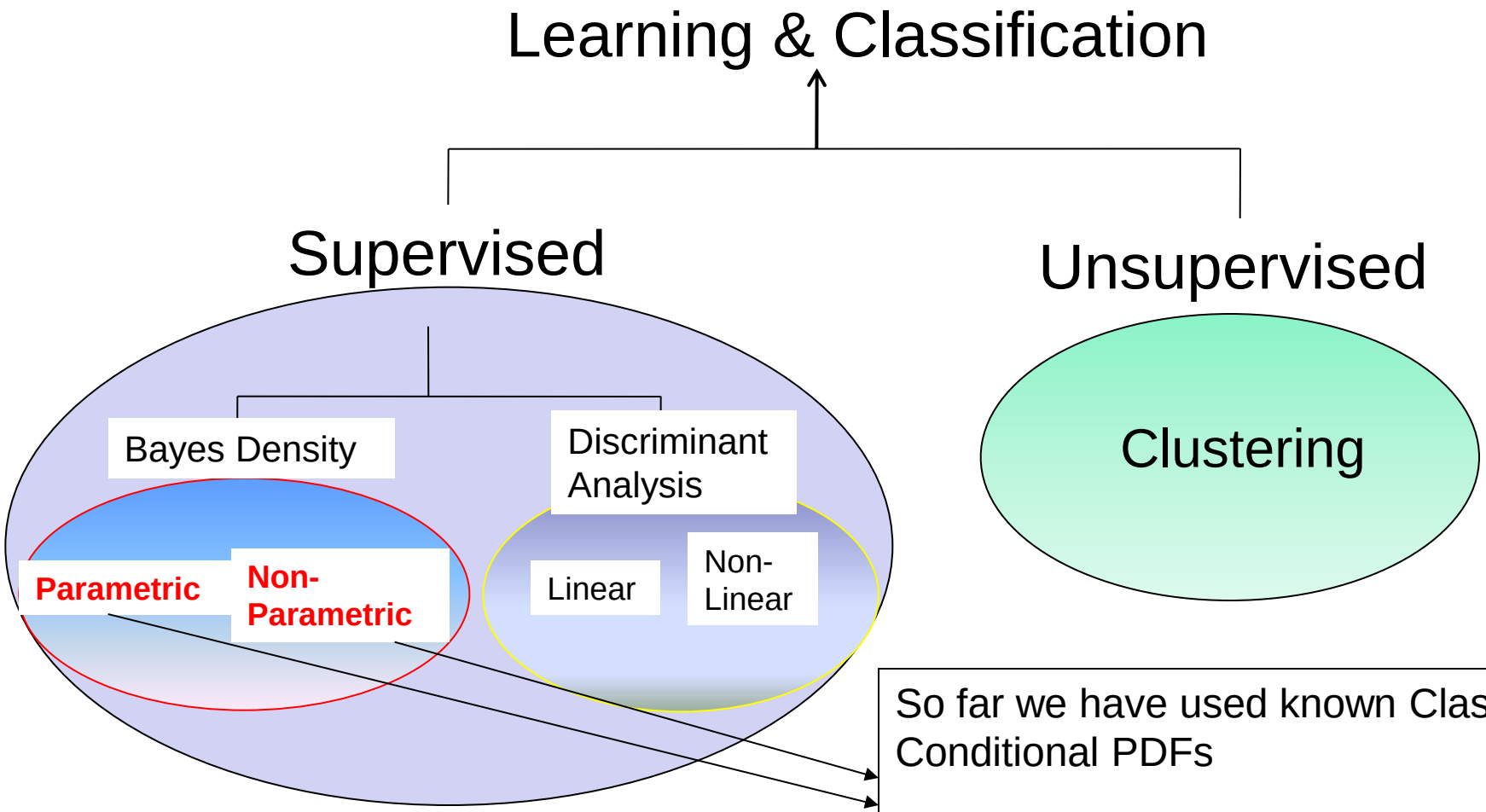
Prof. Marios Savvides

Pattern Recognition Theory

Lecture 7: Parametric & Non Parametric Density Estimation

Electrical & Computer ENGINEERING

Overview Of Pattern Recognition



So far we have used known Class Conditional PDFs

This lecture will focus on how to determine PDFs using Parametric & Non-Parametric Forms

Big Picture

Estimation Parameters

(MLE, MAPE)

$P(D \mid \boldsymbol{\theta})$

$P(\boldsymbol{\theta} \mid D)$

→

Probability

Density

Bayesian Estimation

$P(\mathbf{x}|D)$

→

Class Conditional PDFs

Class priors

$P(\mathbf{x}|\omega_1)$

$P(\omega_1)$

→

Bayes

Classifier

Minimum Error Classifier

Overview

- Previous lectures have shown how to develop classifiers when the underlying statistical structure is known

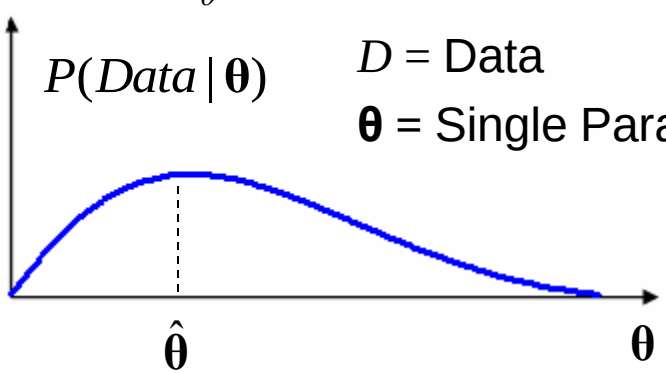
- Parametric Estimation
  - This method assumes a **particular form** of a PDF (e.g. Gaussian) is known so that we only need to determine the **parameters** (e.g. Mean & Variance)
    - Maximum Likelihood Estimation (MLE)
    - Maximum A Posteriori (Bayesian) Estimation (MAPE)

- Non-Parametric Density Estimation
  - This method **does not assume ANY knowledge** about the density
    - Parzen Windows
    - Kernel Density Estimation
    - K-Nearest Neighbor Rule

ML Estimation (MLE)

- **Maximum Likelihood Estimation**
  - Assume $P(\mathbf{x}|\omega)$ has a known parametric form uniquely determined by the parameter vector $\boldsymbol{\theta}$
  - The parameters are assumed to be **FIXED (i.e. NON RANDOM)** but unknown
  - Suppose we have a dataset D with the samples in D having been drawn **independently** according to the probability law $P(\mathbf{x}|\omega)$
  - The MLE is the value of $\boldsymbol{\theta}$ that best explains the data and **once we know this value, we know** $P(\mathbf{x}|\omega)$

$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} \{P(D|\boldsymbol{\theta})\}$$



D = Data

$\boldsymbol{\theta}$ = Single Parameter/Vector of Parameters

"Choose the value of θ that is the most likely to give rise to the data we observe"

MLE

$$D = \{x_1, x_2, \ldots, x_N\}$$ **N independent observations**

$$P(D \mid \boldsymbol{\theta}) = P(x_1, x_2, \ldots, x_N \mid \boldsymbol{\theta}) = \prod_{k=1}^{N} P(x_k \mid \boldsymbol{\theta})$$

The likelihood of observing a particular pattern (random variable)

$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} \{P(Data \mid \boldsymbol{\theta})\}$$

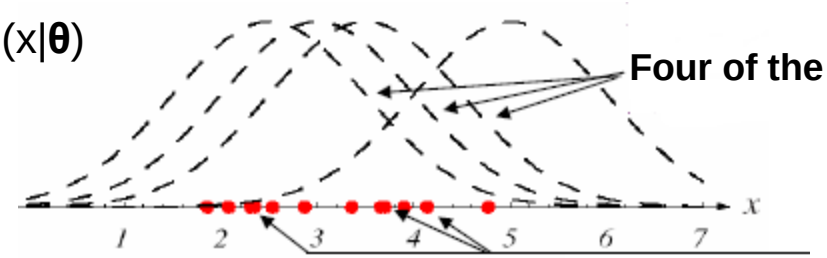
"Choose the value of θ that is the most likely to give rise to the data we observe"

MLE contd..

- It is convenient to work with the log of the likelihood

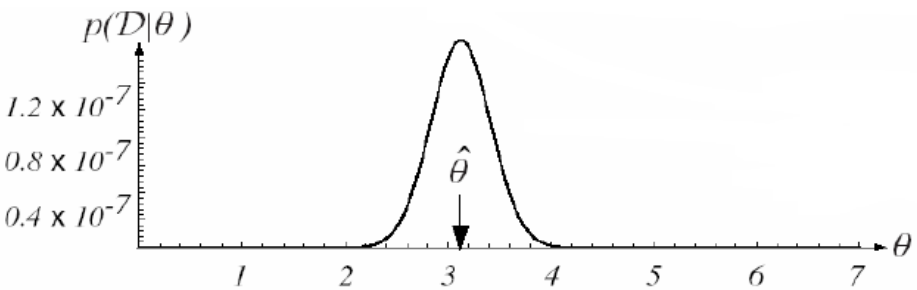
$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} \{P(D|\boldsymbol{\theta})\} = \arg\max_{\theta} \left\{\log\left(P\left(D|\boldsymbol{\theta}\right)\right)\right\}$$

P(x|**θ**)



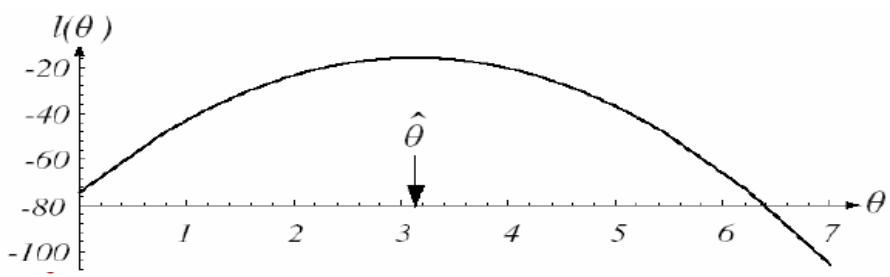
Four of the infinite number of candidate source distributions

Several training 1D points drawn from a Gaussian of a particular variance but unknown mean



The likelihood *P*(Data|**θ**) **as a function of the mean**

If we had a very large number of training points this likelihood would be very narrow



The log of the likelihood (l(θ)) is maximized at the same theta that maximizes the likelihood since log is a monotonically increasing function

How To Solve For The ML Estimate?

- Let $\boldsymbol{\theta}$ be the p-component parameter vector $\boldsymbol{\theta} = \left[\theta_1, \theta_2, \ldots, \theta_p\right]^T$
- Let this be the gradient operator $\nabla_{\boldsymbol{\theta}} = \left[\frac{\partial}{\partial \theta_1}, \frac{\partial}{\partial \theta_2}, \ldots, \frac{\partial}{\partial \theta_p}\right]^T$
- We have $P(D \mid \boldsymbol{\theta}) = \prod_{k=1}^{n} P\left(x_k \mid \boldsymbol{\theta}\right)$
- We define *l(**θ**)* the log-likelihood of the function

$$l\left(\boldsymbol{\theta}\right) = \log\left(P(D \mid \boldsymbol{\theta})\right) = \sum_{k=1}^{n} \log\left(P\left(x_k \mid \boldsymbol{\theta}\right)\right)$$

- And

$$\nabla_{\boldsymbol{\theta}} l\left(\boldsymbol{\theta}\right) = \nabla_{\boldsymbol{\theta}} \log\left(P(D \mid \boldsymbol{\theta})\right) = \sum_{k=1}^{n} \nabla_{\boldsymbol{\theta}} \log\left(P\left(x_k \mid \boldsymbol{\theta}\right)\right)$$

- A set of necessary condition for the ML estimate can be obtained from the set of *p* equations:

$$\boxed{\nabla_{\boldsymbol{\theta}} l\left(\boldsymbol{\theta}\right) = \mathbf{0}}$$

MLE Example: Univariate Gaussian

- Now assume **neither the mean nor the covariance** matrix are known
- First consider univariate case:

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 = \mu \\ \theta_2 = \sigma^2 \end{bmatrix} \quad P(D \mid \boldsymbol{\theta}) = \prod_{k=1}^{n} P\left(x_k \mid \boldsymbol{\theta}\right) \quad \log P(x_k \mid \boldsymbol{\theta}) = -\frac{1}{2}\log\left(2\pi\theta_2\right) - \frac{1}{2\theta_2}\left(x_k - \theta_1\right)^2$$

- Its derivative is:

$$\nabla_{\boldsymbol{\theta}} l = \nabla_{\boldsymbol{\theta}} \log\left(P(\mathrm{x}_k \mid \boldsymbol{\theta})\right) = \begin{bmatrix} \frac{1}{\theta_2}\left(x_k - \theta_1\right) \\ -\frac{1}{2\theta_2} + \frac{\left(x_k - \theta_1\right)^2}{2\theta_2^{\,2}} \end{bmatrix} = 0$$

- Setting it to zero leads to:

$$\sum_{k=1}^{n} \frac{1}{\theta_2}\left(x_k - \theta_1\right) = 0 \quad \textbf{and} \quad -\sum_{k=1}^{n} \frac{1}{\theta_2} + \sum_{k=1}^{n} \frac{(x_k - \theta_1)^2}{\theta_2^2} = 0$$

- Rearranging:

$$\boxed{\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k} \qquad \boxed{\hat{\sigma} = \frac{1}{n}\sum_{k=1}^{n} (x_k - \hat{\mu})^2}$$

ML Estimate

Sample Mean **Sample Variance**

MLE Example: Multivariate Gaussian

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

$$l(\theta) = \log(P(D\,|\,\boldsymbol{\theta})) = \sum_{k=1}^{n} \log(P(\mathbf{x}_k\,|\,\boldsymbol{\theta}))$$

Consider **only the mean is unknown**:

$$\log P(\mathbf{x_k}\,|\,\boldsymbol{\mu}) = -\frac{1}{2}\log\left((2\pi)^d\,|\boldsymbol{\Sigma}|\right) - \frac{1}{2}(\mathbf{x_k}-\boldsymbol{\mu})^T\,\boldsymbol{\Sigma}^{-1}(\mathbf{x_k}-\boldsymbol{\mu})$$

Derivative of log likelihood must be set to 0 to obtain the MLE

$$\nabla_{\boldsymbol{\mu}} \log(P(D\,|\,\boldsymbol{\mu})) = \sum_{k=1}^{n} \boldsymbol{\Sigma}^{-1}(\mathbf{x_k}-\boldsymbol{\mu}) = \mathbf{0}$$

The ML estimate must satisfy:

$$\boxed{\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x_k}}$$

Sample Mean -> ML Estimate

MLE Example: Multivariate Gaussian

- **Neither the mean nor the covariance** matrix are known

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 = \boldsymbol{\mu} \\ \theta_2 = \boldsymbol{\Sigma} \end{bmatrix} \qquad \log P(\mathbf{x_k} \mid \boldsymbol{\theta}) = -\frac{1}{2}\log\left((2\pi)^d |\boldsymbol{\Sigma}|\right) - \frac{1}{2}(\mathbf{x_k} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x_k} - \boldsymbol{\mu})$$

- Derivative of log likelihood is:

$$\nabla_{\boldsymbol{\theta}} l = \nabla_{\boldsymbol{\theta}} \log\left(P(\mathrm{x_k} \mid \boldsymbol{\theta})\right) = \begin{bmatrix} \boldsymbol{\Sigma}^{-1}(\mathbf{x_k} - \boldsymbol{\mu}) \\ ? \end{bmatrix}$$

- How to take the gradient of a determinant of a matrix?

$$P(\mathbf{x} \mid \boldsymbol{\Sigma}) = \prod_{k=1}^{n} P(\mathbf{x_k} \mid \boldsymbol{\Sigma}) = \prod_{k=1}^{n} \left\{ \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x_k} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x_k} - \boldsymbol{\mu}) \right) \right\}$$

$$= \frac{1}{\left[(2\pi)^d |\boldsymbol{\Sigma}|\right]^{n/2}} \exp\left(-\frac{1}{2}\sum_{k=1}^{n} (\mathbf{x_k} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x_k} - \boldsymbol{\mu}) \right)$$

Need to take gradient with respect to Σ

MLE Example: Multivariate Gaussian

$$P(\mathbf{x}\,|\,\boldsymbol{\Sigma}) = \prod_{k=1}^{n} P(\mathbf{x_k}\,|\,\boldsymbol{\Sigma}) = \prod_{k=1}^{n}\left\{\frac{1}{\sqrt{(2\pi)^d\,|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\mathbf{x_k}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x_k}-\boldsymbol{\mu})\right)\right\}$$

$$= \frac{1}{\left[(2\pi)^d\,|\boldsymbol{\Sigma}|\right]^{n/2}}\exp\left(-\frac{1}{2}\sum_{k=1}^{n}(\mathbf{x_k}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x_k}-\boldsymbol{\mu})\right)$$

$$\sum_{k=1}^{n}(\mathbf{x_k}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x_k}-\boldsymbol{\mu})$$

$$= (\mathbf{x}_1-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_1-\boldsymbol{\mu}) + \ldots + (\mathbf{x}_N-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n-\boldsymbol{\mu})$$

$$= trace\left(\boldsymbol{\Sigma}^{-1}(\mathbf{x}_1-\boldsymbol{\mu})(\mathbf{x}_1-\boldsymbol{\mu})^T\right) + \ldots + trace\left(\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n-\boldsymbol{\mu})(\mathbf{x}_n-\boldsymbol{\mu})^T\right)$$

$$= trace\left(\boldsymbol{\Sigma}^{-1}(\mathbf{x}_1-\boldsymbol{\mu})(\mathbf{x}_1-\boldsymbol{\mu})^T + \ldots + \boldsymbol{\Sigma}^{-1}(\mathbf{x}_n-\boldsymbol{\mu})(\mathbf{x}_n-\boldsymbol{\mu})^T\right)$$

$$= trace\left(\boldsymbol{\Sigma}^{-1}\sum_{k=1}^{n}(\mathbf{x}_k-\boldsymbol{\mu})(\mathbf{x}_k-\boldsymbol{\mu})^T\right)$$

$$\mathbf{A} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k-\boldsymbol{\mu})(\mathbf{x}_k-\boldsymbol{\mu})^T$$

Scalar

$$\mathbf{b^T B b} = trace(\mathbf{b^T B b}) = trace(\mathbf{B b b^T})$$

$$trace(\mathrm{A}+\mathrm{B}) = trace(\mathrm{A}) + trace(\mathrm{B})$$

$$trace(\mathbf{C(A+B)}) = trace(\mathbf{CA}) + trace(\mathbf{CB})$$

MLE Example: Multivariate Gaussian

$$\mathbf{A} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^T$$

We can now rewrite:

$$\sum_{k=1}^{n}(\mathbf{x_k} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x_k} - \boldsymbol{\mu}) = trace\left(\boldsymbol{\Sigma}^{-1}\sum_{k=1}^{n}(\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^T\right) = trace\left(n\boldsymbol{\Sigma}^{-1}\mathbf{A}\right) = n.trace\left(\boldsymbol{\Sigma}^{-1}\mathbf{A}\right)$$

$$p(\mathbf{x}\,|\,\boldsymbol{\Sigma}) = \prod_{k=1}^{n} p(\mathbf{x_k}\,|\,\boldsymbol{\Sigma}) = \frac{1}{[(2\pi)^d\,|\,\boldsymbol{\Sigma}\,|]^{n/2}}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

$$= \frac{1}{[(2\pi)^{dn/2}}\,|\,\boldsymbol{\Sigma}\,|]^{-n/2}\exp\left[-\frac{n}{2}trace(\boldsymbol{\Sigma}^{-1}\mathbf{A})\right]$$

- Now define **$\mathbf{B} = \boldsymbol{\Sigma}^{-1}\mathbf{A}$** and let $\lambda_1, \lambda_2, \ldots, \lambda_d$ be its eigenvalues

$$p(\mathbf{x}\,|\,\boldsymbol{\Sigma}) = \frac{1}{[(2\pi)^{dn/2}}\,|\,\boldsymbol{\Sigma}\,|]^{-n/2}\exp\left[-\frac{n}{2}trace(\mathbf{B})\right]$$

$$= \frac{1}{(2\pi)^{dn/2}}\left[\frac{|\,\mathbf{B}\,|}{|\,\mathbf{A}\,|}\right]^{n/2}\exp\left[-\frac{n}{2}trace(\mathbf{B})\right]$$

$$= \frac{1}{(2\pi)^{dn/2}}\,|\,\mathbf{A}\,|^{-n/2}\left(\prod_{i=1}^{d}\lambda_i\right)^{n/2}\exp\left[-\frac{n}{2}\sum_{i=1}^{d}\lambda_i\right]$$

$$\det(\mathrm{AB}) = \det(\mathrm{A})\det(\mathrm{B})$$

$$\det(\mathrm{A}^{-1}) = \det(\mathrm{A})^{-1}$$

$trace(\mathbf{A})$ **Sum of eigenvalues**

$\det(\mathbf{A})$ **Product of eigenvalues**

MLE Example: Multivariate Gaussian

Now consider the log likelihood

$$P(\mathbf{x}\,|\,\boldsymbol{\Sigma}) = (2\pi)^{-dn/2}\,|\mathbf{A}|^{-n/2}\left(\prod_{i=1}^{d}\lambda_i\right)^{n/2}\exp\left(-\frac{n}{2}\sum_{i=1}^{d}\lambda_i\right)$$

$$\log P(\mathbf{x}\,|\,\boldsymbol{\Sigma}) = -\frac{dn}{2}\log(2\pi) - \frac{n}{2}\log|\mathbf{A}| + \frac{n}{2}\sum_{i=1}^{d}\log(\lambda_i) - \frac{n}{2}\sum_{i=1}^{d}\lambda_i$$

Remember that **A** is fixed, and taking the gradient with respect to **Σ** is now equivalent to taking the derivatives with respect to the eigenvalues of **B**

$$\frac{\partial}{\partial\lambda_i}\log P(\mathbf{x}\,|\,\boldsymbol{\Sigma}) = \frac{n}{2\lambda_i} - \frac{n}{2} = 0 \quad \Rightarrow \lambda_i = 1 \text{ for i = 1,...,d}$$

Thus B must have all eigenvalues of 1 and is equivalent to I. This means that the likelihood is maximized if $\boldsymbol{\Sigma}^{-1}\mathbf{A}=\mathbf{I}$ or $\boldsymbol{\Sigma}=\mathbf{A}$

$$\boxed{\hat{\boldsymbol{\Sigma}}_{\mathbf{ML}} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^T}$$

MLE - Sample Covariance

MLE vs The Bayesian Approach

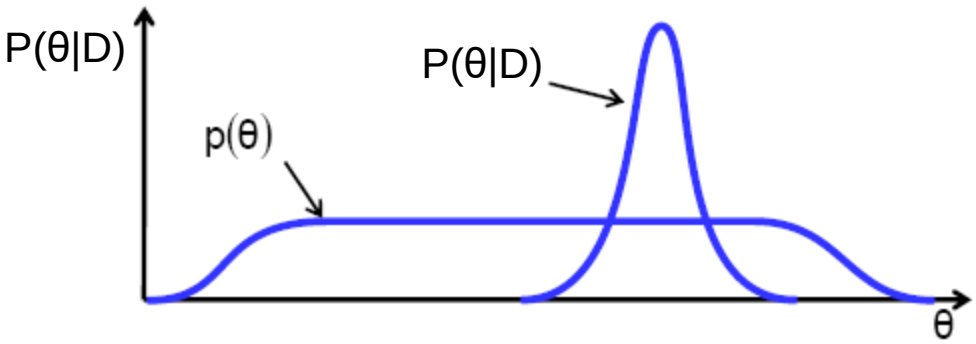
- **Maximum Likelihood Estimation (MLE)**
  - The parameters are assumed to be **FIXED (i.e. NON RANDOM)** but unknown
  - The ML seeks the solutions that best explains the data

$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} \{P(Data \mid \boldsymbol{\theta})\}$$

- **Bayesian Estimation (BE)**
  - The parameters are assumed to be **RANDOM VARIABLES** with some known **PRIORI DISTRIBUTION**
  - Bayesian approach aims at estimating the posterior density $P(\boldsymbol{\theta}|Data)$
  - The MAPE (Maximum A Posteriori Estimate) of $\boldsymbol{\theta}$ is the value of $\boldsymbol{\theta}$ that maximizes the posterior density (i.e. it is the mode of the posterior)

Bayesian Estimation

- In the Bayesian approach, our uncertainty about the parameters is represented by a PDF
- The parameters are described by a prior density $P(\boldsymbol{\theta})$ which indicates which parameters are more likely than others
- We make use of Bayes theorem to find the posterior $P(\boldsymbol{\theta}|D)$
- Ideally, we want the training data to "sharpen" the posterior $P(\boldsymbol{\theta}|D)$ or reduce our uncertainty about the parameters



Bayesian Estimation

- We ideally want to estimate a PDF - $P(x)$. Best we can do is estimate it by observing the training data to obtain $P(x|D)$. We also assume that is has a known parametric form. So $P(x|\boldsymbol{\theta})$ is completely known. But $\boldsymbol{\theta}$ is random (unlike in MLE) and has its own PDF.

$$P(x \mid D) = \int P(x, \boldsymbol{\theta} \mid D)\, d\theta$$

Using the theorem of total probability

$$= \int P(x \mid \boldsymbol{\theta}) P(\boldsymbol{\theta} \mid D)\, d\theta$$

This integration is typically very hard

Known

Unknown

$\boldsymbol{\theta}$ is random and has its own PDF

Applying Bayes rule:

$$P(\boldsymbol{\theta} \mid D) = \frac{P(D \mid \boldsymbol{\theta}) P(\boldsymbol{\theta})}{P(D)} = \frac{P(D \mid \boldsymbol{\theta}) P(\boldsymbol{\theta})}{\int P(D \mid \boldsymbol{\theta}) P(\boldsymbol{\theta})\, d\boldsymbol{\theta}}$$

Posterior

$$P(D \mid \boldsymbol{\theta}) = \prod_{k=1}^{N} P\left(x_k \mid \boldsymbol{\theta}\right)$$

http://people.stat.sc.edu/Hitchcock/stat535slidesday18.pdf

MAP Estimation (MAPE)

- MLE aims at determining the value $\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathbf{MLE}}$ that maximizes the likelihood

$$\boldsymbol{\theta}_{\text{MLE}} = \arg\max_{\boldsymbol{\theta}} \{P(D \mid \boldsymbol{\theta})\}$$

- MAPE assumes that $\boldsymbol{\theta}$ is not a fixed, but is an RV with a PDF given by $P(\boldsymbol{\theta})$ and it aims at maximizing the posterior

$$\boldsymbol{\theta}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \{P(\boldsymbol{\theta} \mid D)\}$$

where $P(\boldsymbol{\theta} \mid D) = \dfrac{P(D \mid \boldsymbol{\theta})P(\boldsymbol{\theta})}{P(D)} = \dfrac{P(D \mid \boldsymbol{\theta})P(\boldsymbol{\theta})}{\int P(D \mid \boldsymbol{\theta})P(\boldsymbol{\theta})d\boldsymbol{\theta}} = \dfrac{P(\boldsymbol{\theta})\prod_{k=1}^{N} P\left(x_k \mid \boldsymbol{\theta}\right)}{\int P(D \mid \boldsymbol{\theta})P(\boldsymbol{\theta})d\boldsymbol{\theta}}$

Since $P(D)$ is constant we can also view MAP estimate as

$$\boldsymbol{\theta}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \{P(D \mid \boldsymbol{\theta})P(\boldsymbol{\theta})\}$$

- Thus the MAPE of $\boldsymbol{\theta}$ is simply the mode of the posterior $P(\boldsymbol{\theta}|D)$ and MAPE differs from MLE as it determines a value of $\boldsymbol{\theta}$ which maximizes the posterior instead of the likelihood

MAPE Example: Univariate Gaussian

- Compute $P(\boldsymbol{\theta}|D)$ and the desired PDF $P(x|D)$ where $p(\mu) = N(\mu_0, \sigma_0^2)$
- σ (the uncertainty about μ) is known and fixed.
- Assume the known prior knowledge about the mean can be expressed by a *known* prior density assumed to be normal:

$$p(\mu) = N(\mu_0, \sigma_0^2)$$ Known!

$$P(\mu \mid D) = \frac{P(D \mid \mu)P(\mu)}{\int P(D \mid \mu)P(\mu)d\mu} = \alpha \prod_{k=1}^{n} P(x_k \mid \mu)P(\mu)$$

$$P(\mu \mid D) = \alpha \prod_{k=1}^{n} \overbrace{\frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x_k - \mu}{\sigma}\right)^2\right]}^{P(x_k|\mu)} \overbrace{\frac{1}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{1}{2}\left(\frac{\mu - \mu_0}{\sigma_0}\right)^2\right]}^{P(\mu)}$$

$$= \alpha' \exp\left[-\frac{1}{2}\left(\sum_{k=1}^{n}\left(\frac{x_k - \mu}{\sigma}\right)^2 + \left(\frac{\mu - \mu_0}{\sigma_0}\right)^2\right)\right]$$

$$= \alpha'' \exp\left[-\frac{1}{2}\left[\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}\right)\mu^2 - 2\left(\frac{1}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0^2}\right)\mu\right]\right]$$ GAUSSIAN!

MAPE Example: Univariate Gaussian

Gaussian Likelihood && Gaussian Prior ➔ Gaussian Posterior

$$P(\mu \mid D) = \alpha'' \exp\left[-\frac{1}{2}\left[\underbrace{\left(\frac{n}{\sigma^2}+\frac{1}{\sigma_0{}^2}\right)}\mu^2 - 2\underbrace{\left(\frac{1}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0{}^2}\right)}\mu\right]\right]$$

compare

$$P(\mu \mid D) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right] = \alpha \exp\left[-\frac{1}{2}\left[\mu^2\overbrace{\left(\frac{1}{\sigma_n{}^2}\right)} - 2\mu\left(\frac{\mu_n}{\sigma_n^2}\right) + \left(\frac{\mu_n}{\sigma_n}\right)^2\right]\right]$$

We get $\frac{1}{\sigma_n{}^2} = \frac{n}{\sigma^2} + \frac{1}{\sigma_0{}^2}$ and $\frac{\mu_n}{\sigma_n^2} = \frac{1}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0{}^2} = \frac{n}{\sigma^2}\hat{\mu}_n + \frac{\mu_0}{\sigma_0{}^2}$

where $\hat{\mu}_n = \frac{1}{n}\sum_{k=1}^{n} x_k$ and is the sample mean

So $\boxed{\sigma_n{}^2 = \frac{\sigma^2\sigma_0{}^2}{n\sigma_0{}^2+\sigma^2}}$ $\boxed{\mu_n = \left(\frac{n\sigma_0{}^2}{n\sigma_0{}^2+\sigma^2}\right)\hat{\mu}_n + \left(\frac{\sigma^2}{n\sigma_0{}^2+\sigma^2}\right)\mu_0}$

MAPE Example: Univariate Gaussian

$$\mu_n = \left(\frac{n\sigma_0^{\ 2}}{n\sigma_0^{\ 2}+\sigma^2}\right)\hat{\mu}_n + \left(\frac{\sigma^2}{n\sigma_0^{\ 2}+\sigma^2}\right)\mu_0$$

Best guess for μ

$$\sigma_n^{\ 2} = \frac{\sigma^2\sigma_0^{\ 2}}{n\sigma_0^{\ 2}+\sigma^2}$$

Uncertainty for μ

- μ_n is a linear combination of $\hat{\mu}_n$ and μ_0 (always lies between them)
  - For small samples -> More weights on prior ($\hat{\mu}_0$)
  - For large samples -> More weights on observation ($\hat{\mu}_n$)
- If $\sigma_n \neq 0$, then $\mu_n \rightarrow \hat{\mu}_n$ as $n \rightarrow \infty$ (MLE = MAPE)
- If $\sigma_0 = 0$, then $\mu_n = \mu_0$
- If $\sigma_0 >> \sigma$ then $\mu_n = \hat{\mu}_n$ (MLE = MAPE)

MAPE Example: Univariate Gaussian

- μ_n represents our best guess for μ after observing n samples, and σ_n^2 measures our uncertainty about this guess.

$$\mu_n = \underbrace{\left(\frac{n\sigma_0^2}{n\sigma_0^2+\sigma^2}\right)\hat{\mu}_n}_{ML} + \underbrace{\left(\frac{\sigma^2}{n\sigma_0^2+\sigma^2}\right)\mu_0}_{PRIOR\_KNOWLEDGE}$$

$$\sigma_n^2 = \frac{\sigma^2\sigma_0^2}{n\sigma_0^2+\sigma^2}$$

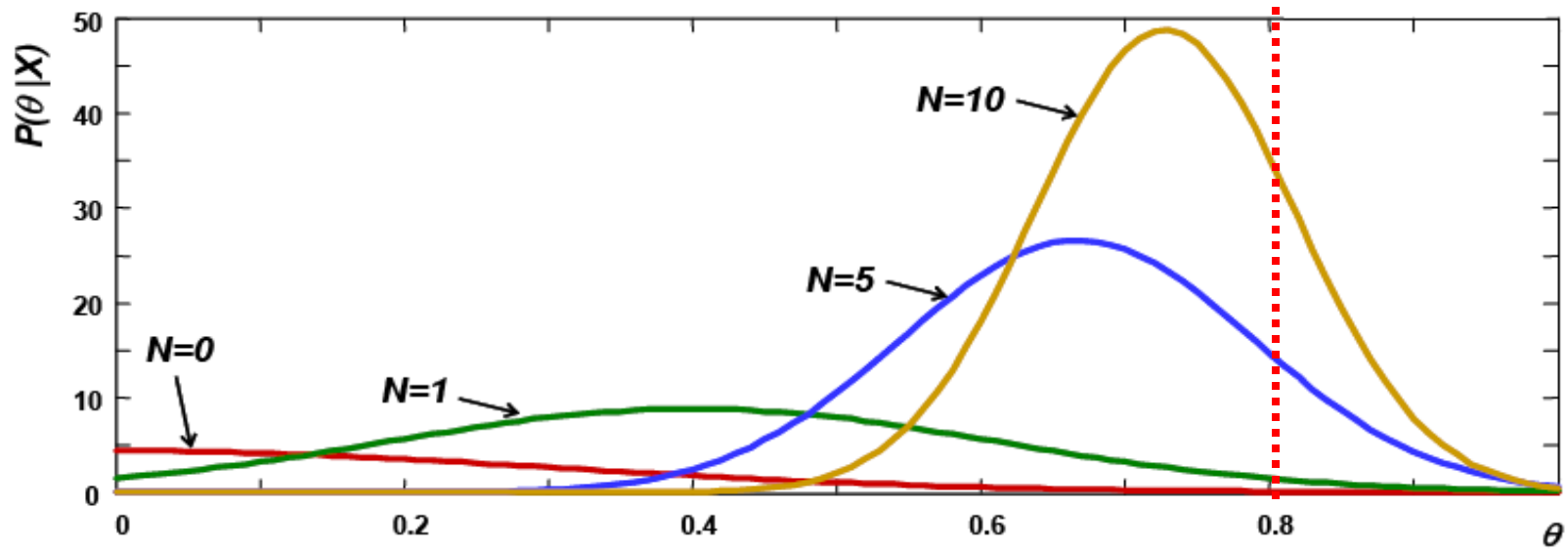
Decreases as $n \to \infty$

- Each additional observation decreases our uncertainty and $P(\mu|D)$ becomes narrower and sharply peaked around the true value of μ and becomes a Dirac delta function. This is called ***Bayesian Learning.***

- A class of PDF $P(\boldsymbol{\theta})$ is said to be **conjugate** to a class of likelihood functions $P(x|\boldsymbol{\theta})$ if the resulting posterior distributions $P(\boldsymbol{\theta}|x)$ are in the same family as $P(\boldsymbol{\theta})$.
  - The Gaussian family is conjugate to itself if the likelihood function is Gaussian, choosing a Gaussian prior will ensure that the posterior distribution is also Gaussian

MAPE Example: Gaussians

- Assume that the true mean of the $P(x)$ is N(0.8, 0.09).
  - In reality this is something we cannot know
- We generate a number of examples from this distribution
- We don't know where the mean will be, so we assume a Gaussian prior
  - $P_0(\mu)=N(0, 0.09)$
- As the number of training examples increases, the estimates μ_N approaches its true value of 0.8 and the spread decreases



MAPE Example: Multivariate Gaussian

- Compute $P(\boldsymbol{\mu}|D)$ and the desired PDF $P(\mathbf{x}|D)$ where $P(\mathbf{x}\,|\,\boldsymbol{\mu}) = N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- Assume the known prior knowledge about the mean can be expressed by a *known* prior density assumed to be normal:

$$P(\boldsymbol{\mu}) = N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

Known!

$$P(\boldsymbol{\mu}\,|\,D) = \frac{P(D\,|\,\boldsymbol{\mu})P(\boldsymbol{\mu})}{\int P(D\,|\,\boldsymbol{\mu})P(\boldsymbol{\mu})d\boldsymbol{\mu}} = \alpha \prod_{k=1}^{n} P(\mathbf{x}_\mathbf{k}\,|\,\boldsymbol{\mu})P(\boldsymbol{\mu})$$

$$P(\boldsymbol{\mu}\,|\,D) = \alpha \prod_{k=1}^{n} P(\mathbf{x}_\mathbf{k}\,|\,\boldsymbol{\mu})P(\boldsymbol{\mu})$$

$$= \alpha' \exp\left[-\frac{1}{2}\left(\boldsymbol{\mu}^T\left(n\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}_0^{-1}\right)\boldsymbol{\mu} - 2\boldsymbol{\mu}^T\left(\boldsymbol{\Sigma}^{-1}\sum_{k=1}^{n}\mathbf{x}_\mathbf{k} + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right)\right)\right]$$

Which has a Gaussian Form

$$= \alpha'' \exp\left[-\frac{1}{2}\left(\boldsymbol{\mu} - \boldsymbol{\mu}_n\right)^T \boldsymbol{\Sigma}_n^{-1}\left(\boldsymbol{\mu} - \boldsymbol{\mu}_n\right)\right]$$

MAPE Example: Multivariate Gaussian

- Thus $P(\boldsymbol{\mu}|D)$ is $N(\boldsymbol{\mu}_\mathbf{n}, \boldsymbol{\Sigma}_\mathbf{n})$. Equating coefficients, we obtain:

$$\boldsymbol{\Sigma}_n^{-1} = n\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}_0^{-1} \qquad \text{and} \qquad \boldsymbol{\Sigma}_n^{-1}\boldsymbol{\mu}_\mathbf{n} = n\boldsymbol{\Sigma}^{-1}\hat{\boldsymbol{\mu}}_\mathbf{n} + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0$$

where $\hat{\boldsymbol{\mu}}_\mathbf{n} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_\mathbf{k}$ is the sample mean

After some manipulation (which we don't prove) we get:

$$\boldsymbol{\mu}_\mathbf{n} = \boldsymbol{\Sigma}_0\left(\boldsymbol{\Sigma}_0 + \frac{1}{n}\boldsymbol{\Sigma}\right)^{-1}\hat{\boldsymbol{\mu}}_\mathbf{n} + \frac{1}{n}\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}_0 + \frac{1}{n}\boldsymbol{\Sigma}\right)^{-1}\boldsymbol{\mu}_0$$

$$\boldsymbol{\Sigma}_n = \boldsymbol{\Sigma}_0\left(\boldsymbol{\Sigma}_0 + \frac{1}{n}\boldsymbol{\Sigma}\right)^{-1}\frac{1}{n}\boldsymbol{\Sigma}$$

- Finally, it can be shown that

$$P(\mathbf{x} \mid D) = N(\boldsymbol{\mu}_\mathbf{n}, \boldsymbol{\Sigma} + \boldsymbol{\Sigma}_\mathbf{n})$$

MAPE Example: Gaussians

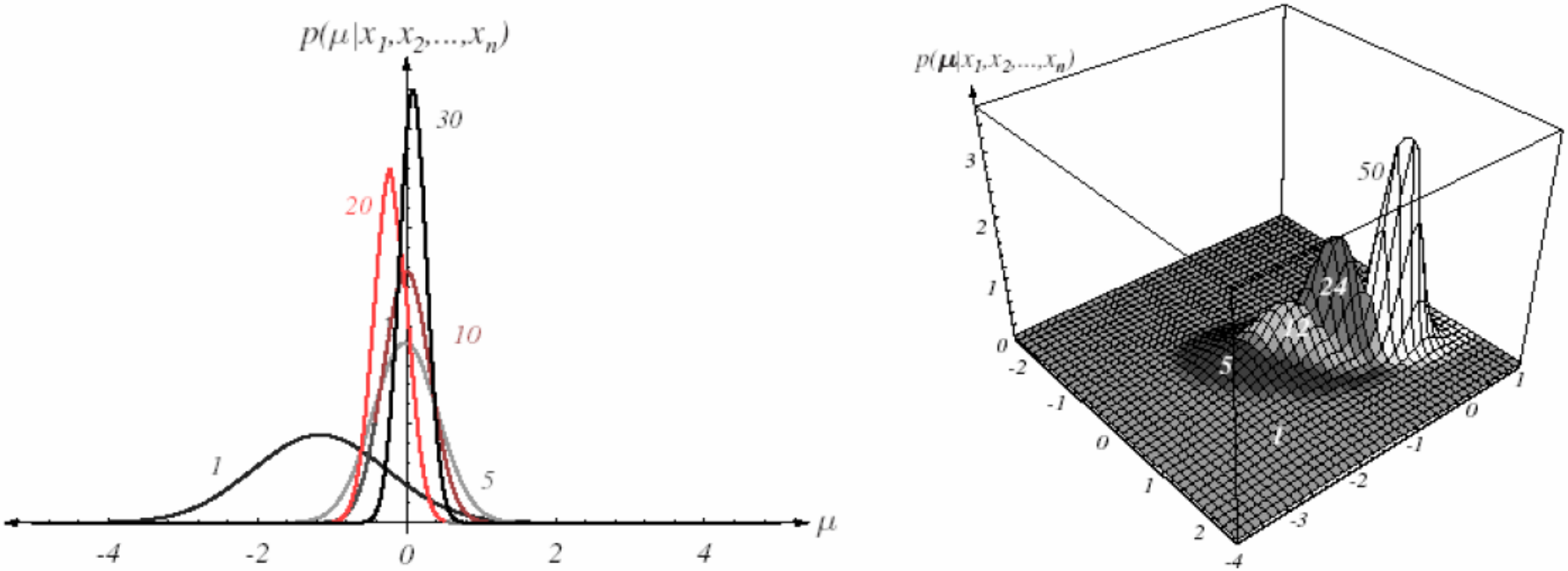


FIGURE 3.2. Bayesian learning of the mean of normal distributions in one and two dimensions. The posterior distribution estimates are labeled by the number of training samples used in the estimation. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Recap: ML vs Bayesian Estimation

- BE assumes that the parameters come from a distribution with known priors
- BE provides a distribution for $\boldsymbol{\theta}$ rather than point values. BE provides more information, but is in general much more difficult to compute (integration)
- For most times, if the assumptions are correct, MLE gives good enough results

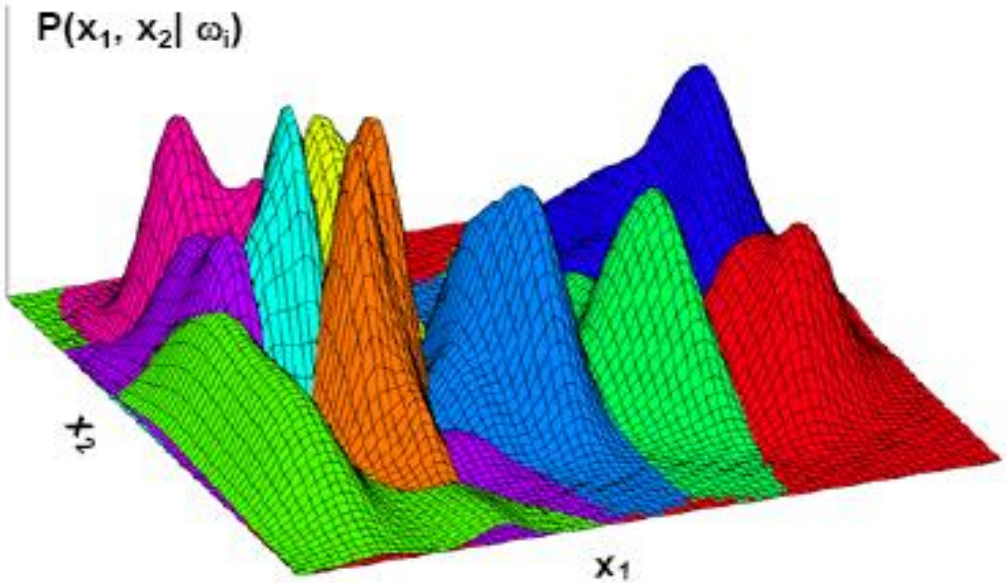
Recap: ML vs Bayesian Estimation

- Amount of Training data
  - The two methods are equivalent assuming infinite training samples.
  - They differ for smaller training data
- Computational Complexity
  - ML uses **differential calculus** or gradient search
  - Bayesian estimation needs complex multidimensional **integration techniques**
- Solution Complexity
  - ML solution is easy to interpret
  - A Bayes Estimation solution **might not be** of the parametric form assumed
- Prior distribution
  - If the prior $P(\boldsymbol{\theta})$ is **uniform (flat),** BE solutions are equivalent to ML solutions

Non Parametric Density Estimation

- Problems with parametric techniques
  - Most forms of distributions are unimodal
  - Most of the time we assumed that features are independent

- In non-parametric approach we don't need these assumtions
  - We have to attempt to estimate the density directly from the data without making any parametric assumptions about the true density

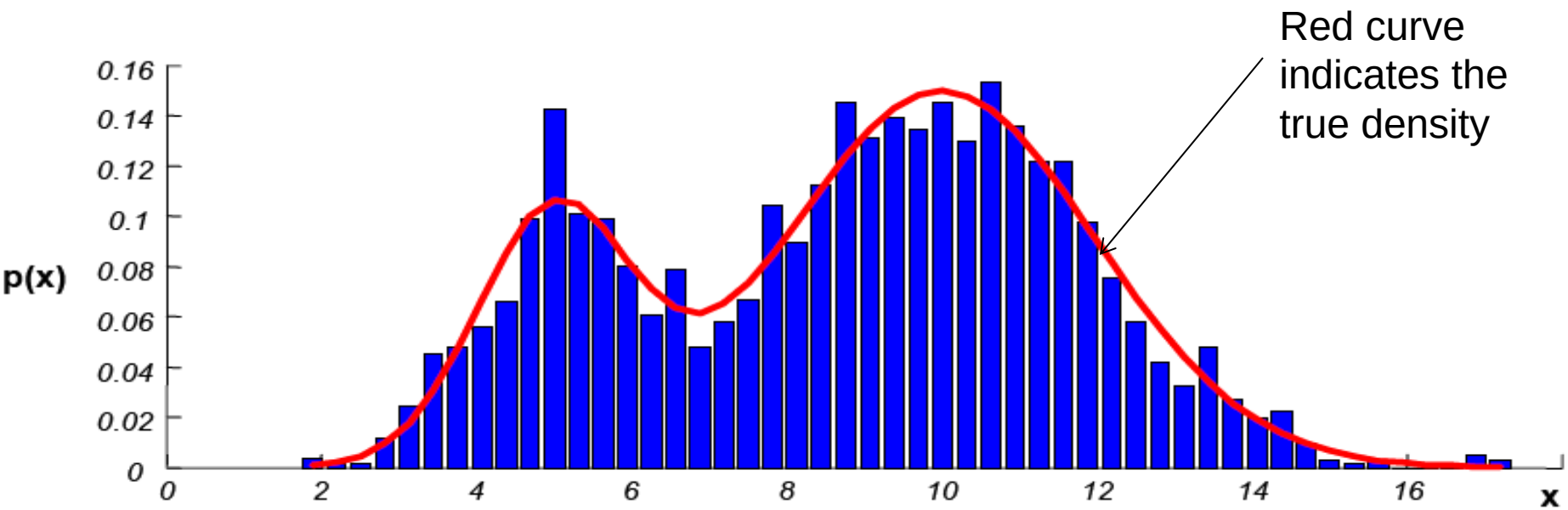


The Histogram

- The simplest form of non-parametric density estimation
  - Divide the sample space into a number of bins and approximate the density at the center of each bin using the fraction of training samples that fall into the corresponding bin.

$$P_H(x) = \frac{1}{n}\frac{\left[\text{number of samples in the same bin as } x\right]}{\left[\text{width of bin containing } x\right]}$$

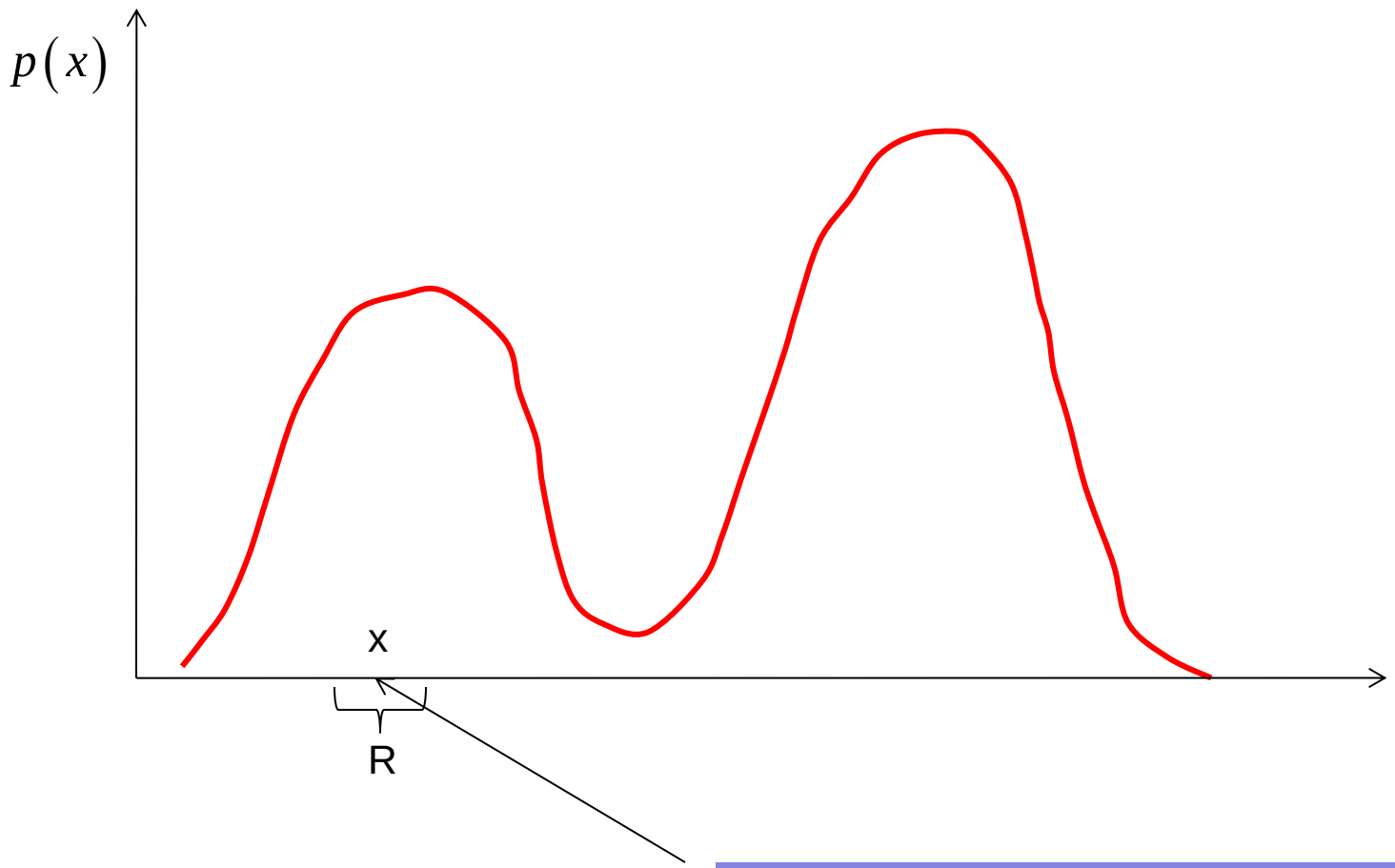
  - The histogram requires two tuning options to be defined:
    - Bin width
    - Starting position of the first bin



The Histogram

- The histogram is simple, but has flaws
  - The bin location can introduce discontinuity artifacts that are not due to the true underlying density
  - The histogram approach becomes impractical when the number of dimensions increases. The number of bins needed grows exponentially.
- Histograms are useful for visualizations in one or two dimensions

Non Parametric Density Estimation



Let's estimate distribution at this point x

Consider a region R around the point x → analogous to the bin in a histogram

Non Parametric Density Estimation

- From basic probability, we know that
  - The probability that a vector x, drawn from some distribution p(x), will fall in a given region R of the sample space is:

$$P = \int_R p(x')dx'$$

(i.e. a space averaged value)

  - Suppose now that *n* vectors $\{x_1, x_2, \ldots, x_n\}$ are drawn from the distribution. The probability that *k* of these *n* vectors fall in R is given by the binomial distribution

  - From the properties of the binomial pmf, we have that

$$P_k = \binom{n}{k} P^k (1-P)^{n-k}$$

$$E[k] = nP \qquad Var[k] = nP(1-P)$$

$$E[k/n] = P \qquad Var[k/n] = P(1-P)/n$$

Non Parametric Density Estimation

$$E\left[k/n\right] = P \qquad Var\left[k/n\right] = P\left(1-P\right)/n$$

- As $n \rightarrow \infty$, the distribution becomes sharper and hence a good estimate of the probability P can be obtained from the mean fraction of the points that fall within R

$$P = \int_R p(x')dx' \cong \frac{k}{n}$$

- If we now assume that R is so small that p(x) is almost constant within it, then

$$\int_R p(x')dx' \cong p(x)V$$

where V is the volume enclosed by the region R

For this one dimensional example, V will just be the length of the region

Non Parametric Density Estimation

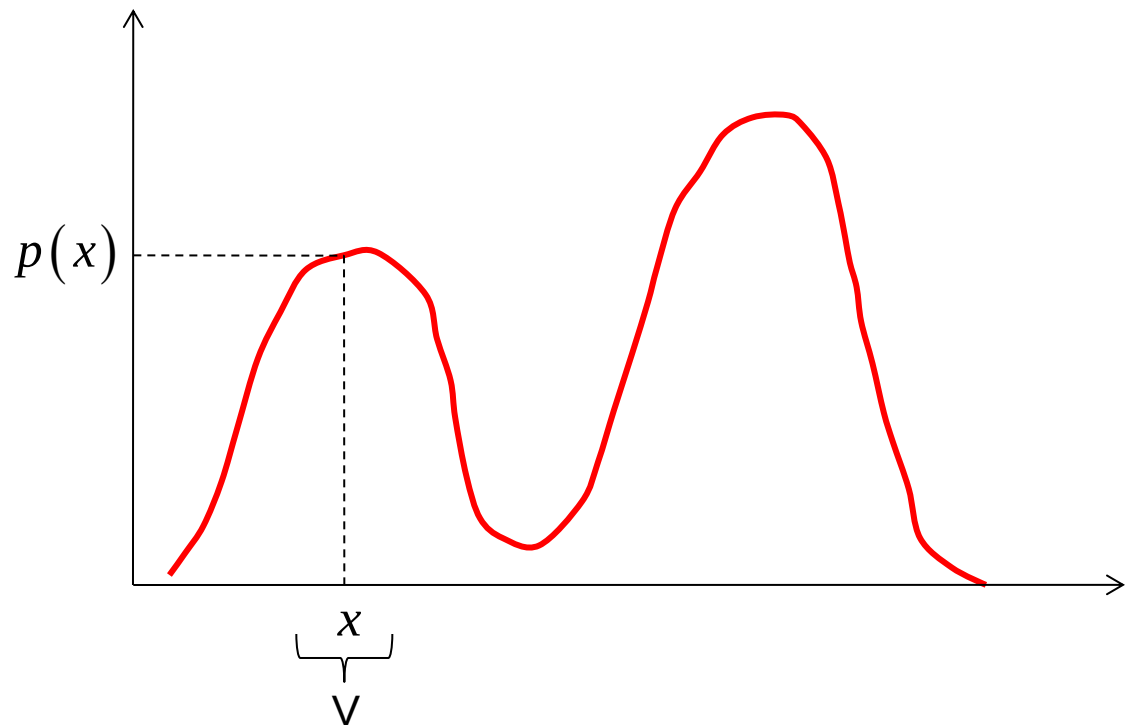
$$P = \int_R p(x')dx' \cong \frac{k}{n}$$

$$\int_R p(x')dx' \cong p(x)V$$

– Merging the two results we obtain:

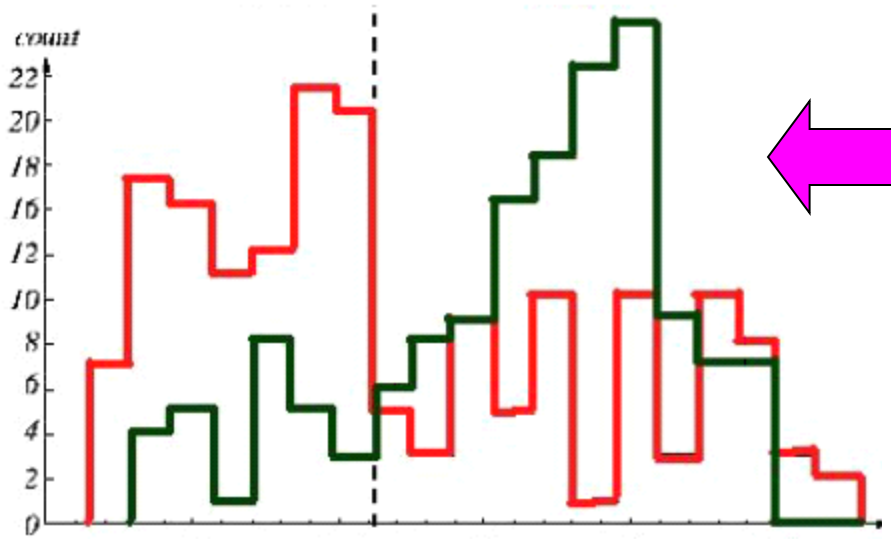
$$p(x) = \frac{k}{nV}$$

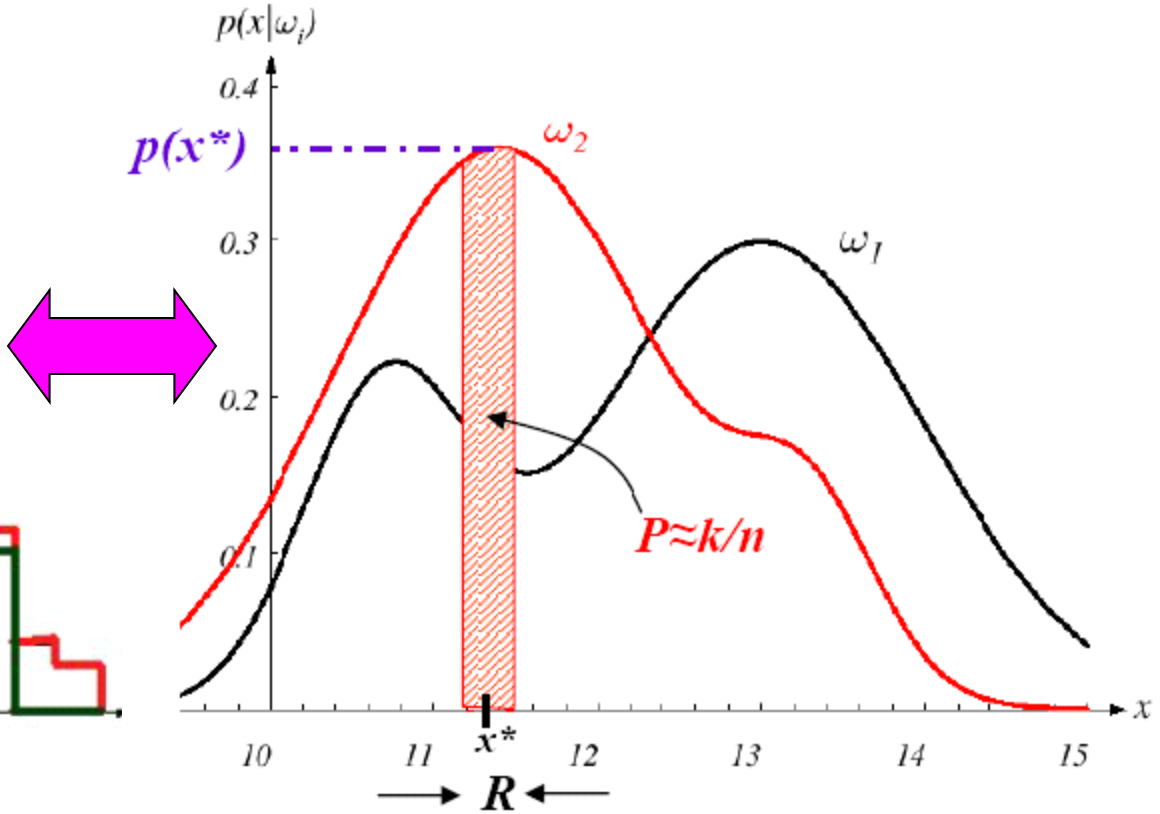
What we get is a space averaged value depending on V



Non Parametric Density Estimation

- Example:
  - If we observe a large number of fish, count those whose length fall within the range defined by R, *k/n* can be used as an estimate of P when $n \rightarrow \infty$





What are the Assumptions Here?

$$\boxed{p(x) \cong \frac{k}{nV}}$$

1. V is so small that p(x) is constant in R.

– But if V is really small, then it may not contain any samples! Then (k/n)=0

2. V is sufficiently large so that we get enough number of samples (k) such that

$$P \cong \frac{k}{n}$$

– Two contradictory assumptions! So we'll have to accept
  – A certain amount of variance in (k/n)
  – A space averaged value for p(x) (since V will not be very small)

Approaches

$$\boxed{p(x) \cong \frac{k}{nV}}$$

- Two basic approaches can be used:
  - Choose a fixed value of V around the point x and determine *k* from the data. Then determine p(x) from above. This is called Kernel Density Estimation (KDE).
  - Choose a fixed value of *k* and determine the corresponding volume V around x that includes k samples. This is called k Nearest Neighbor (kNN) approach.

Approaches

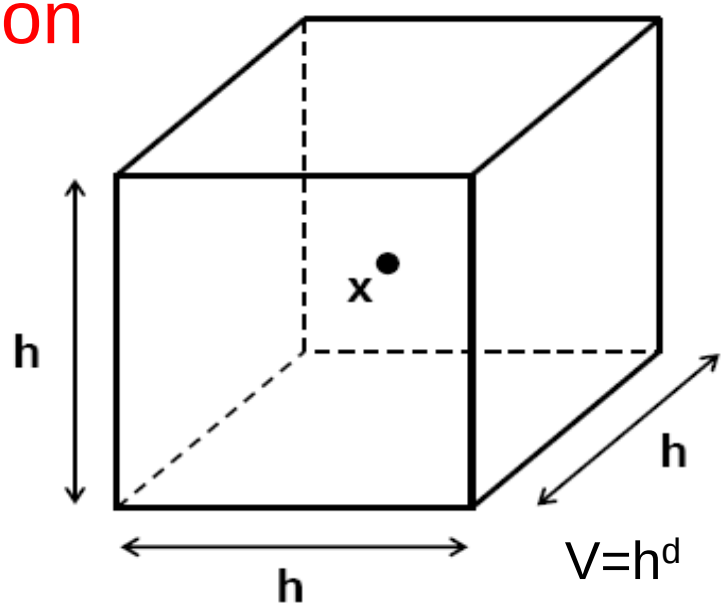
$$\boxed{p(x) \cong \frac{k}{nV}}$$

- Two basic approaches can be used:
  - Choose a fixed value of V around the point x and determine *k* from the data. Then determine p(x) from above. This is called Kernel Density Estimation (KDE).
  - Choose a fixed value of *k* and determine the corresponding volume V around x that includes k samples. This is called k Nearest Neighbor (kNN) approach.

Parzen Windows

- Suppose that the region R that encloses k examples is a d-dimensional hypercube with sides of length h centered at the estimation point x.
- To the find the number of instances that fall within this region we define a Kernel Function

$$\varphi(u_j) = \begin{cases} 1 & \text{if } |u_j| < 1/2 \quad \forall j = 1,\ldots d \\ 0 & otherwise \end{cases}$$



Emanuel Parzen

$$\varphi\left(\frac{(x - x_i)}{h}\right) = \begin{cases} 1 & \text{if } x_i \text{ inside hypercube of side h centered on x} \\ 0 & otherwise \end{cases}$$

Parzen Windows

- The total number of points inside the hypercube is then

$$k = \sum_{i=1}^{n} \varphi\left(\frac{(x - x_i)}{h}\right)$$

- Substitute back into the expression for the density estimate:

$$\boxed{p_{KDE}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} \varphi\left(\frac{(x - x_i)}{h}\right)}$$

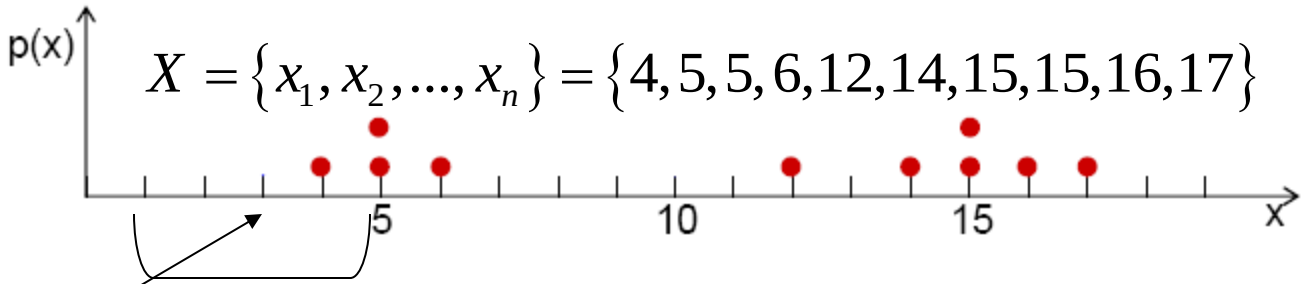
Number of samples falling into R

Volume covered by the window function

Width of the window function

Parzen Windows Example

- Given the dataset below, use Parzen windows of bandwidth h=4 to estimate the density p(x)



$$X = \{x_1, x_2, \ldots, x_n\} = \{4,5,5,6,12,14,15,15,16,17\}$$

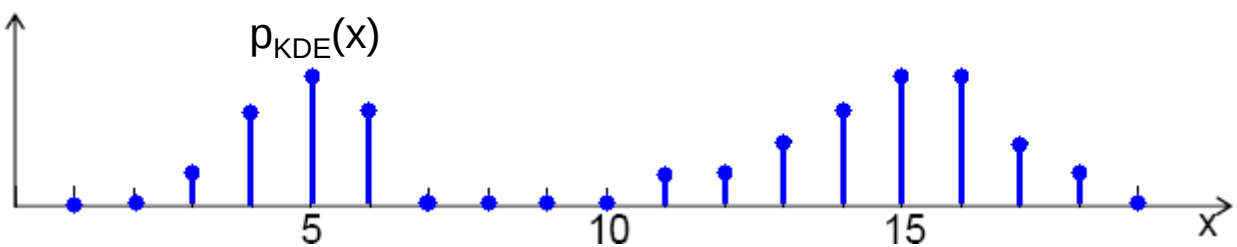
$$p_{KDE}(x=3) = \frac{1}{nh^d}\sum_{n=1}^{n}\varphi\left(\frac{x-x_n}{h}\right) = \frac{1}{10.4^1}\left[\varphi\underbrace{\left(\frac{3-4}{4}\right)}_{-1/4} + \varphi\underbrace{\left(\frac{3-5}{4}\right)}_{-1/2} + \varphi\underbrace{\left(\frac{3-5}{4}\right)}_{-1/2} + \varphi\underbrace{\left(\frac{3-6}{4}\right)}_{-1} + \ldots + \varphi\underbrace{\left(\frac{3-17}{4}\right)}_{-14/4}\right]$$

$$= \frac{1}{10.4^1}\left[1+0+0+0+\ldots+0\right] = 0.025$$

$$p_{KDE}(x=5) = \frac{1}{10.4^1}\left[1+1+1+1+0+0+0+0+0+0\right] = 0.10$$

$$p_{KDE}(x=10) = \frac{1}{10.4^1}\left[0+0+0+0+0+0+0+0+0+0\right] = 0$$

$$p_{KDE}(x=15) = \frac{1}{10.4^1}\left[0+0+0+0+0+1+1+1+1+0\right] = 0.10$$



Smooth Kernels

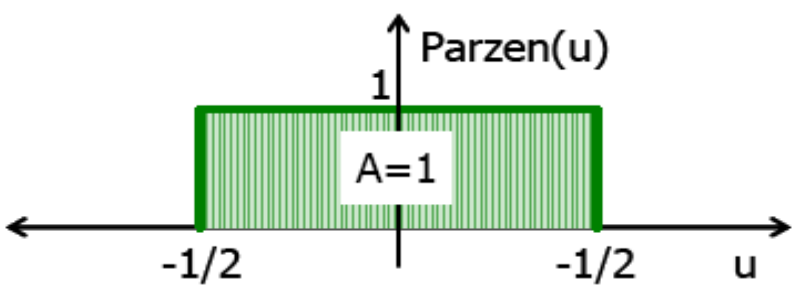
- The Parzen window has several issues
  - Errors due to discontinuities at window edges (same problem as with histograms)
  - Weights all points equally regardless of their distance to the estimation point x
- We can do better by generalizing to a smooth kernel function φ such that

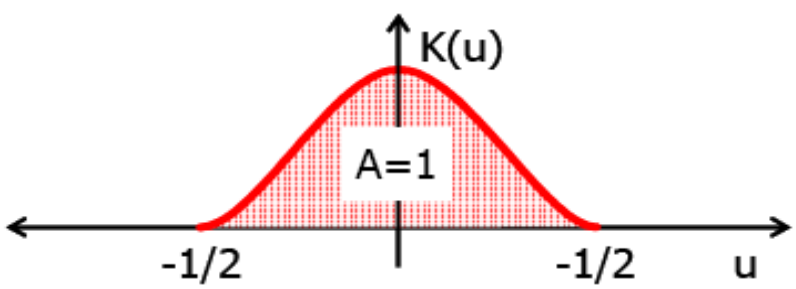
$$\int_{R^d} \varphi(x)\,dx = 1 \quad \text{and} \quad \varphi(x) \geq 0$$

  - Usually but not necessarily, φ(u) will be a radially symmetric and unimodal PDF, such as the multivariate Gaussian

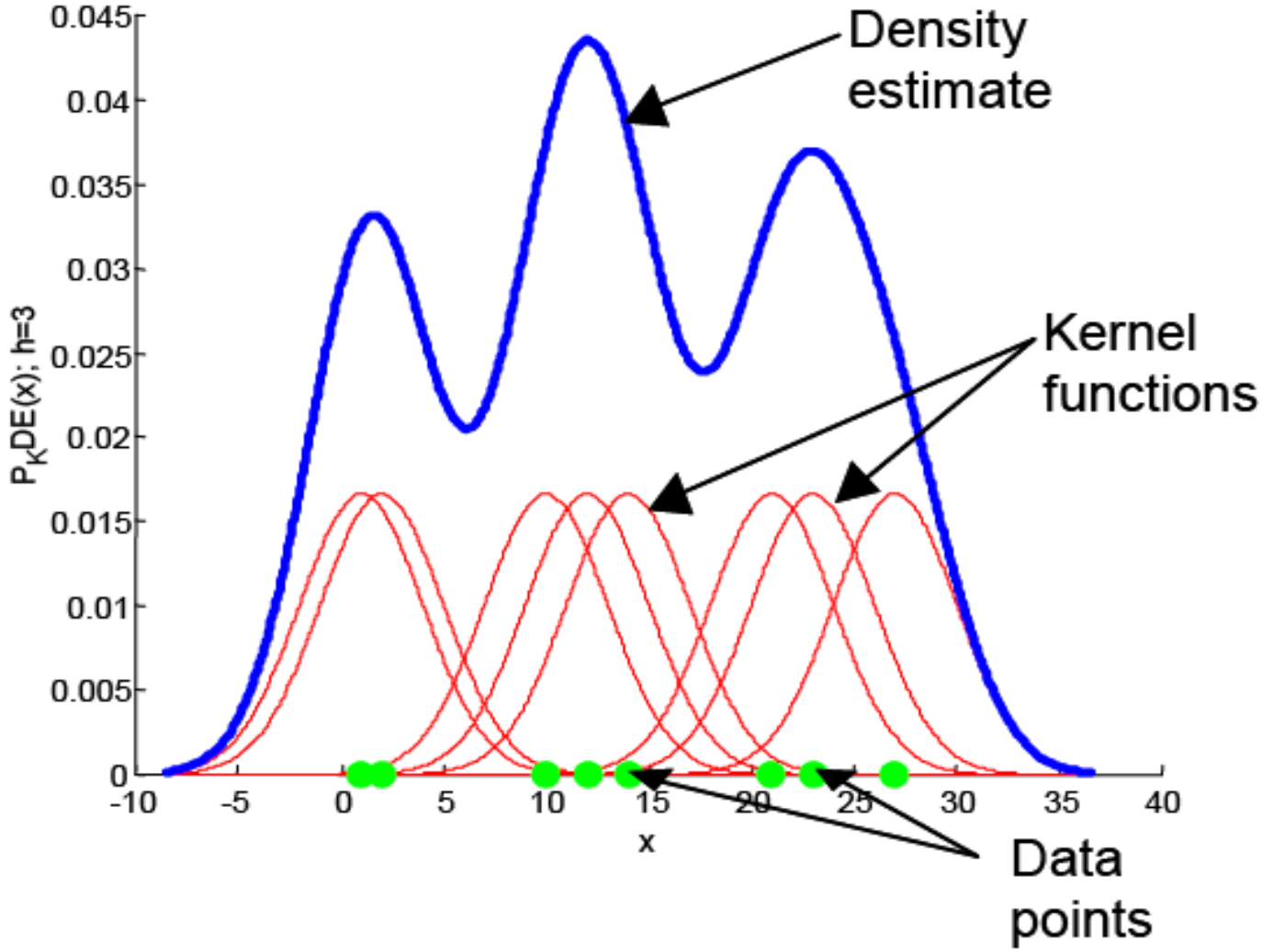
$$p_{KDE}(x) = \frac{1}{nh^d}\sum_{n=1}^{n}\varphi\left(\frac{(x-x_n)}{h}\right) = \frac{1}{nh^d}\sum_{n=1}^{n}\frac{1}{(2\pi)^{d/2}}\exp\left\{-\frac{\|x-x_n\|^2}{2h^2}\right\}$$

  - The expression of the density estimate remains the same as with Parzen windows



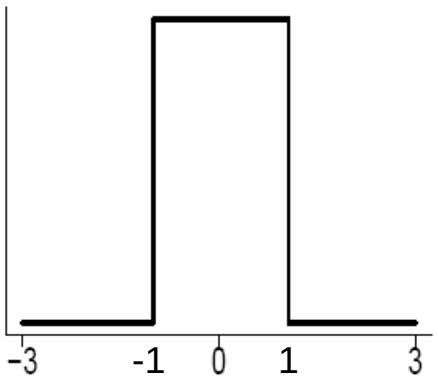


Kernel Density Estimation

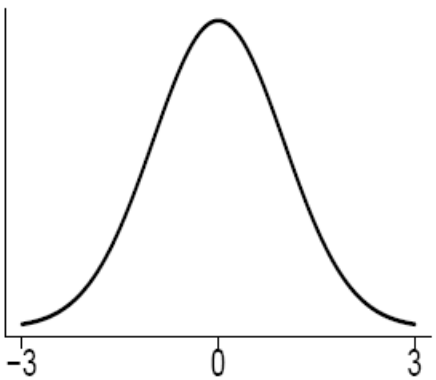


Commonly Used Kernels

Boxcar: $K(x) = \frac{1}{2} I(x)$

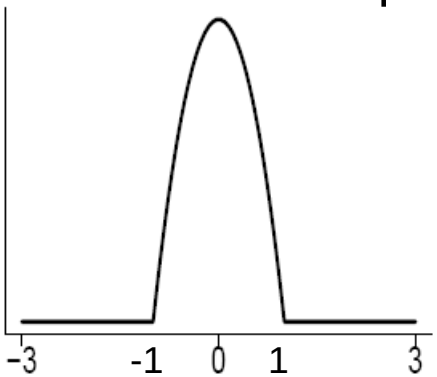


Gaussian: $K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$

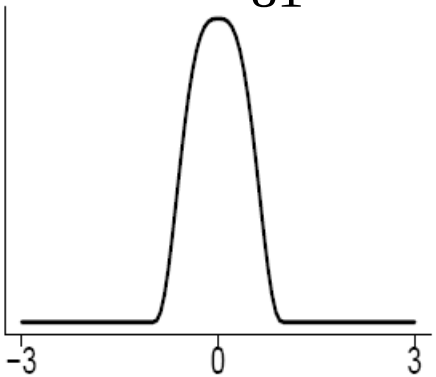


$$I(x) = \begin{cases} 1 & \text{if } |x| \le 1 \\ 0 & \text{else} \end{cases}$$

Epanechnikov: $K(x) = \frac{3}{4}(1 - x^2) I(x)$

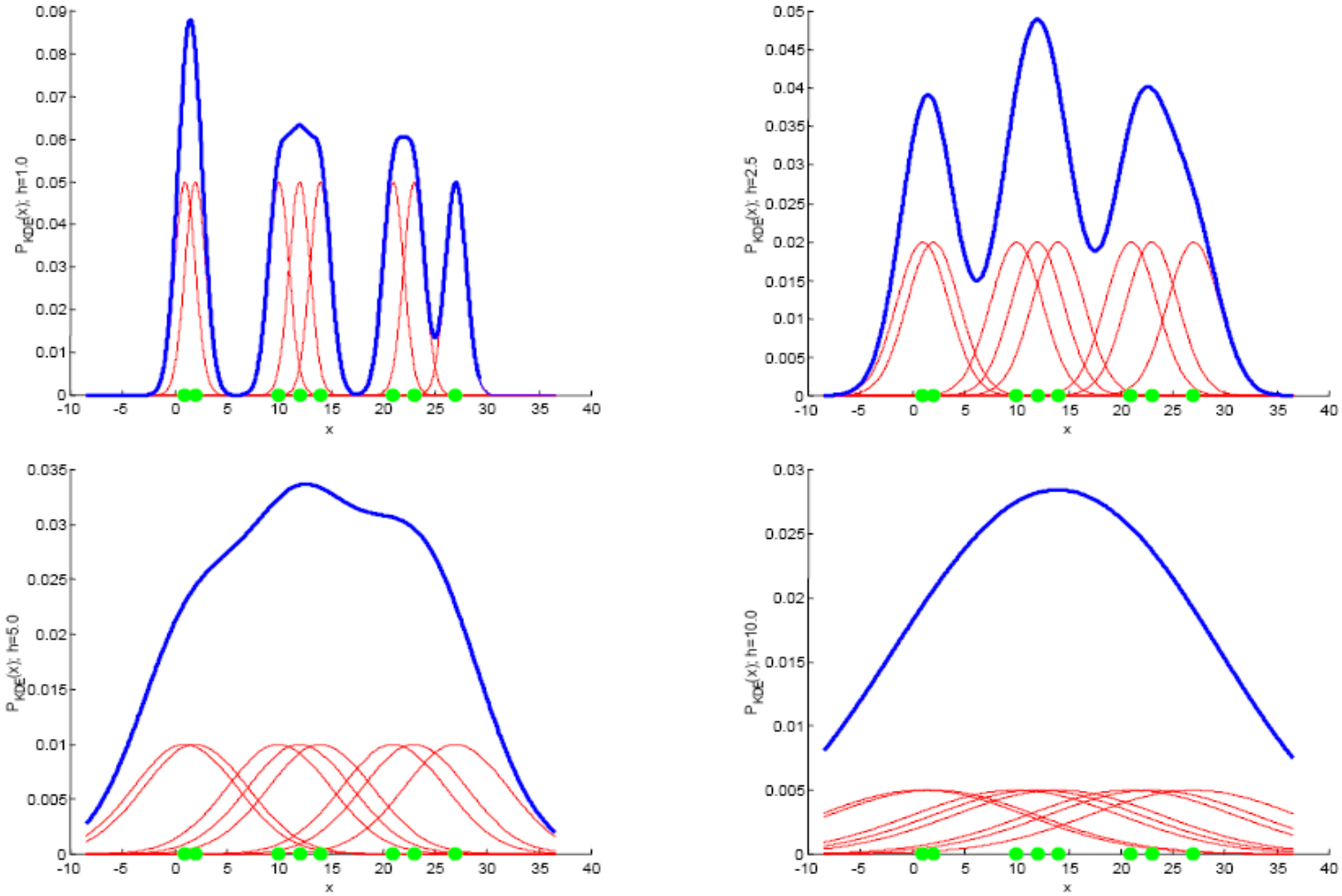


Tricube: $K(x) = \frac{70}{81}(1 - |x|^3)^3 I(x)$



Importance of Bandwidth

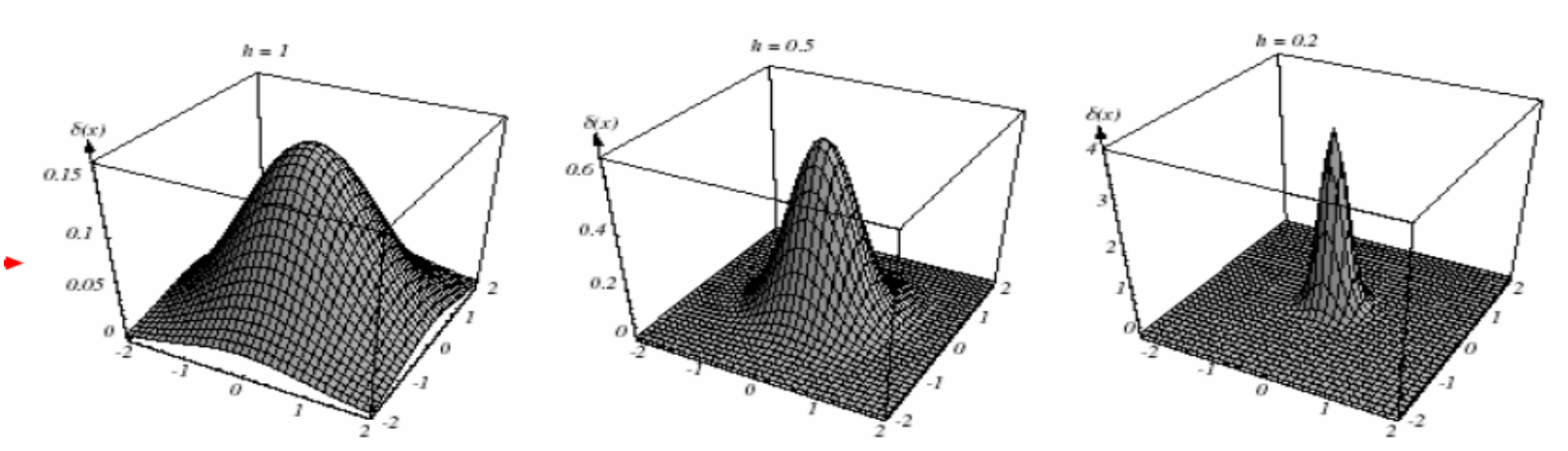
- A good compromise is needed for the bandwidth parameter
  - A large bandwidth will over smooth the density and erode the true structure of the data
  - A small bandwidth will yield a spiky density estimate



Small BW

Large BW

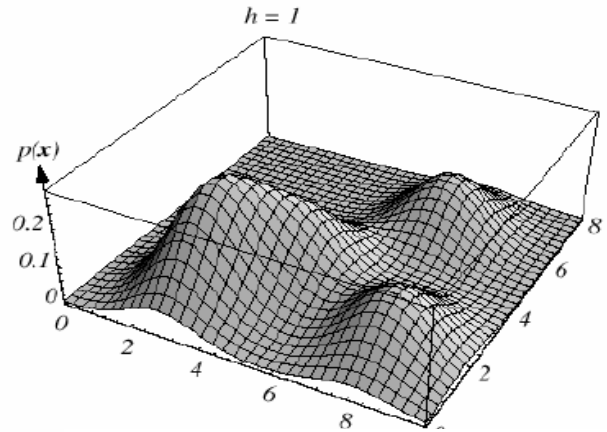
Importance of Bandwidth



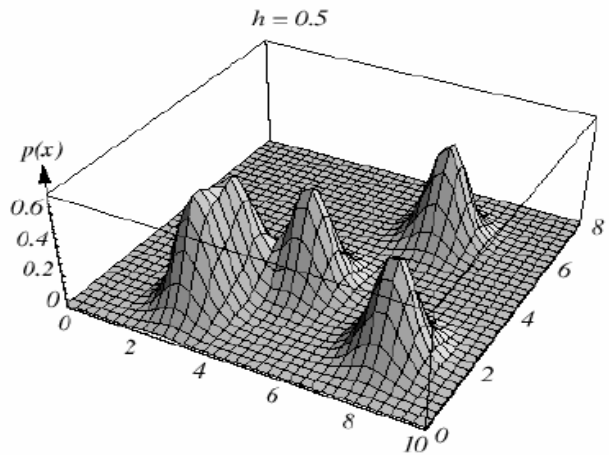
2-D

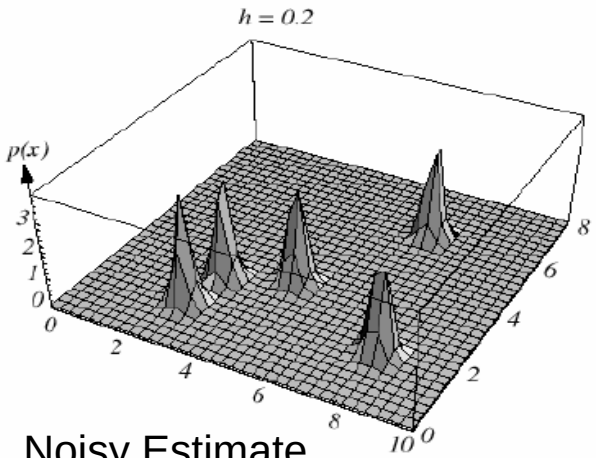
Large V_n ➔ too little resolution

Small V_n ➔ too much statistical variability



Out Of Focus Estimate





Noisy Estimate

Choosing the Bandwidth

- We would like to get a value of the smoothing parameter that minimizes the error between the estimated density and the true density
  - A natural measure is the MSE at the estimation point x, given by:

$$MSE\left(p_{KDE}\right) = E\left[\left(p_{KDE}(x) - p(x)\right)^2\right] = \left\{E\left[p_{KDE}(x) - p(x)\right]\right\}^2 + \mathrm{var}\left(p_{KDE}(x)\right)$$

$$\because \mathrm{var}\left(p_{KDE}(x)\right) = E\left[\left(p_{KDE}(x) - p(x)\right)^2\right] - \left\{E\left[p_{KDE}(x) - p(x)\right]\right\}^2$$

Bias-Variance Tradeoff

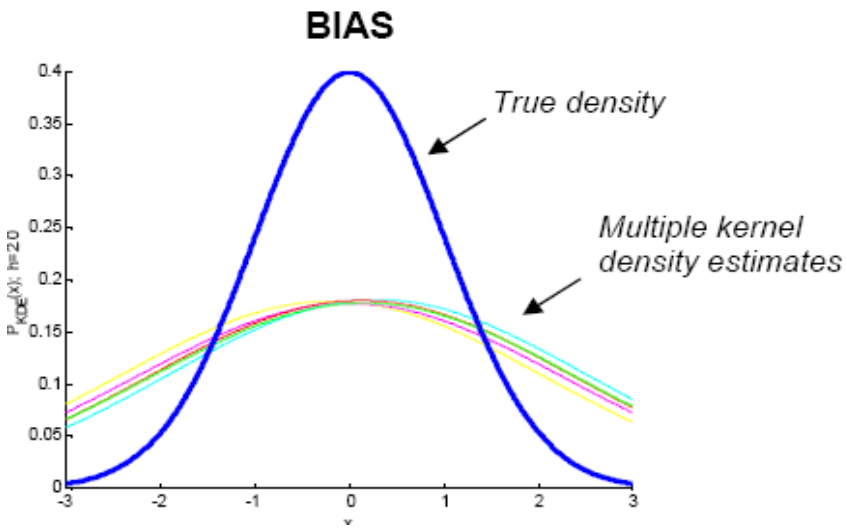
$$MSE\left(p_{KDE}\right)=\left\{E\left[p_{KDE}(x)-p(x)\right]\right\}^2+\mathrm{var}\left(p_{KDE}(x)\right)$$

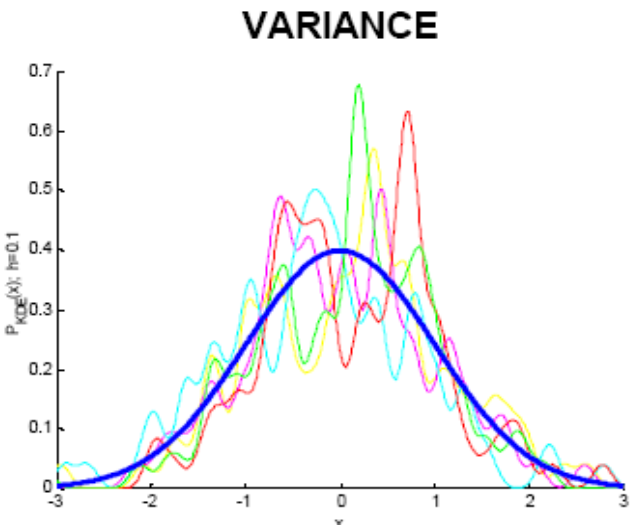
$$MSE\left(p_{KDE}\right)=\left(\mathrm{Bias}\right)^2+\mathrm{Variance}$$

- Bias can be reduced at the expense of variance and vice versa
  - The bias of an estimate is the systematic error incurred in the estimation
  - The variance of an estimate is the random error incurred in the estimation

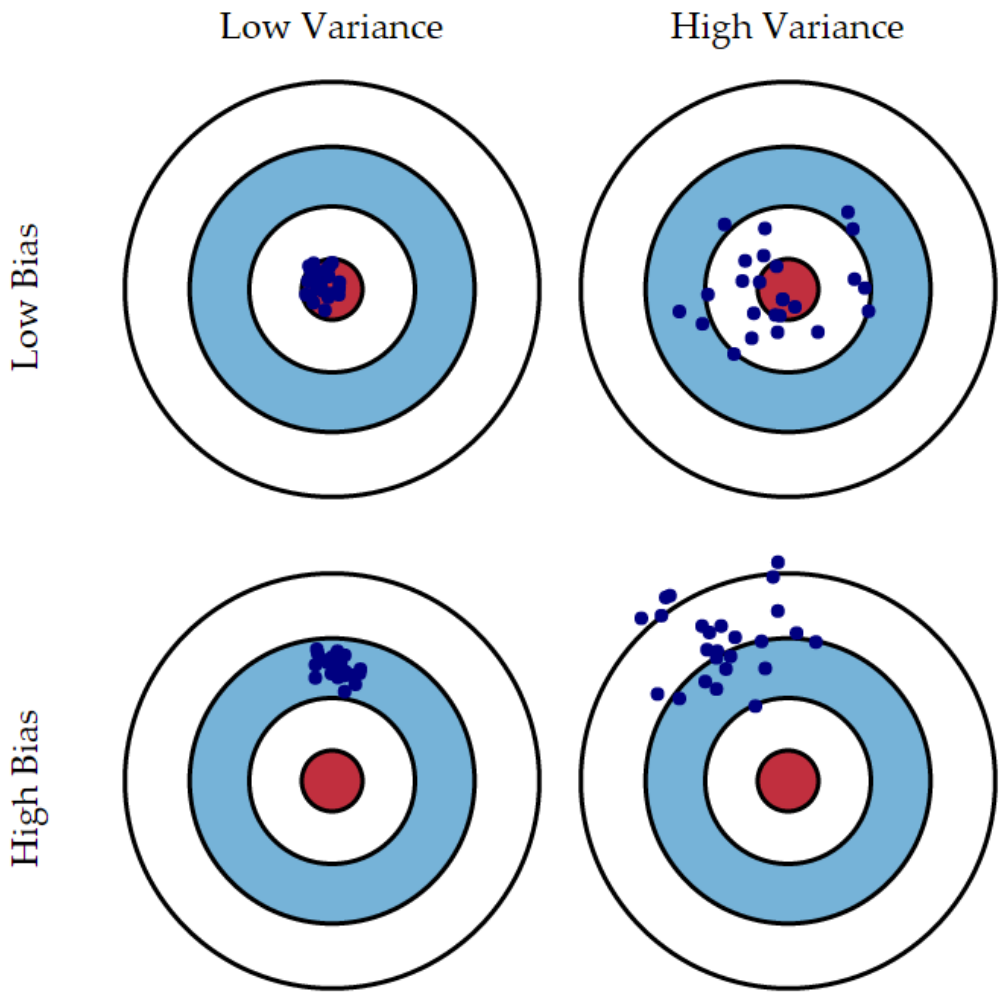
Bias-Variance Tradeoff

- The bias-variance dilemma of the bandwidth means that:
  - A large bandwidth will reduce the differences among the estimates of $p_{KDE}(x)$ for different datasets (i.e. the variance) but will increase the bias of $p_{KDE}(x)$ with respect to the true density $p(x)$
  - A small bandwidth will reduce the bias of $p_{KDE}(x)$ at the expense of a large variance in the estimates of $p_{KDE}(x)$





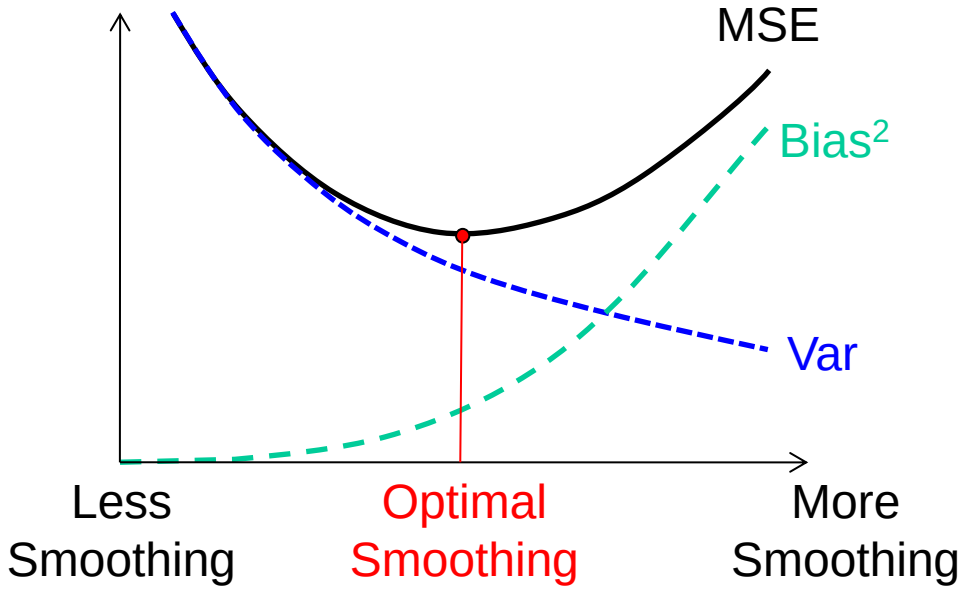
Bias-Variance Tradeoff



http://scott.fortmann-roe.com/docs/BiasVariance.html

Bias-Variance Tradeoff

The bias increases and the variance decreases with the amount of smoothing



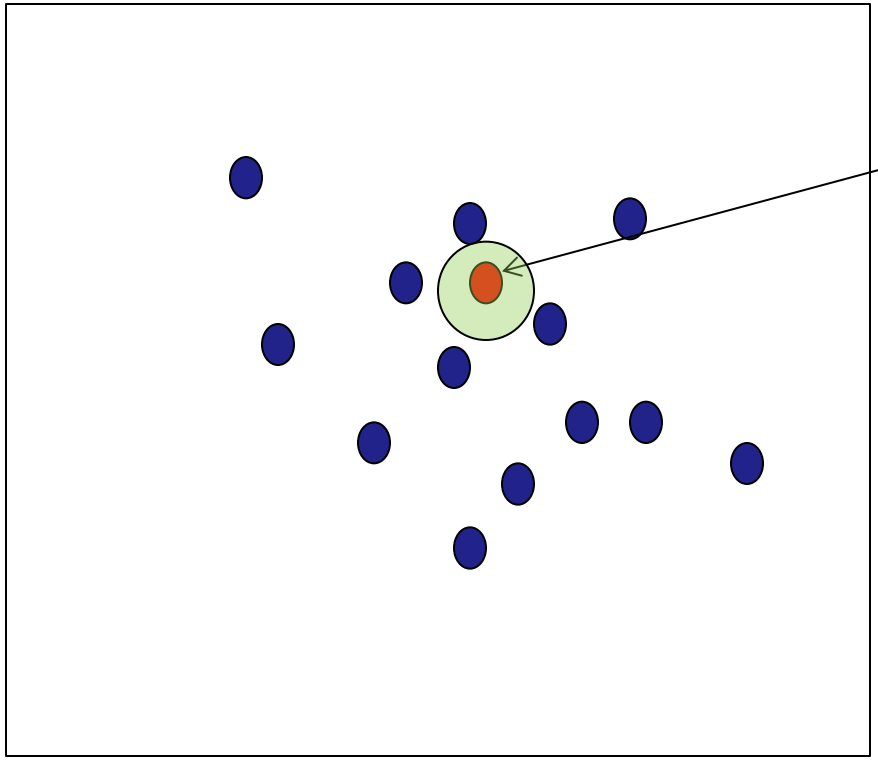
Density Estimation

$$\boxed{p(x) \cong \frac{k}{nV}}$$

- Two basic approaches can be used:
  - Choose a fixed value of V around the point x and determine *k* from the data. Then determine p(x) from above. This is called Kernel Density Estimation (KDE).
  - Choose a fixed value of *k* and determine the corresponding volume V around x that includes k samples. This is called k Nearest Neighbor (kNN) approach.

kNN Density Estimation

- In the kNN method we grow the volume surrounding the estimation point x until it encloses a total of *k* data points.



Estimate the density at this point

Volume V enclose k=4 samples

$$p(x) \cong \frac{k/n}{V}$$

kNN Estimate of Bayes Probabilities

- kNN can be used to estimate the prior, posterior, and likelihoods.
  - Assume that we have n total samples, and c total classes, and n_i samples inside each class such that $n = n_1+n_2+\ldots+n_c$
  - To classify an unknown data point x, we draw a hyper-sphere V around x. Assume that this volume contains a total of k samples, k_i of which come from class ωi, then,

$$p(\mathbf{x}\,|\,\omega_i) \cong \frac{k_i}{n_i V} \qquad p(\mathbf{x}) \cong \frac{k}{nV} \qquad p(\omega_i) \cong \frac{n_i}{n}$$

  - Put them all together using Bayes formula

$$P(\omega_i\,|\,\mathbf{x}) = \frac{p(\mathbf{x}\,|\,\omega_i)\,p(\omega_i)}{p(\mathbf{x})} \cong \frac{\dfrac{k_i}{n_i V}\cdot\dfrac{n_i}{n}}{\dfrac{k}{nV}} = \frac{k_i}{k}$$

- For k = 1, this is called the Nearest Neighbor rule

kNN Classification Rule

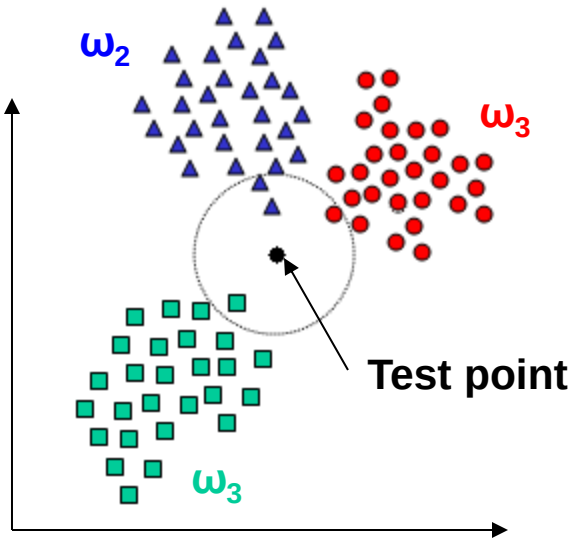
- kNN Rule is a simple and intuitive method that classifies unlabeled instances based on their proximity to examples in the training set.

$$p(\omega_i \mid \mathbf{x}) = \frac{k_i}{k}$$

  - Given an unlabeled test sample, find the k "closest" labeled examples in the training set and **assign it to the class that appears the most frequently** in this k-subset.

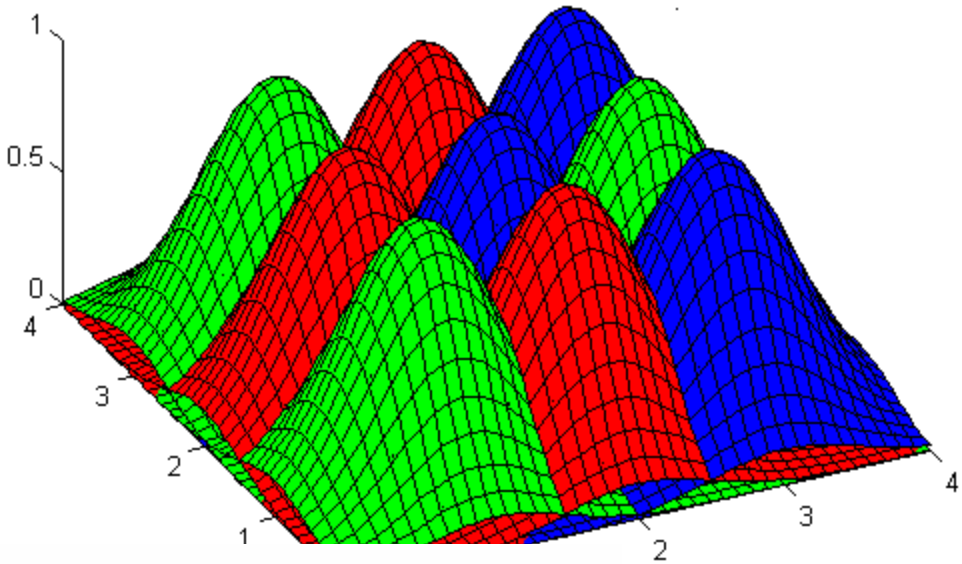
- kNN rule requires:
  - An integer k
  - A set of labeled training instances
  - A metric to measure proximity.

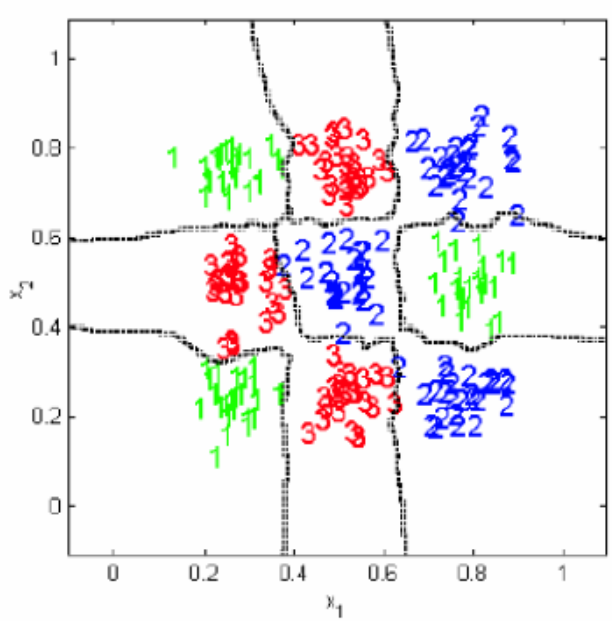
- In this example:
  - Use k = 4
  - Euclidian distance
  - The test point is assigned to class 2

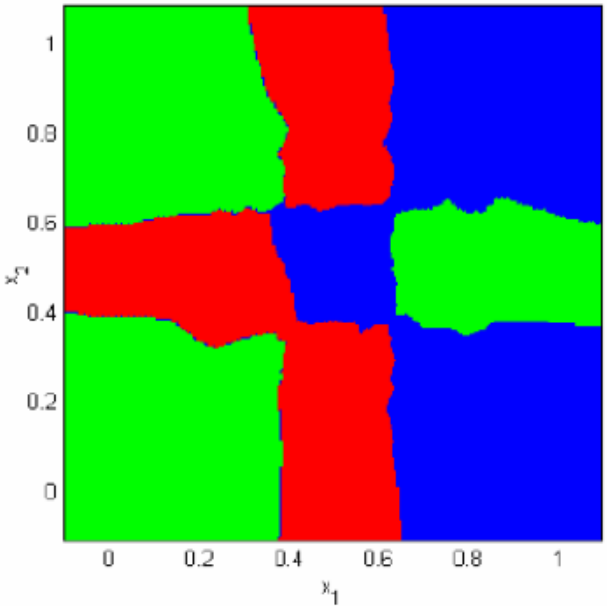


kNN Classification Rule

- 3-class problem unimodal non-linearly separable.
- Use k =5 and euclidian distance
- The resulting boundaries and decision regions are shown below





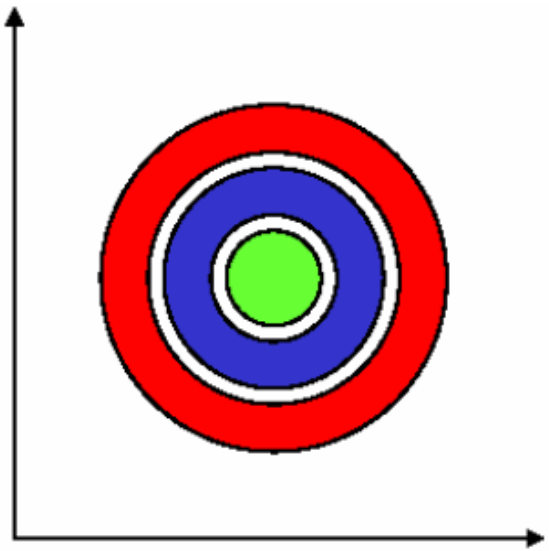


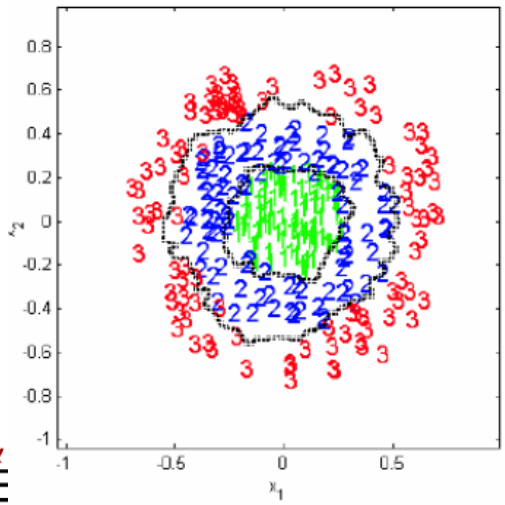
ENGINEERING

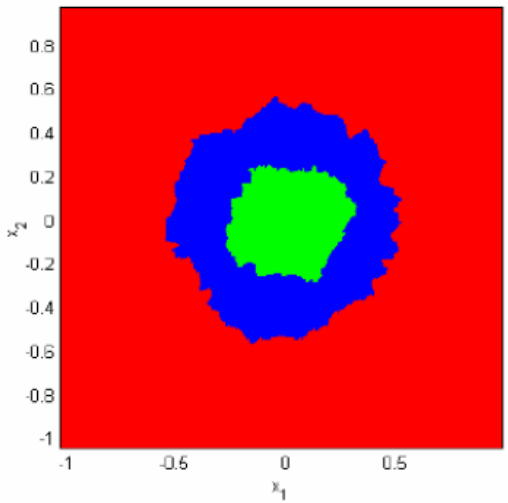
Courtesy of R. Gutierrez

kNN Classification Rule

- 3-class problem unimodal non-linearly separable.
- Use k =5 and euclidian distance
- The resulting boundaries and decision regions are shown below







Courtesy of R. Gutierrez

kNN Decision Boundary

The nearest neighbor rule leads to a Voronoi Tesselation

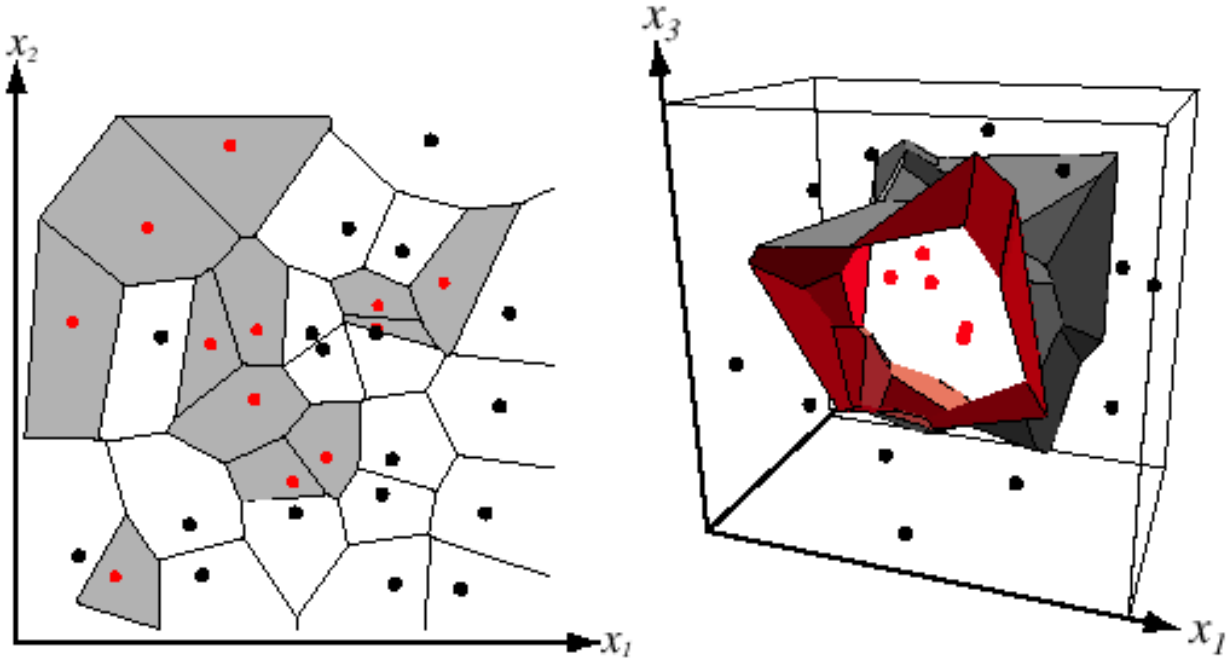


FIGURE 4.13. In two dimensions, the nearest-neighbor algorithm leads to a partitioning of the input space into Voronoi cells, each labeled by the category of the training point it contains. In three dimensions, the cells are three-dimensional, and the decision boundary resembles the surface of a crystal. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Georgy Voronoy

1868-1908

Johann Peter Gustav Lejeune Dirichlet

1805-1859

Recap

- Non-Parametric Density Estimation
  - Problem Formulation
  - Different Approaches
- Histograms
- Kernels and Kernel Density Estimation
  - Parzen Windows
- kNN Density Estimation and Classification