^{\frac{1}{14}}}{\sqrt{\pi}} \approx 1.037 \times V^{\frac{1}{14}}$
15	$\frac{256\pi^7}{2027025} R^{15} \approx 0.381 \times R^{15}$	$\left(\frac{2027025V}{256\pi^7}\right)^{\frac{1}{15}} \approx 1.066 \times V^{\frac{1}{15}}$
n	$V_n(R)$	$R_n(V)$

K-NN Estimator



Thank you!

Appendix: Kernel Estimator

Optional

- To get a smooth estimate, we use a smooth weight function, *kernel function*, e.g., the Gaussian kernel
- For the univariate case, i.e., each data point is 1-dimensional, the Gaussian kernel is defined as

$$k(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right)$$

- The Kernel Estimator is computed via

$$\hat{P}(x) = \frac{1}{N\Delta} \sum_{i=1}^{N} k\left(\frac{x_i - x}{\Delta}\right)$$

Kernel Estimator (cont.)

Optional

- For the multivariate case, i.e., each data point is d-dimensional, the Gaussian kernel is defined as

$$k(\boldsymbol{u}) = \frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{\|\boldsymbol{u}\|_2^2}{2}\right)$$

- The Kernel Estimator is computed via

$$\hat{P}(\boldsymbol{x}) = \frac{1}{NV} \sum_{i=1}^{N} k\left(\frac{\boldsymbol{x}_i - \boldsymbol{x}}{h}\right)$$

Kernel Estimator

Optional