Last updated: September 17, 2012

BAYESIAN DECISION THEORY

J. Elder

CSE 4404/5327 Introduction to Machine Learning and Pattern Recognition

Problems

- The following problems from the textbook are relevant:
  - 2.1 – 2.9, 2.11, 2.17
- For this week, please at least solve Problem 2.3. We will go over this in class.

Credits

- Some of these slides were sourced and/or modified from:
  - Christopher Bishop, Microsoft UK
  - Simon Prince, University College London
  - Sergios Theodoridis, University of Athens & Konstantinos Koutroumbas, National Observatory of Athens

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. Nonparametric Density Estimation
6. Training and Evaluation Methods

Bayesian Decision Theory: Topics

1. **Probability**
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. Nonparametric Density Estimation
6. Training and Evaluation Methods

Probability

- "Probability theory is nothing but common sense reduced to calculation"

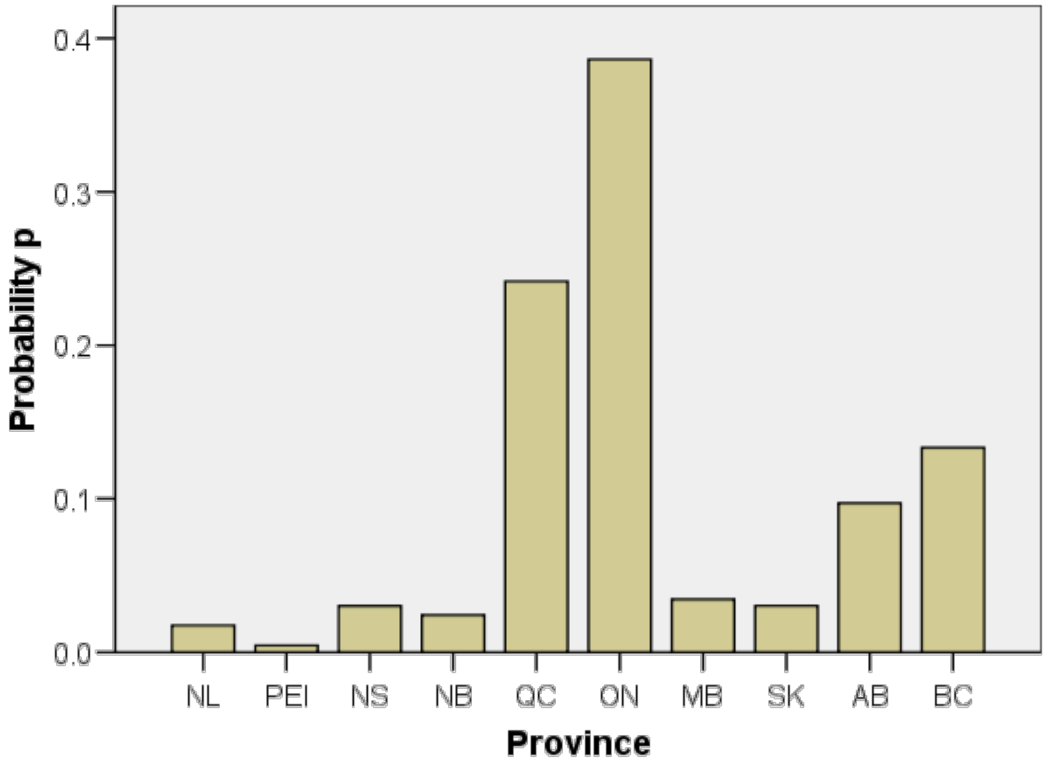
  - Pierre Laplace, 1812.

Random Variables

- A **random variable** is a variable whose value is uncertain.
- For example, the height of a randomly selected person in this class is a random variable – I won't know its value until the person is selected.
- Note that we are not completely uncertain about most random variables.
  - For example, we know that height will probably be in the 5'-6' range.
  - In addition, 5'6" is more likely than 5'0" or 6'0".
- The function that describes the probability of each possible value of the random variable is called a **probability distribution**.

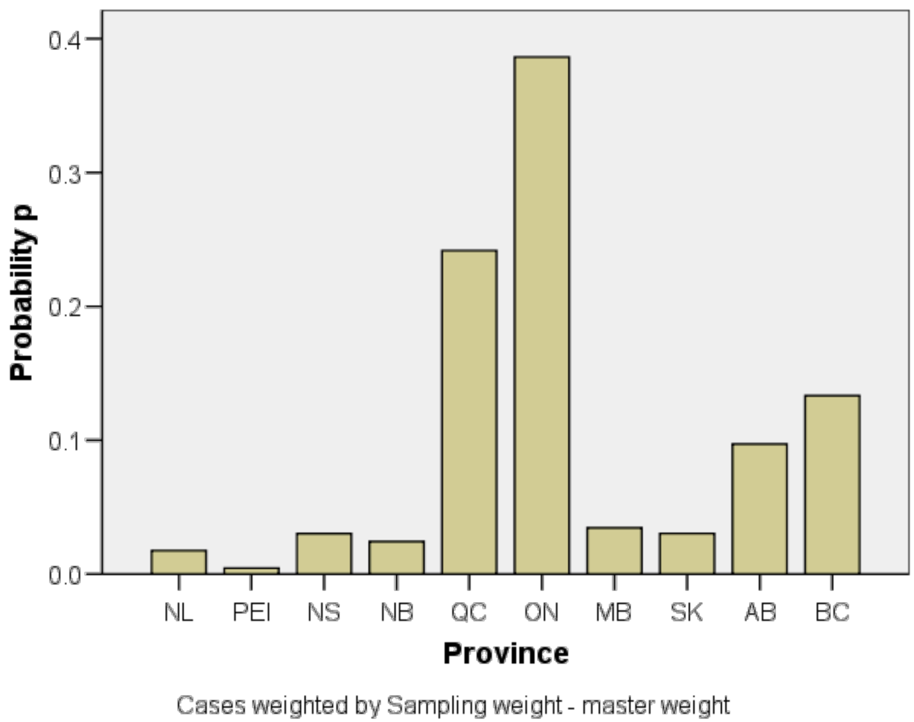
Probability Distributions

- For a **discrete** distribution, the probabilities over all possible values of the random variable must **sum** to 1.



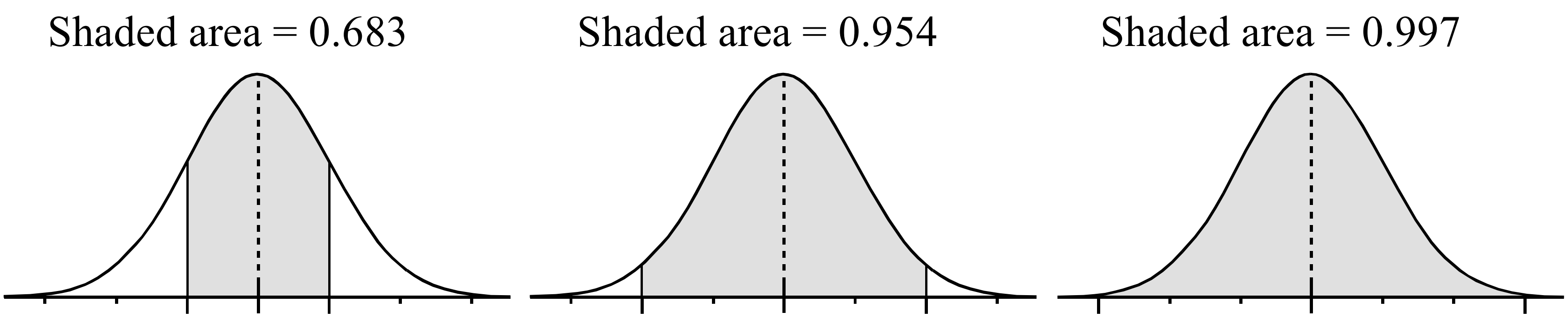
Probability Distributions

- For a **discrete** distribution, we can talk about the probability of a particular score occurring, e.g., p(Province = Ontario) = 0.36.
- We can also talk about the probability of any one of a subset of scores occurring, e.g., p(Province = Ontario or Quebec) = 0.50.
- In general, we refer to these occurrences as **events**.



Probability Distributions

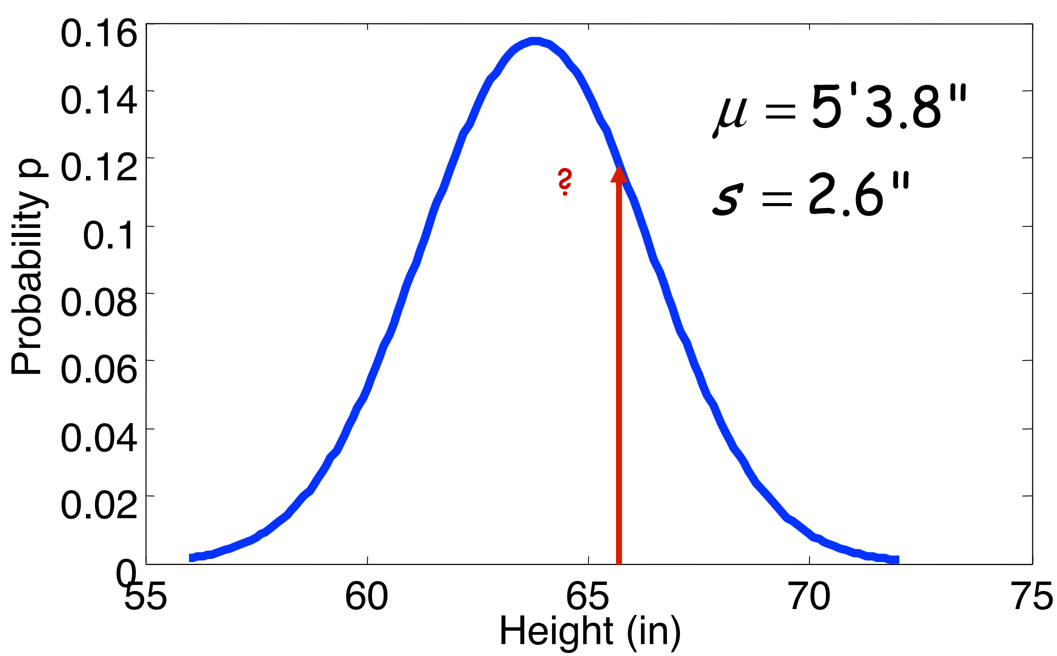
- For a **continuous** distribution, the probabilities over all possible values of the random variable must **integrate** to 1 (i.e., the area under the curve must be 1).
- Note that the height of a continuous distribution can exceed 1!



Continuous Distributions

- For continuous distributions, it **does not** make sense to talk about the probability of an exact score.
  - e.g., what is the probability that your height is exactly 65.485948467… inches?

Normal Approximation to probability distribution for height of Canadian females
(parameters from General Social Survey, 1991)

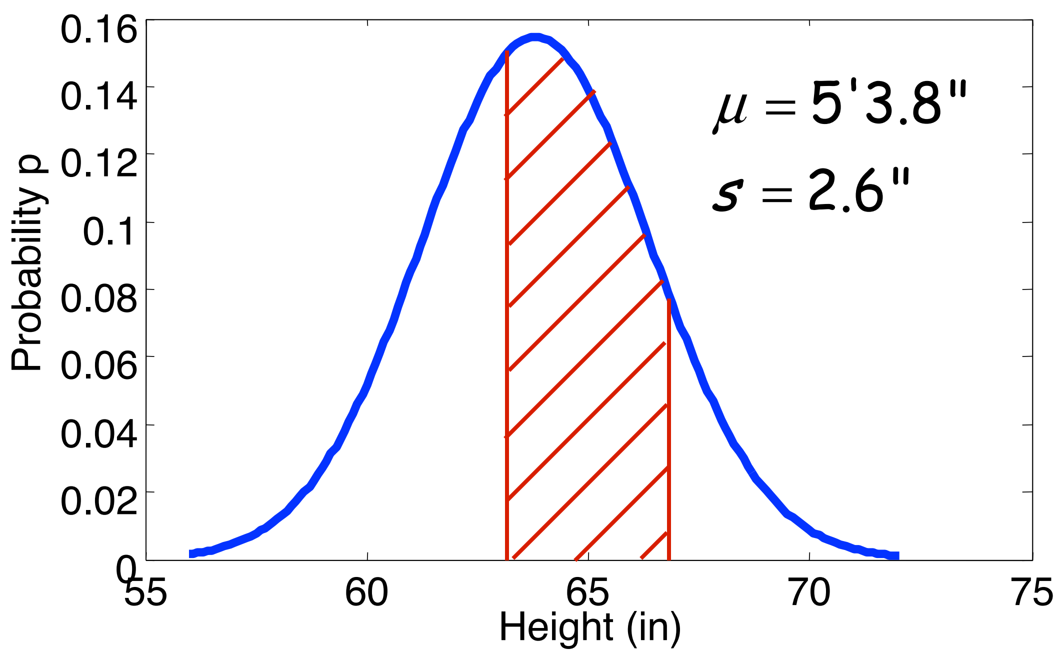


Continuous Distributions

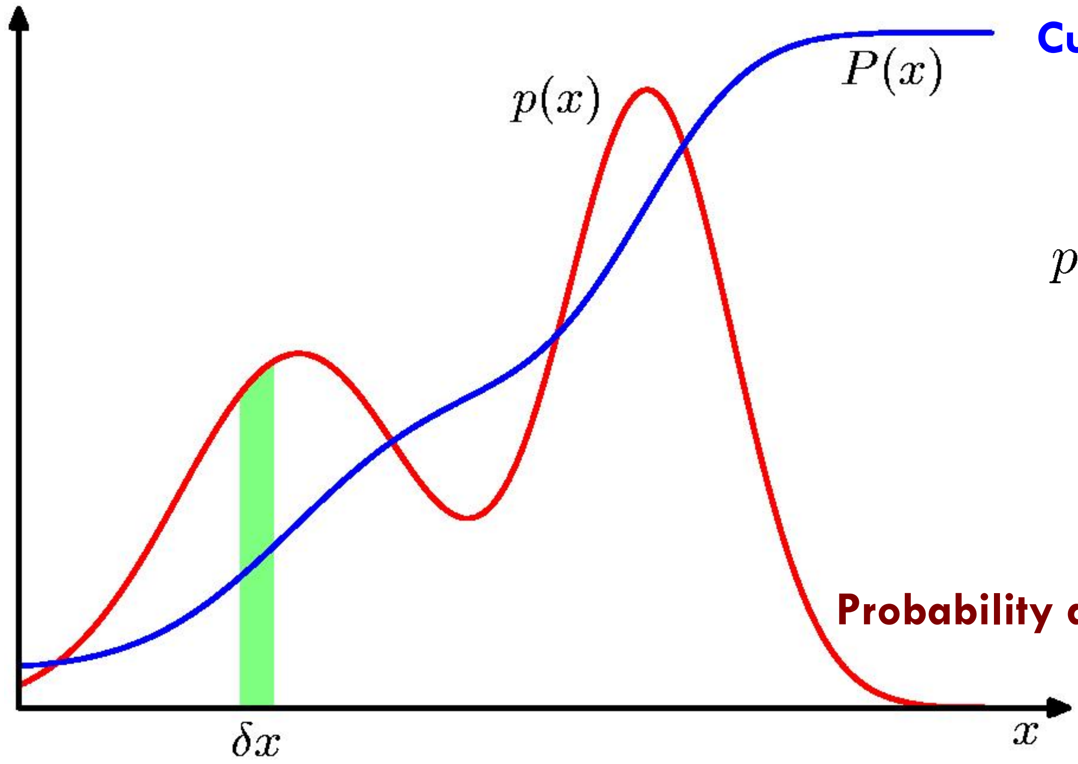
- It **does** make sense to talk about the probability of observing a score that falls within a certain range
  - e.g., what is the probability that you are between 5'3" and 5'7"?
  - e.g., what is the probability that you are less than 5'10"?

Valid events

Normal Approximation to probability distribution for height of Canadian females
(parameters from General Social Survey, 1991)



Probability Densities



Cumulative distribution (CDF)

$$p(x \in (a, b)) = \int_a^b p(x)\,\mathrm{d}x$$

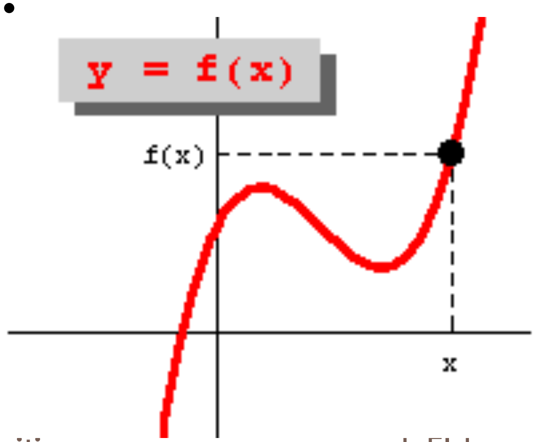
$$P(z) = \int_{-\infty}^{z} p(x)\,\mathrm{d}x$$

Probability density (PDF)

$$p(x) \geqslant 0 \qquad \int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1$$

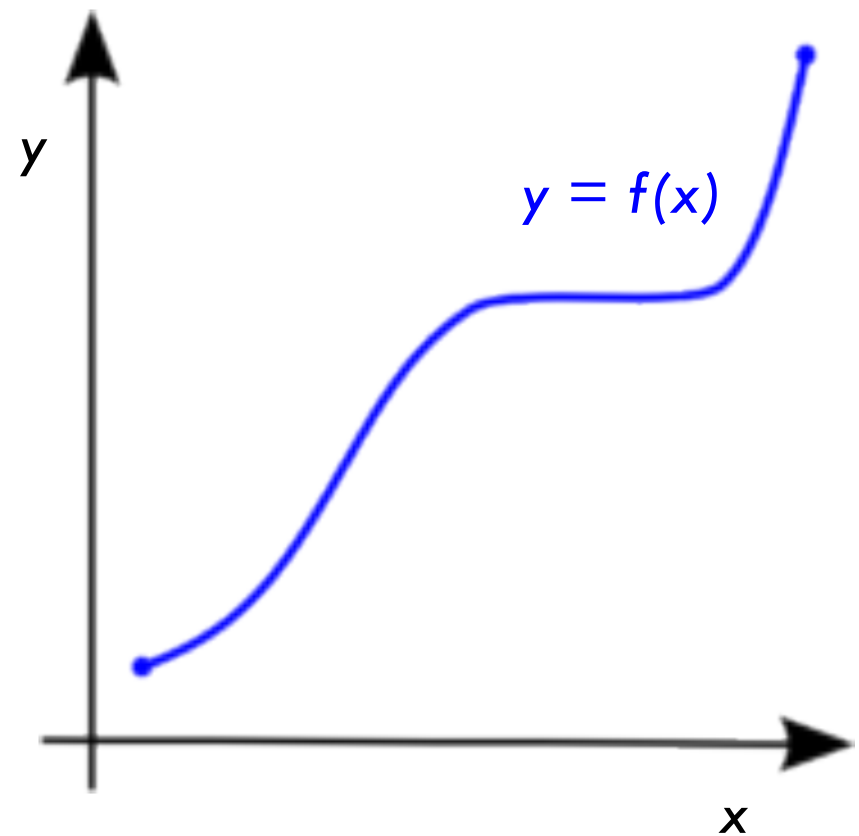
Transformed Densities

- Consider a random variable x with probability density $p_x(x)$.
- Suppose you have another variable y that is defined to be a function of x: $y = f(x)$.
- y is also a random variable. What is its probability density $p_y(y)$?
- **Caution**: in general, $p_y(y) \neq p_x(f^{-1}(y))$.



Transformed Densities

- This is a difficult problem in general.
- However, it is tractable when $f(x)$ is monotonic, and hence invertible.
- In this case, we can solve for the pdf $p_y(y)$ by differentiating the cdf $P_y(y)$.



Transformed Densities

- Let's assume that y is monotonically increasing in x. Then we can write

$$P_y(y) = P\left(f(x) \le y\right) = P\left(x \le f^{-1}(y)\right) = P_x\left(f^{-1}(y)\right)$$

- Taking derivatives, we get

$$p_y(y) \triangleq \frac{d}{dy} P_y(y) = \frac{d}{dy} P_x\left(f^{-1}(y)\right) = \frac{dx}{dy}\frac{d}{dx} P_x\left(x\right) = \frac{dx}{dy} p_x\left(x\right)$$

where $x = f^{-1}(y)$.

Note that $\dfrac{dx}{dy} > 0$ in this case.

Transformed Densities

- If y is monotonically decreasing in x, using the same method it is easy to show that

$$p_y(y) = -\frac{dx}{dy} p_x(x)$$

where $x = f^{-1}(y)$.

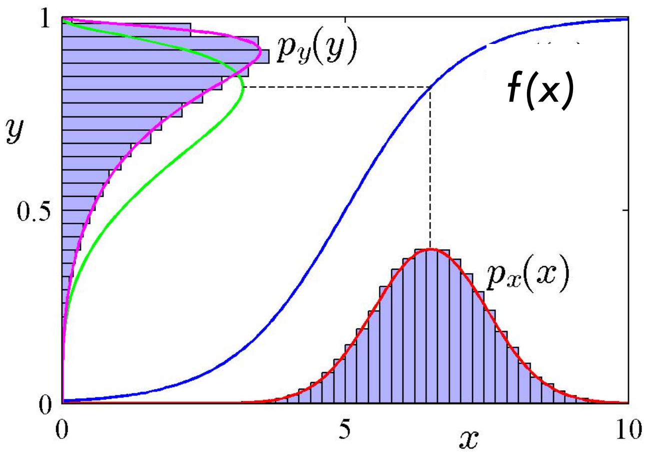
Note that $\frac{dx}{dy} < 0$ in this case.

- Thus a general expression that applies when y is monotonic on x is:

$$p_y(y) = \left|\frac{dx}{dy}\right| p_x(x),$$

where $x = f^{-1}(y)$.

Transformed Densities: Intuition



Observations falling within $(x+\delta x)$ tranform to the range $(y+\delta y)$

$$\rightarrow p_x(x)\left|\delta x\right| = p_y(y)\left|\delta y\right|$$

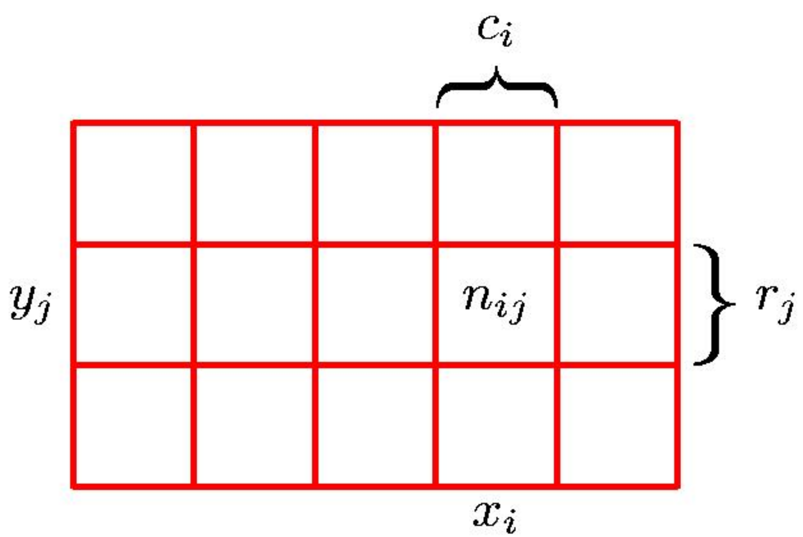
$$\rightarrow p_y(y) \simeq p_x(x)\left|\frac{\delta x}{\delta y}\right|$$

Note that in general, $\delta y \neq \delta x$.

Rather, $\dfrac{\delta y}{\delta x} \rightarrow \dfrac{dy}{dx}$ as $\delta x \rightarrow 0$.

Thus $p_y(y) = p_x(x)\left|\dfrac{dx}{dy}\right|$

Joint Distributions



Marginal Probability

$$p(X = x_i) = \frac{c_i}{N}.$$

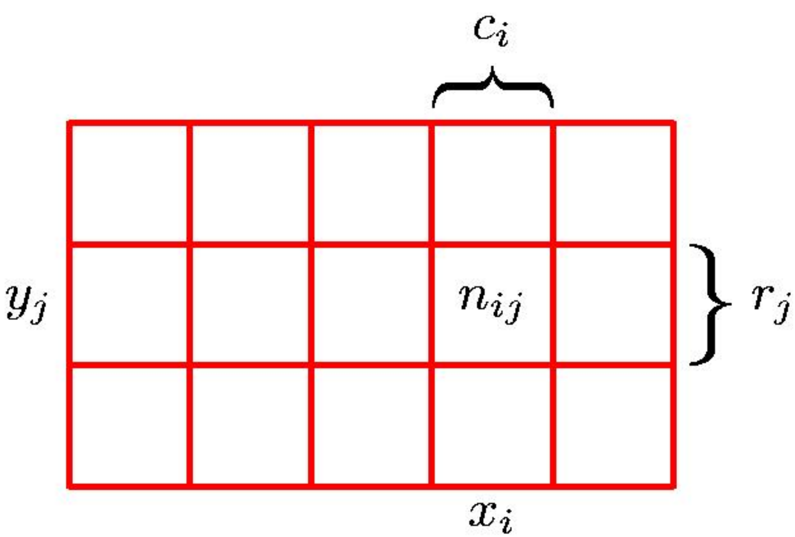
Joint Probability

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

Conditional Probability

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

Joint Distributions



Sum Rule

$$p(X = x_i) = \frac{c_i}{N} = \frac{1}{N}\sum_{j=1}^{L} n_{ij}$$
$$= \sum_{j=1}^{L} p(X = x_i, Y = y_j)$$

Product Rule

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N}$$
$$= p(Y = y_j | X = x_i) p(X = x_i)$$

Joint Distributions: The Rules of Probability

- Sum Rule $$p(X) = \sum_Y p(X, Y)$$

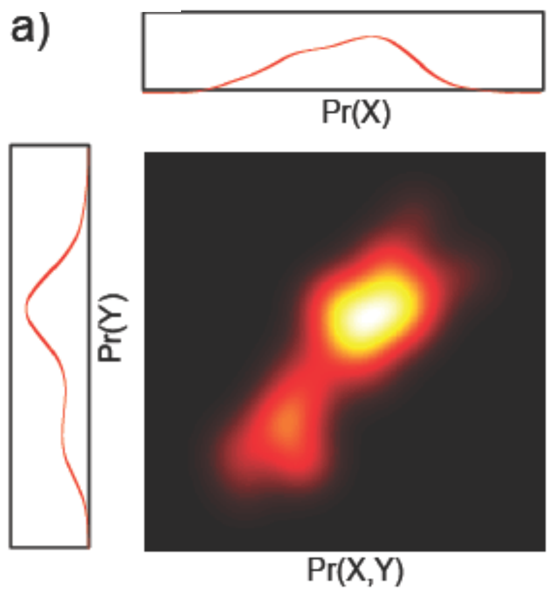
- Product Rule $$p(X, Y) = p(Y|X)p(X)$$

Marginalization

We can recover probability distribution of any variable in a joint distribution by integrating (or summing) over the other variables

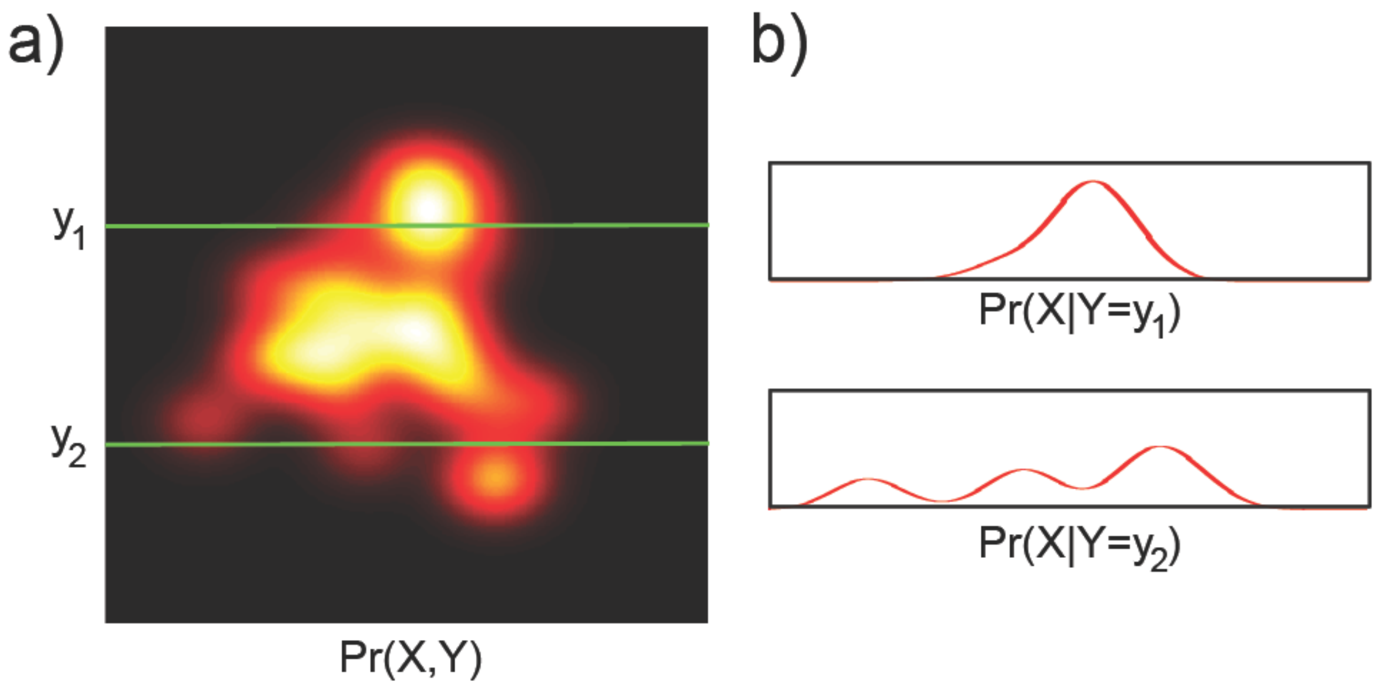
$$
\begin{aligned}
Pr(X) &= \int Pr(X,Y)dY \\
Pr(Y) &= \int Pr(X,Y)dX
\end{aligned}
$$

$$Pr(X,Y) = \sum_W \sum_Z Pr(W,X,Y,Z)$$



Conditional Probability

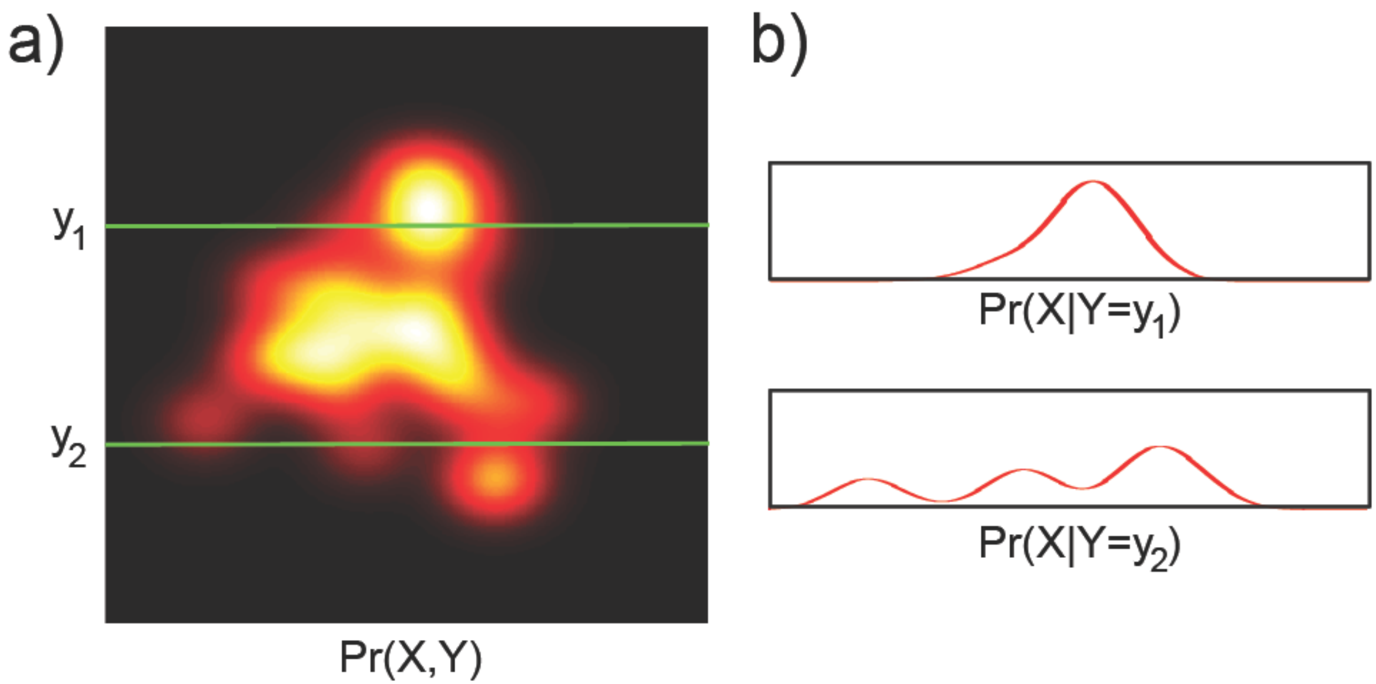
- Conditional probability of X given that Y=y* is relative propensity of variable X to take different outcomes given that Y is fixed to be equal to y*
- Written as Pr(X|Y=y*)



Conditional Probability

- Conditional probability can be extracted from joint probability
- Extract appropriate slice and normalize

$$Pr(X|Y=y^*) = \frac{Pr(X,Y=y^*)}{\int (Pr(X,Y=y^*)dX} = \frac{Pr(X,Y=y^*)}{Pr(Y=y^*)}$$



Conditional Probability

$$Pr(X|Y=y^*) = \frac{Pr(X,Y=y^*)}{\int(Pr(X,Y=y^*)dX} = \frac{Pr(X,Y=y^*)}{Pr(Y=y^*)}$$

- More usually written in compact form

$$Pr(X|Y) = \frac{Pr(X,Y)}{Pr(Y)}$$

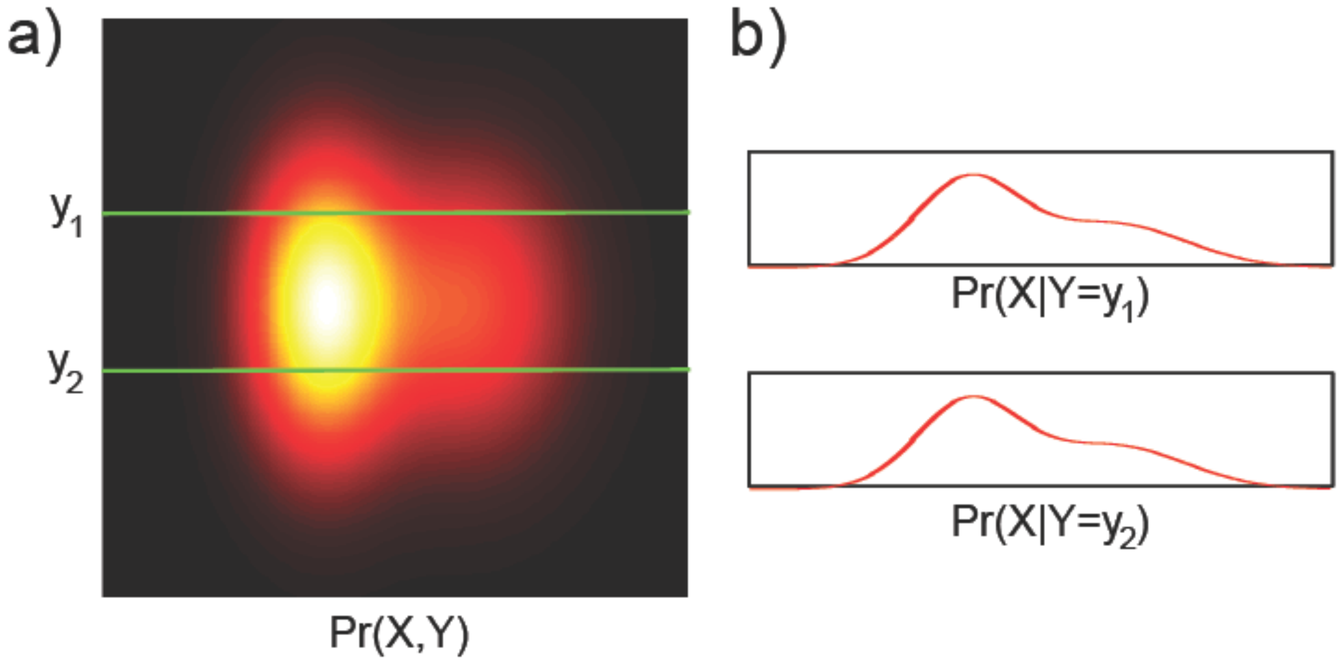
- Can be re-arranged to give

$$Pr(X,Y) = Pr(X|Y)Pr(Y)$$

Independence

- If two variables X and Y are independent then variable X tells us nothing about variable Y (and vice-versa)

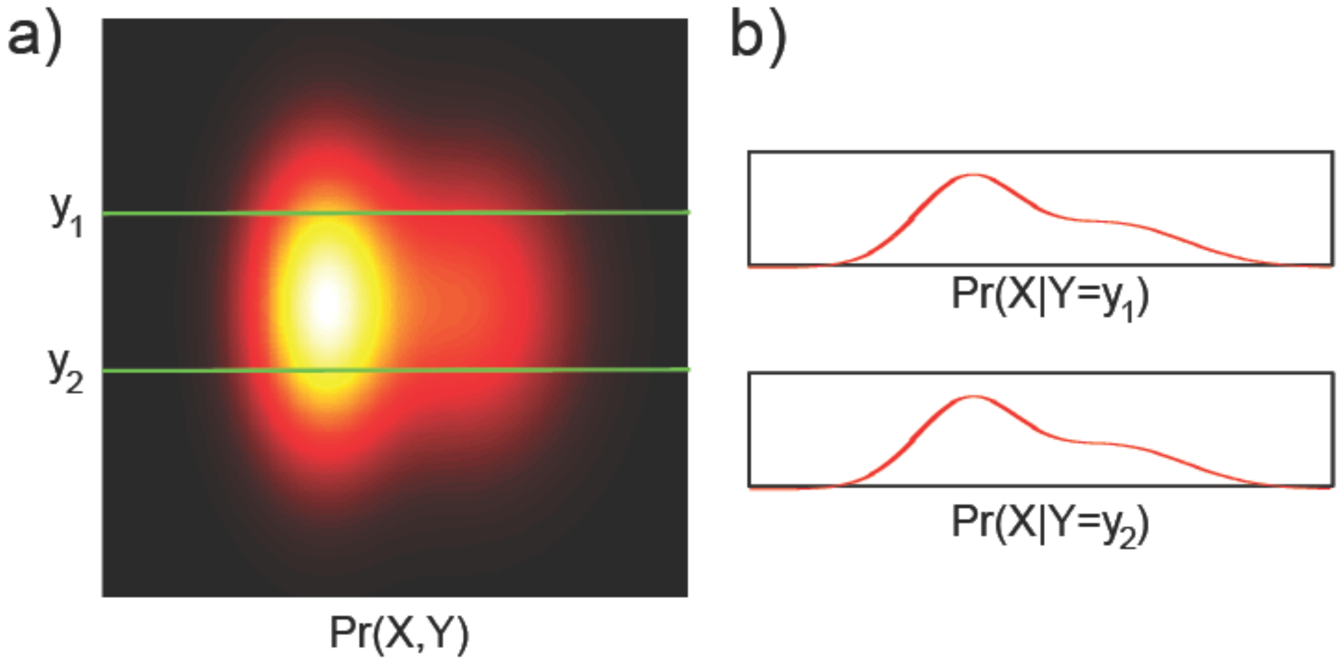
$$
\begin{aligned}
Pr(X|Y) &= Pr(X) \\
Pr(Y|X) &= Pr(Y)
\end{aligned}
$$



Independence

- When variables are independent, the joint factorizes into a product of the marginals:

$$\begin{aligned} Pr(X,Y) &= Pr(X|Y)Pr(Y) \\ &= Pr(X)Pr(Y) \end{aligned}$$



Bayes' Rule

From before:

$$Pr(X,Y) = Pr(X|Y)Pr(Y)$$
$$Pr(X,Y) = Pr(Y|X)Pr(X)$$

Combining:

$$Pr(Y|X)Pr(X) = Pr(X|Y)Pr(Y)$$

Re-arranging:

$$\begin{aligned} Pr(Y|X) &= \frac{Pr(X|Y)Pr(Y)}{Pr(X)} \\ &= \frac{Pr(X|Y)Pr(Y)}{\int Pr(X,Y)dY} \\ &= \frac{Pr(X|Y)Pr(Y)}{\int Pr(X|Y)Pr(Y)dY} \end{aligned}$$

Bayes' Rule Terminology

Likelihood – propensity for observing a certain value of X given a certain value of Y

Prior – what we know about y before seeing x

$$Pr(Y|X) = \frac{Pr(X|Y)Pr(Y)}{Pr(X)}$$

Posterior – what we know about y after seeing x

Evidence –a constant to ensure that the left hand side is a valid distribution

Expectations

- Let *f(x)* be some function of a random variable *x*. Then we define:

$$\mathbb{E}[f] = \sum_x p(x) f(x)$$

$$\mathbb{E}[f] = \int p(x) f(x) \,\mathrm{d}x$$

$$\mathbb{E}_x[f|y] = \sum_x p(x|y) f(x)$$

Conditional Expectation
(discrete)

$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^{N} f(x_n)$$

Approximate Expectation
(discrete and continuous)

Variances and Covariances

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

$$\begin{aligned}\mathrm{cov}[x, y] &= \mathbb{E}_{x,y}\left[\{x - \mathbb{E}[x]\}\{y - \mathbb{E}[y]\}\right] \\ &= \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y]\end{aligned}$$

$$\begin{aligned}\mathrm{cov}[\mathbf{x}, \mathbf{y}] &= \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\}\{\mathbf{y}^{\mathrm{T}} - \mathbb{E}[\mathbf{y}^{\mathrm{T}}]\}\right] \\ &= \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbf{x}\mathbf{y}^{\mathrm{T}}] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^{\mathrm{T}}]\end{aligned}$$

End of Lecture

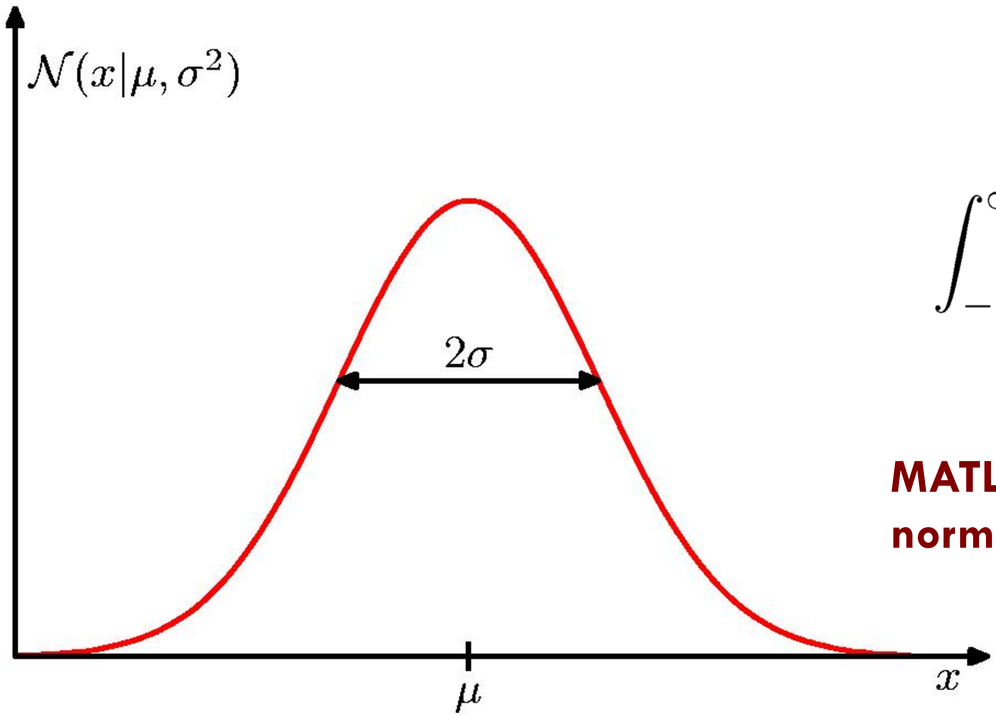
Sept 10, 2012

Bayesian Decision Theory: Topics

1. Probability
2. **The Univariate Normal Distribution**
3. Bayesian Classifiers
4. Minimizing Risk
5. Nonparametric Density Estimation
6. Training and Evaluation Methods

The Gaussian Distribution

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



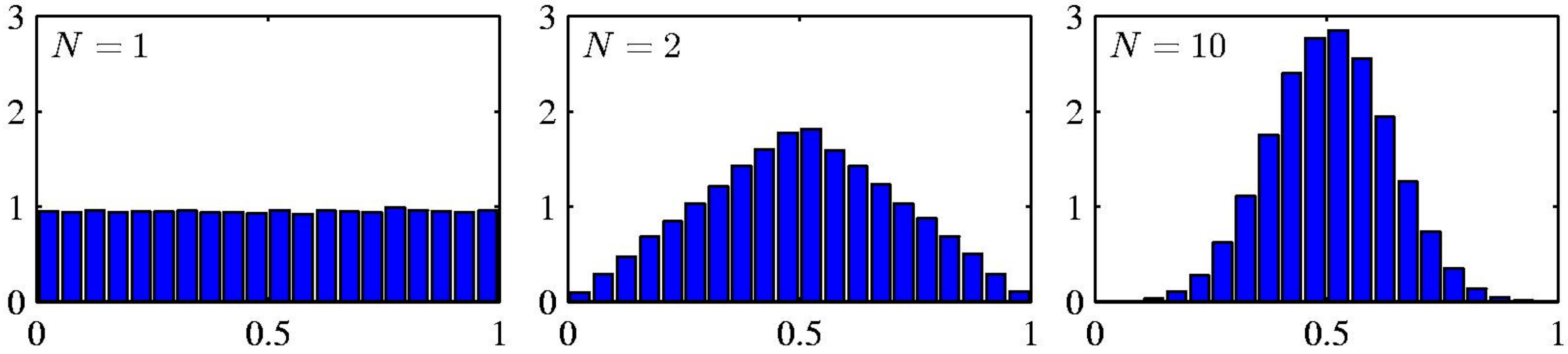
$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

$$\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right) \mathrm{d}x = 1$$

MATLAB Statistics Toolbox Function:
normpdf(x,mu,sigma)

Central Limit Theorem

- The distribution of the mean of N i.i.d. random variables becomes increasingly Gaussian as N grows.
- Example: N uniform [0,1] random variables.



Gaussian Mean and Variance

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu, \sigma^2\right) x \, \mathrm{d}x = \mu$$

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu, \sigma^2\right) x^2 \, \mathrm{d}x = \mu^2 + \sigma^2$$

$$\mathrm{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2$$

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. **Bayesian Classifiers**
4. Minimizing Risk
5. Nonparametric Density Estimation
6. Training and Evaluation Methods

Bayesian Classification

- Input feature vectors

$$\mathbf{x} = \left[x_1, x_2, \ldots, x_l \right]^T$$

- Assign the pattern represented by feature vector **x** to the **most probable** of the available classes

$$\omega_1, \omega_2, \ldots, \omega_M$$

That is, $\mathbf{x} \rightarrow \omega_i : P(\omega_i \mid \mathbf{x})$ is maximum.

Posterior

Bayesian Classification

- Computation of posterior probabilities
  - Assume known
    - Prior probabilities

$$P(\omega_1), P(\omega_2) \ldots, P(\omega_M)$$

    - Likelihoods

$$p\left(\mathbf{x} \mid \omega_i\right), \qquad i = 1, 2, \ldots, M$$

Bayes' Rule for Classification

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

where

$$p(\mathbf{x}) = \sum_{i=1}^{M} p(\mathbf{x} \mid \omega_i)\, p(\omega_i)$$

M=2 Classes

- Given **x** classify it according to the rule

$$\boxed{\begin{array}{l} \text{If } P(\omega_1|\mathbf{x}) > P(\omega_2|\mathbf{x}) \quad \rightarrow \omega_1 \\ \text{If } P(\omega_2|\mathbf{x}) > P(\omega_1|\mathbf{x}) \quad \rightarrow \omega_2 \end{array}}$$

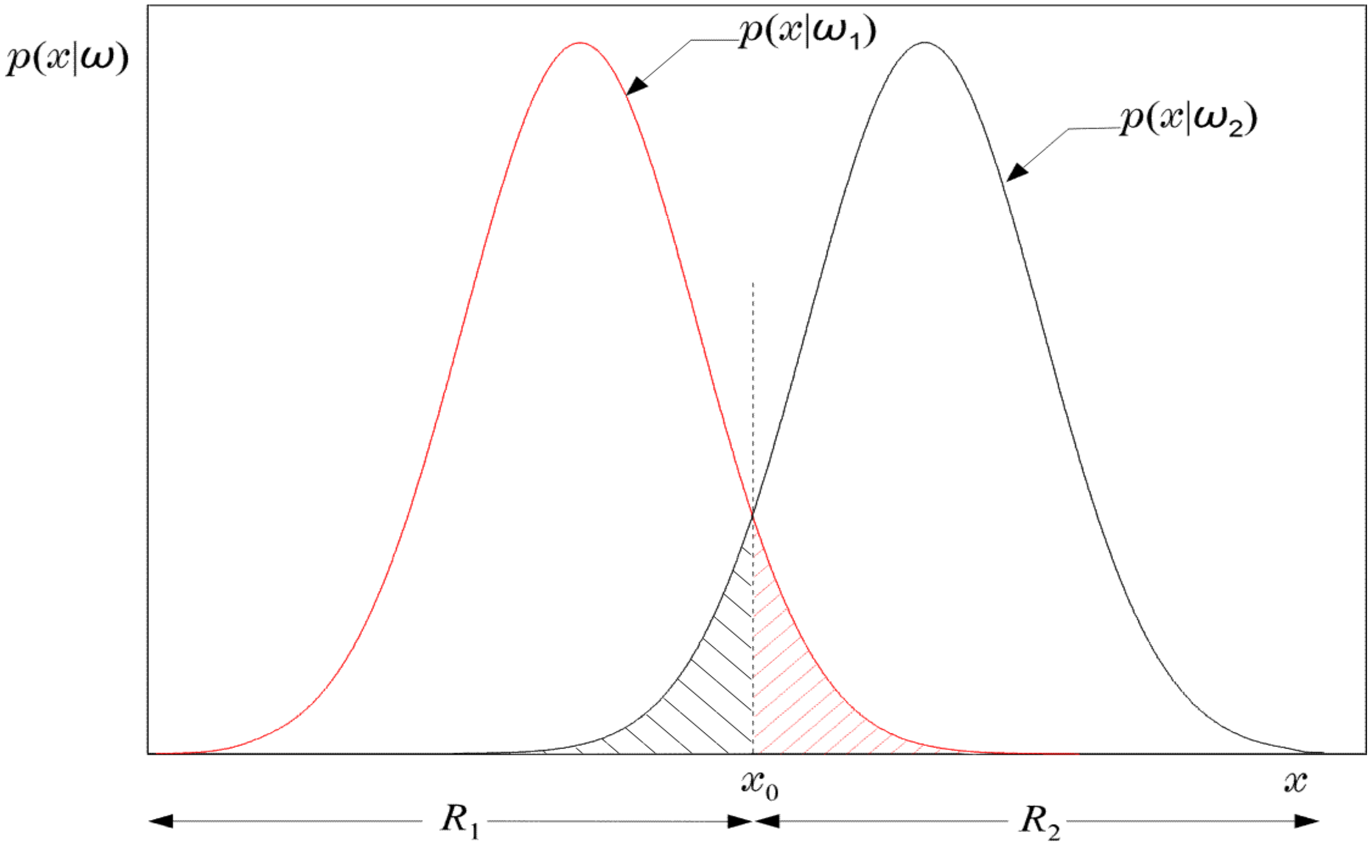
- Equivalently: classify **x** according to the rule

$$\boxed{\begin{array}{l} \text{If } \mathrm{p}\left(\mathbf{x}\,|\,\omega_1\right)P\left(\omega_1\right) > \mathrm{p}\left(\mathbf{x}\,|\,\omega_2\right)P\left(\omega_2\right) \rightarrow \omega_1 \\ \text{If } \mathrm{p}\left(\mathbf{x}\,|\,\omega_2\right)P\left(\omega_2\right) > \mathrm{p}\left(\mathbf{x}\,|\,\omega_1\right)P\left(\omega_1\right) \rightarrow \omega_2 \end{array}}$$

- For equiprobable classes the test becomes

$$\boxed{\begin{array}{l} \text{If } \mathrm{p}\left(\mathbf{x}\,|\,\omega_1\right) > \mathrm{p}\left(\mathbf{x}\,|\,\omega_2\right) \rightarrow \omega_1 \\ \text{If } \mathrm{p}\left(\mathbf{x}\,|\,\omega_2\right) > \mathrm{p}\left(\mathbf{x}\,|\,\omega_1\right) \rightarrow \omega_2 \end{array}}$$

Example: Equiprobable Classes



$$R_1(\rightarrow \omega_1) \text{ and } R_2(\rightarrow \omega_2)$$

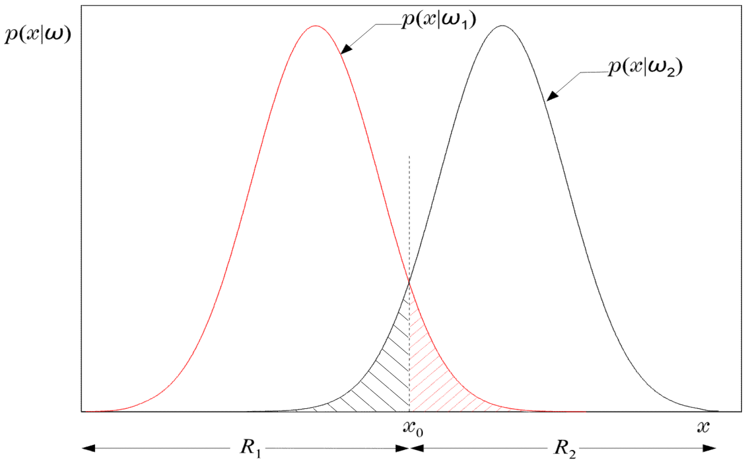
Example: Equiprobable Classes

- Probability of error
  - The black and red shaded areas represent

$$P\left(\text{error} \mid \omega_2\right) = \int_{-\infty}^{x_0} p(x|\omega_2)dx \quad \text{and} \quad P\left(\text{error} \mid \omega_1\right) = \int_{x_0}^{\infty} p(x|\omega_1)dx$$

  - Thus

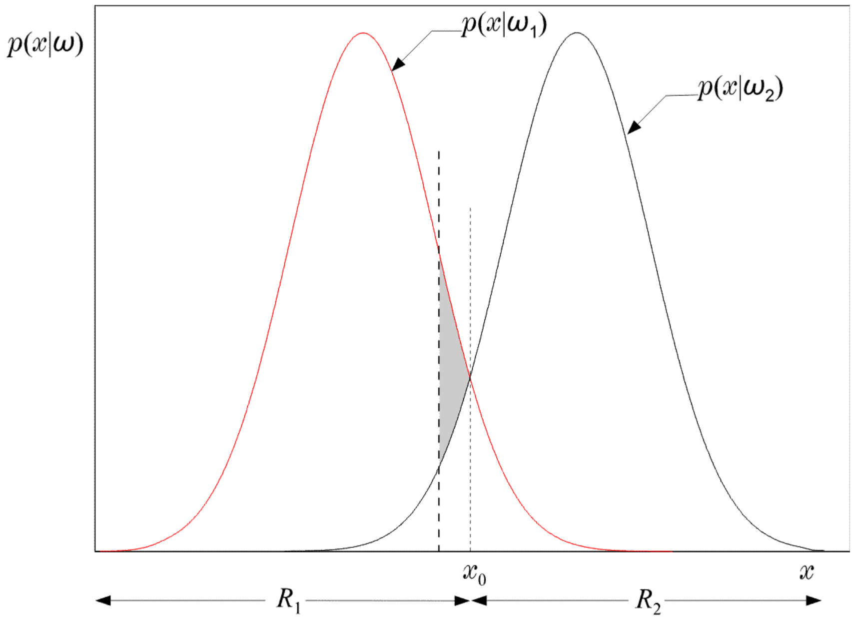
$$\begin{aligned} P_e &\triangleq P(\text{error}) \\ &= P\left(\omega_2\right)P\left(\text{error}|\omega_2\right) + P\left(\omega_1\right)P\left(\text{error}|\omega_1\right) \\ &= \frac{1}{2}\int_{-\infty}^{x_0} p(x|\omega_2)dx + \frac{1}{2}\int_{x_0}^{+\infty} p(x|\omega_1)dx \end{aligned}$$



- **Bayesian classifier is OPTIMAL: it minimizes the classification error probability**

Example: Equiprobable Classes

- To see this, observe that shifting the threshold increases the error rate for one class of patterns more than it decreases the error rate for the other class.



The General Case

- In general, for M classes and unequal priors, the decision rule

$$\boxed{P(\omega_i \mid \mathbf{x}) > P(\omega_j \mid \mathbf{x}) \;\; \forall j \neq i \quad \rightarrow \omega_i}$$

minimizes the expected error rate.

Types of Error

- Minimizing the expected error rate is a pretty reasonable goal.
- However, it is not always the best thing to do.
- Example:
  - You are designing a pedestrian detection algorithm for an autonomous navigation system.
  - Your algorithm must decide whether there is a pedestrian crossing the street.
  - There are two possible types of error:
    - False positive: there is no pedestrian, but the system thinks there is.
    - Miss: there is a pedestrian, but the system thinks there is not.
  - Should you give equal weight to these 2 types of error?

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. **Minimizing Risk**
5. Nonparametric Density Estimation
6. Training and Evaluation Methods

Topic 4. Minimizing Risk

The Loss Matrix

- To deal with this problem, instead of minimizing error rate, we minimize something called the **risk**.
- First, we define the **loss matrix L**, which quantifies the cost of making each type of error.
- Element λ_{ij} of the loss matrix specifies the cost of deciding class *j* when in fact the input is of class *i*.
- Typically, we set $\lambda_{ii}=0$ for all i.
- Thus a typical loss matrix for the M = 2 case would have the form

$$L = \begin{bmatrix} 0 & \lambda_{12} \\ \lambda_{21} & 0 \end{bmatrix}$$

Risk

- Given a loss function, we can now define the risk associated with each class *k* as:

$$r_k = \sum_{i=1}^{M} \lambda_{ki} \underbrace{\int_{R_i} p\left(\mathbf{x} \mid \omega_k\right) d\mathbf{x}}_{\text{Probability we will decide Class } \omega_i \text{ given pattern from Class } \omega_k}$$

- where R_i is the region of the input space where we will decide ω_i.

Minimizing Risk

- Now the goal is to minimize the expected risk ***r***, given by

$$r = \sum_{k=1}^{M} r_k P\left(\omega_k\right)$$

Minimizing Risk

$$r = \sum_{k=1}^{M} r_k P(\omega_k) \quad \text{where} \quad r_k = \sum_{i=1}^{M} \lambda_{ki} \int_{R_i} p(\mathbf{x} \mid \omega_k) d\mathbf{x}$$

- We need to select the decision regions R_i to minimize the risk r.
- Note that the set of R_i are disjoint and exhaustive.
- Thus we can minimize the risk by ensuring that each input $\mathbf{x}$ falls in the region R_i that minimizes the expected loss for that particular input, i.e.,

$$\text{Letting } l_i = \sum_{k=1}^{M} \lambda_{ki} p(\mathbf{x} \mid \omega_k) P(\omega_k),$$

we select the partioning regions such that

$$\mathbf{x} \in R_i \text{ if } l_i < l_j \quad \forall j \neq i$$

Example: M=2

- For the 2-class case:

$$l_1 = \lambda_{11} p(\mathbf{x} \mid \omega_1) P(\omega_1) + \lambda_{21} p(\mathbf{x} \mid \omega_2) P(\omega_2)$$

and

$$l_2 = \lambda_{12} p(\mathbf{x} \mid \omega_1) P(\omega_1) + \lambda_{22} p(\mathbf{x} \mid \omega_2) P(\omega_2)$$

- Thus we assign **x** to ω_1 if

$$(\lambda_{21} - \lambda_{22}) p(\mathbf{x} \mid \omega_2) P(\omega_2) < (\lambda_{12} - \lambda_{11}) p(\mathbf{x} \mid \omega_1) P(\omega_1)$$

- i.e., if

Likelihood Ratio Test

$$\frac{p(\mathbf{x} \mid \omega_1)}{p(\mathbf{x} \mid \omega_2)} > \frac{P(\omega_2)(\lambda_{21} - \lambda_{22})}{P(\omega_1)(\lambda_{12} - \lambda_{11})}.$$

Likelihood Ratio Test

$$\frac{p(\mathbf{x} \mid \omega_1)}{p(\mathbf{x} \mid \omega_2)} \overset{?}{>} \frac{P(\omega_2)(\lambda_{21} - \lambda_{22})}{P(\omega_1)(\lambda_{12} - \lambda_{11})}.$$

- Typically, the loss for a correct decision is 0. Thus the likelihood ratio test becomes

$$\frac{p(\mathbf{x} \mid \omega_1)}{p(\mathbf{x} \mid \omega_2)} \overset{?}{>} \frac{P(\omega_2)\lambda_{21}}{P(\omega_1)\lambda_{12}}.$$

- In the case of equal priors and equal loss functions, the test reduces to

$$\frac{p(\mathbf{x} \mid \omega_1)}{p(\mathbf{x} \mid \omega_2)} \overset{?}{>} 1.$$

Example

- Consider a one-dimensional input space, with features generated by normal distributions with identical variance:

$$p(x|\omega_1) \sim N\left(\mu_1, \sigma^2\right)$$
$$p(x|\omega_2) \sim N\left(\mu_2, \sigma^2\right)$$

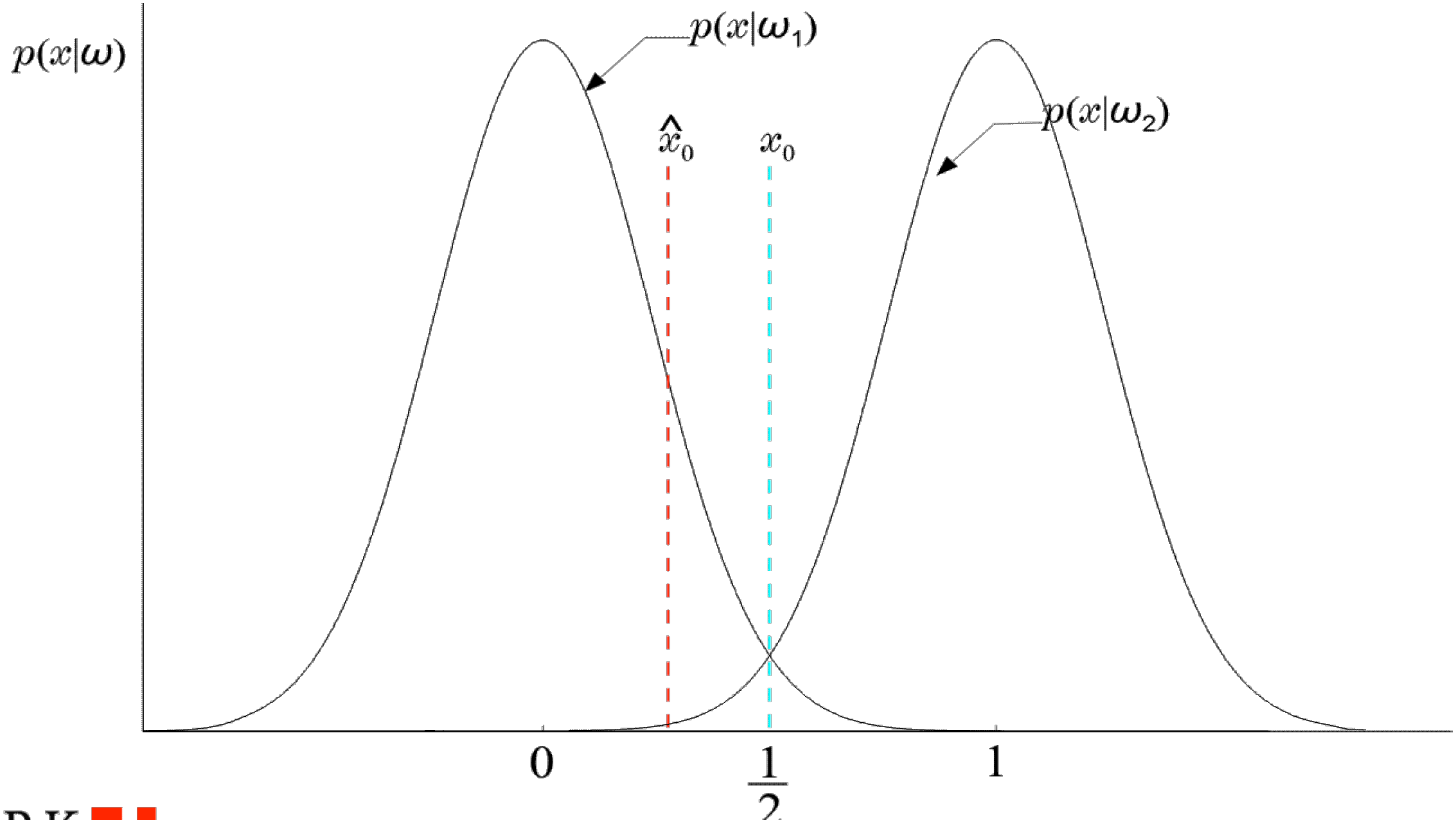
where $\mu_1 = 0, \mu_2 = 1, \text{and } \sigma^2 = \frac{1}{2}$

- Let's assume equiprobable classes, and higher loss for errors on Class 2, specifically:

$$\lambda_{21} = 1,\ \lambda_{12} = \frac{1}{2}.$$

Results

- The threshold has shifted to the left – why?



End of Lecture

Sept 12, 2012

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. **Nonparametric Density Estimation**
6. Training and Evaluation Methods

Nonparametric Methods

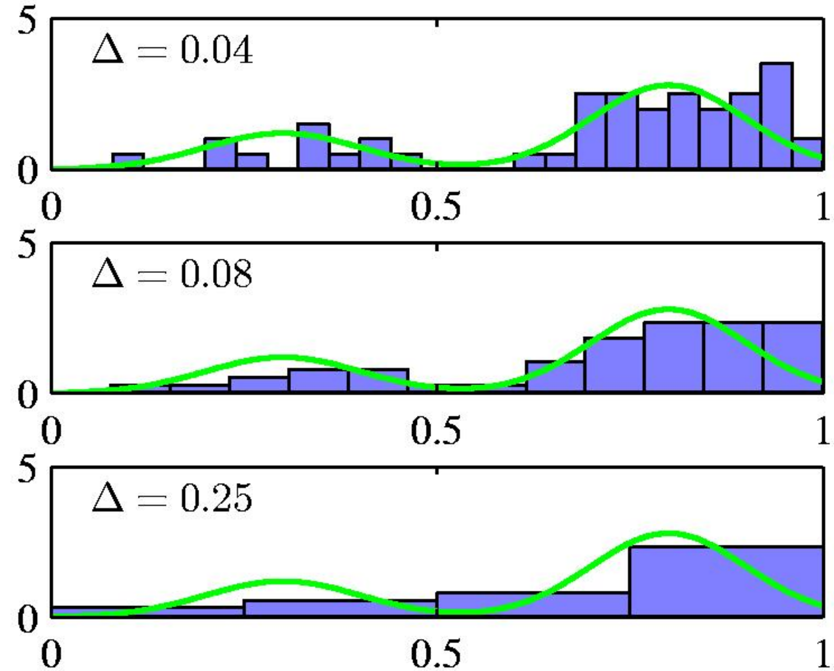
- Parametric distribution models are restricted to specific forms, which may not always be suitable; for example, consider modelling a multimodal distribution with a single, unimodal model.
- You can use a mixture model, but then you have to decide on the number of components, and hope that your parameter estimation algorithm (e.g., EM) converges to a global optimum!
- Nonparametric approaches make few assumptions about the overall shape of the distribution being modelled, and in some cases may be simpler than using a mixture model.

Histogramming

- **Histogram methods** partition the data space into distinct bins with widths Δ_i and count the number of observations, n_i, in each bin.

$$p_i = \frac{n_i}{N\Delta_i}$$

- Often, the same width is used for all bins, $\Delta_i = \Delta$.
- Δ acts as a smoothing parameter.



- In a D-dimensional space, using M bins in each dimension will require M^D bins!

The curse of dimensionality

Kernel Density Estimation

- Assume observations drawn from a density *p(x)* and consider a small region *R* containing *x* such that

$$P = \int_{\mathcal{R}} p(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

- The expected number *K* out of *N* observations that will lie inside *R* is given by

$$K \simeq NP.$$

- If the volume *V* of *R* is sufficiently small, *p(x)* is approximately constant over *R* and

$$P \simeq p(\mathbf{x})V$$

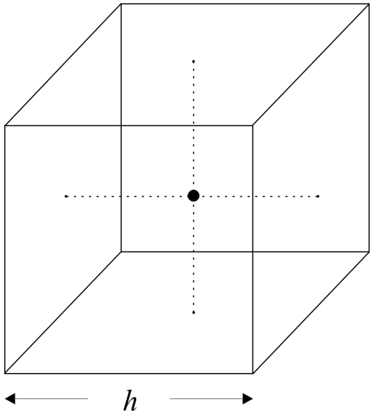
- Thus $$\boxed{p(\mathbf{x}) = \frac{K}{NV}.}$$

Kernel Density Estimation

Kernel Density Estimation: fix V, estimate K from the data. Let R be a hypercube centred on x and define the kernel function (Parzen window)

$$\boxed{p(\mathbf{x}) = \frac{K}{NV}.}$$

$$k((\mathbf{x}-\mathbf{x}_n)/h) = \begin{cases} 1, & |(x_i - x_{ni})/h| \leqslant 1/2, \qquad i = 1, \ldots, D, \\ 0, & \text{otherwise.} \end{cases}$$



It follows that

$$K = \sum_{n=1}^{N} k\left(\frac{\mathbf{x}-\mathbf{x}_n}{h}\right)$$

and hence

$$p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{h^D} k\left(\frac{\mathbf{x}-\mathbf{x}_n}{h}\right).$$

Kernel Density Estimation

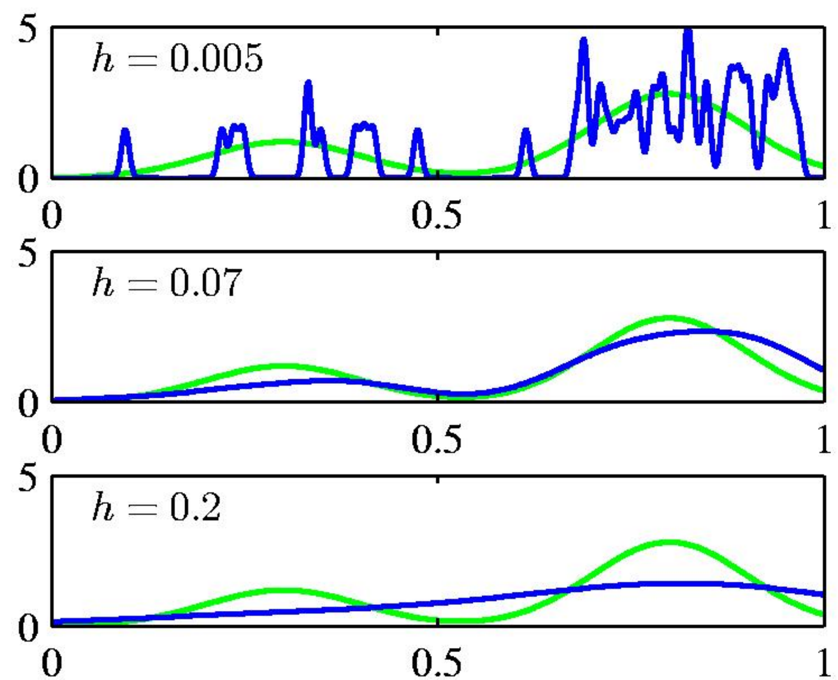
To avoid discontinuities in p(x), use a smooth kernel, e.g. a Gaussian

$$p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{(2\pi h^2)^{D/2}} \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}_n\|^2}{2h^2}\right\}$$

(Any kernel *k(u)* such that

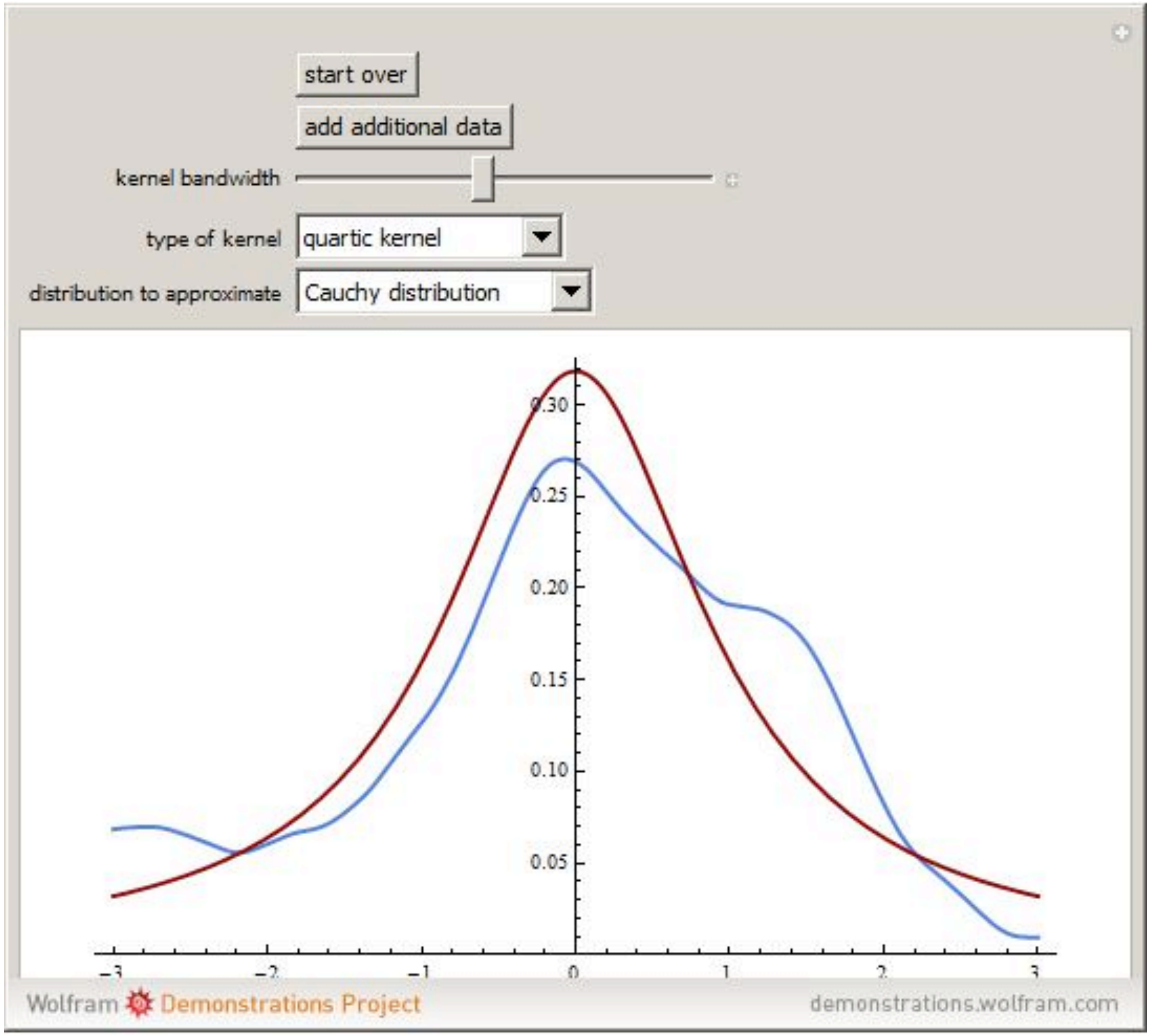
$$\begin{aligned} k(\mathbf{u}) &\geqslant 0, \\ \int k(\mathbf{u})\,\mathrm{d}\mathbf{u} &= 1 \end{aligned}$$

will work.)



h acts as a smoother.

KDE Example


start over
add additional data
kernel bandwidth
type of kernel
quartic kernel
distribution to approximate
Cauchy distribution
0.30
0.25
0.20
0.15
0.10
0.05
−3
−2
−1
0
1
2
3
Wolfram Demonstrations Project
demonstrations.wolfram.com

Kernel Density Estimation

- Problem: if V is fixed, there may be too few points in some regions to get an accurate estimate.

Nearest Neighbour Density Estimation

Nearest Neighbour Density Estimation: fix *K*, estimate *V* from the data. Consider a hypersphere centred on *x* and let it grow to a volume *V** that includes *K* of the given *N* data points. Then

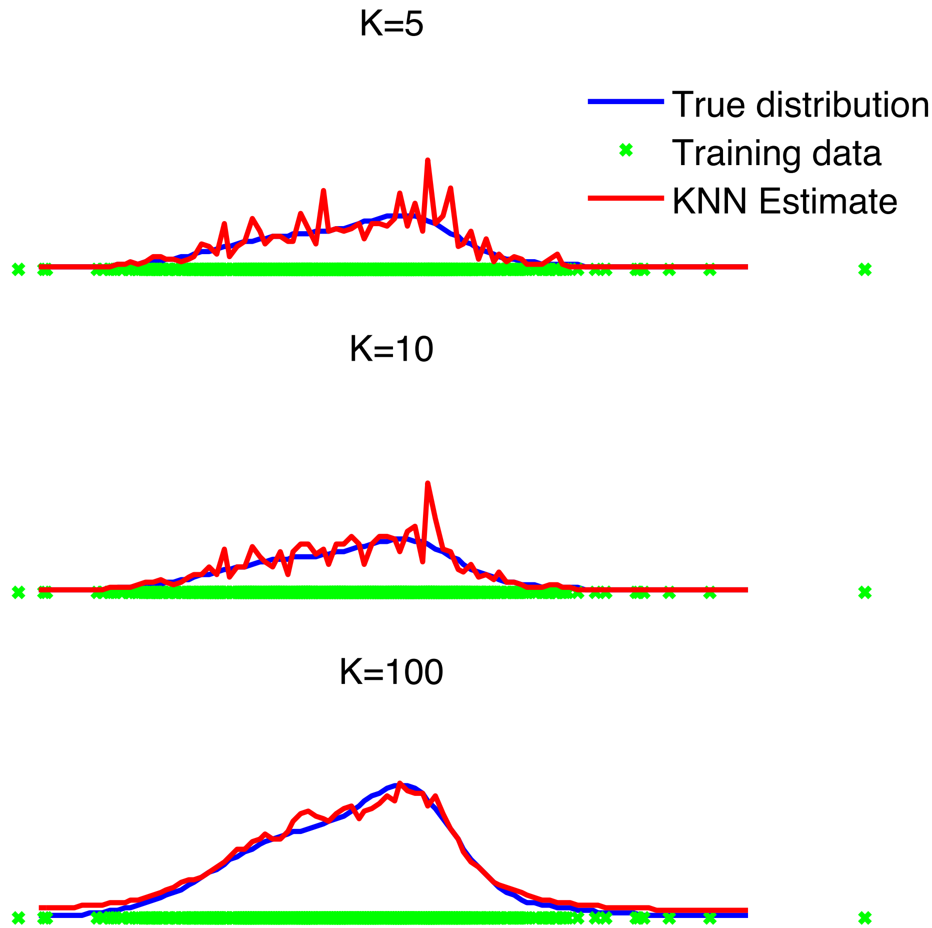
$$p(\mathbf{x}) \simeq \frac{K}{NV^\star}.$$

```
for j=1:np
    d=sort(abs(x(j)-xi));
    V=2*d(K(i));
    phat(j)=K(i)/(N*V);
end
```

Nearest Neighbour Density Estimation

Nearest Neighbour Density Estimation: fix *K*, estimate *V* from the data. Consider a hypersphere centred on *x* and let it grow to a volume *V** that includes *K* of the given *N* data points. Then

$$p(\mathbf{x}) \simeq \frac{K}{NV^\star}.$$



Nearest Neighbour Density Estimation

- Problem: does not generate a proper density (for example, integral is unbounded on $\mathbb{R}^D$)
- In practice, on finite domains, can normalize.
- But makes strong assumption on tails $\left(\propto \frac{1}{x} \right)$

Nonparametric Methods

- Nonparametric models (not histograms) require storing and computing with the entire data set.
- Parametric models, once fitted, are much more efficient in terms of storage and computation.

K-Nearest-Neighbours for Classification

- Given a data set with N_k data points from class C_k and $\sum_k N_k = N$, we have

$$p(\mathbf{x}) = \frac{K}{NV}$$

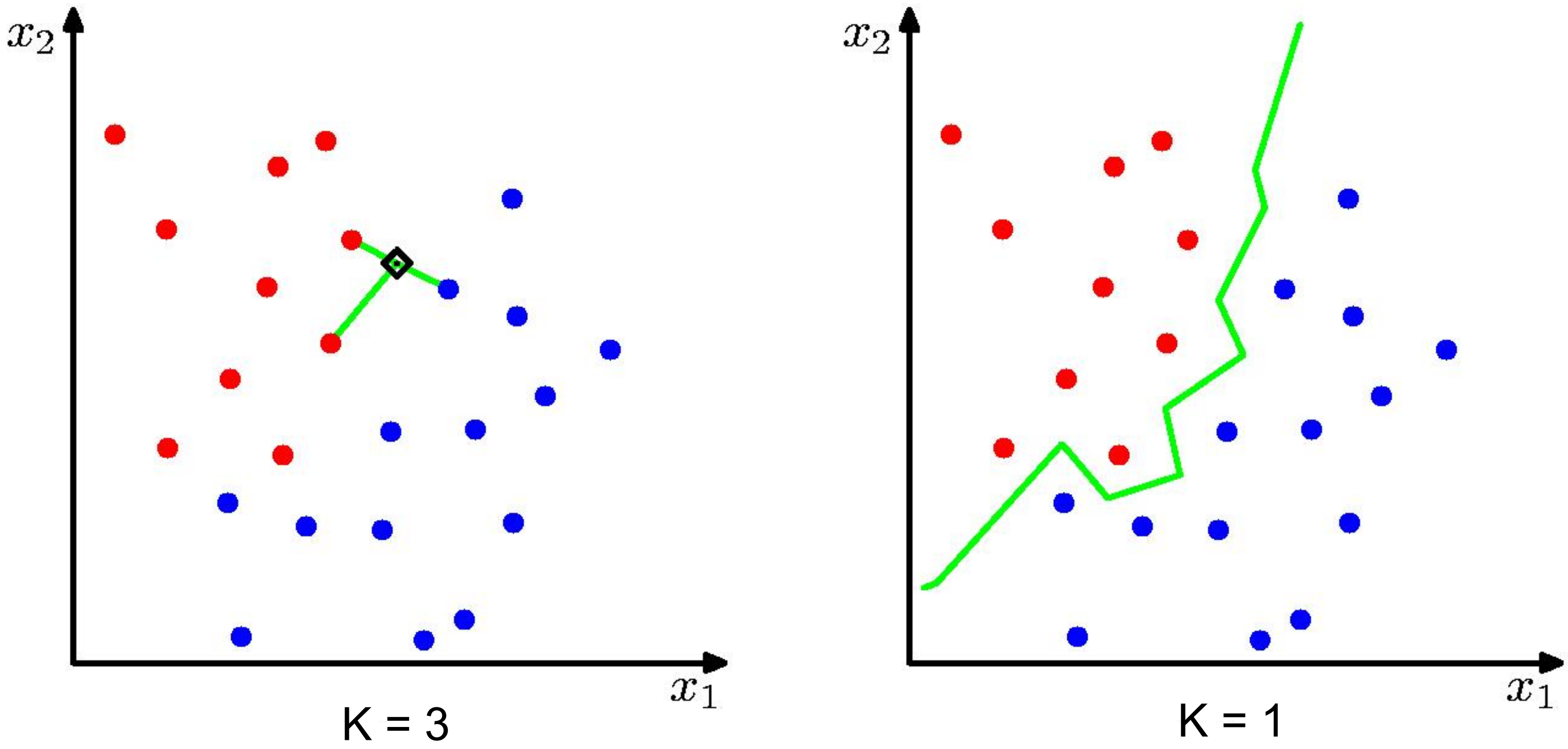
- and correspondingly

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{K_k}{N_k V}.$$

- Since $p(\mathcal{C}_k) = N_k/N$, Bayes' theorem gives

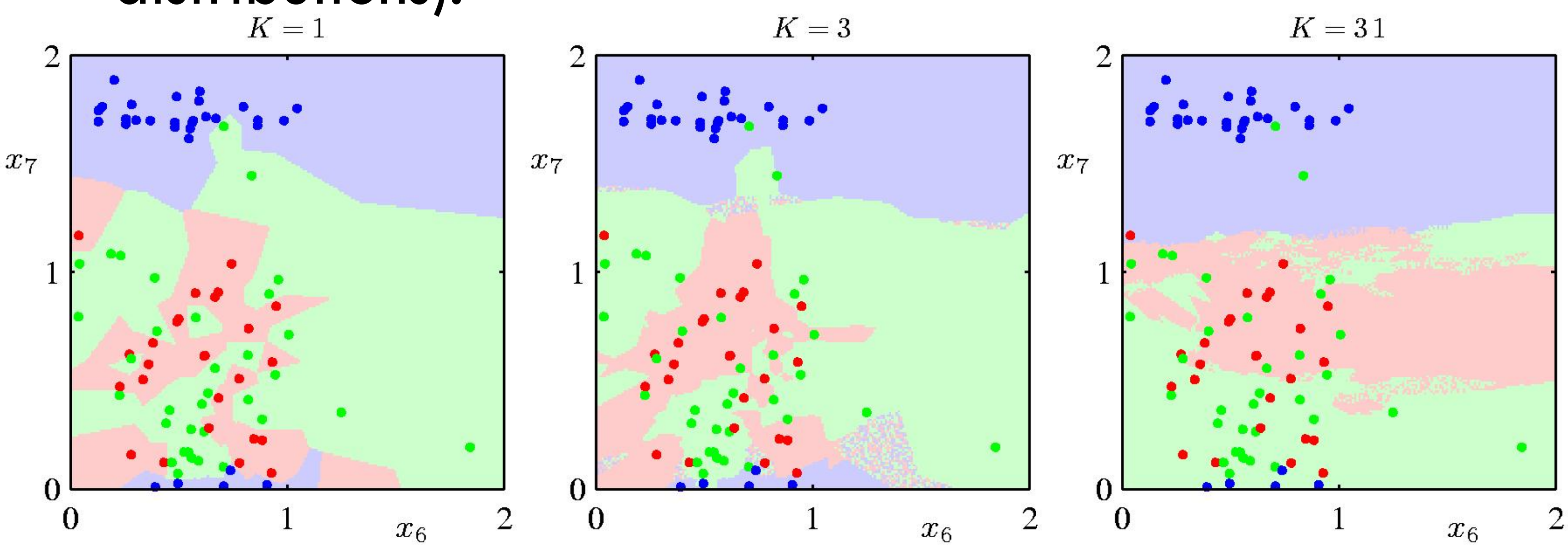
$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})} = \frac{K_k}{K}.$$

K-Nearest-Neighbours for Classification

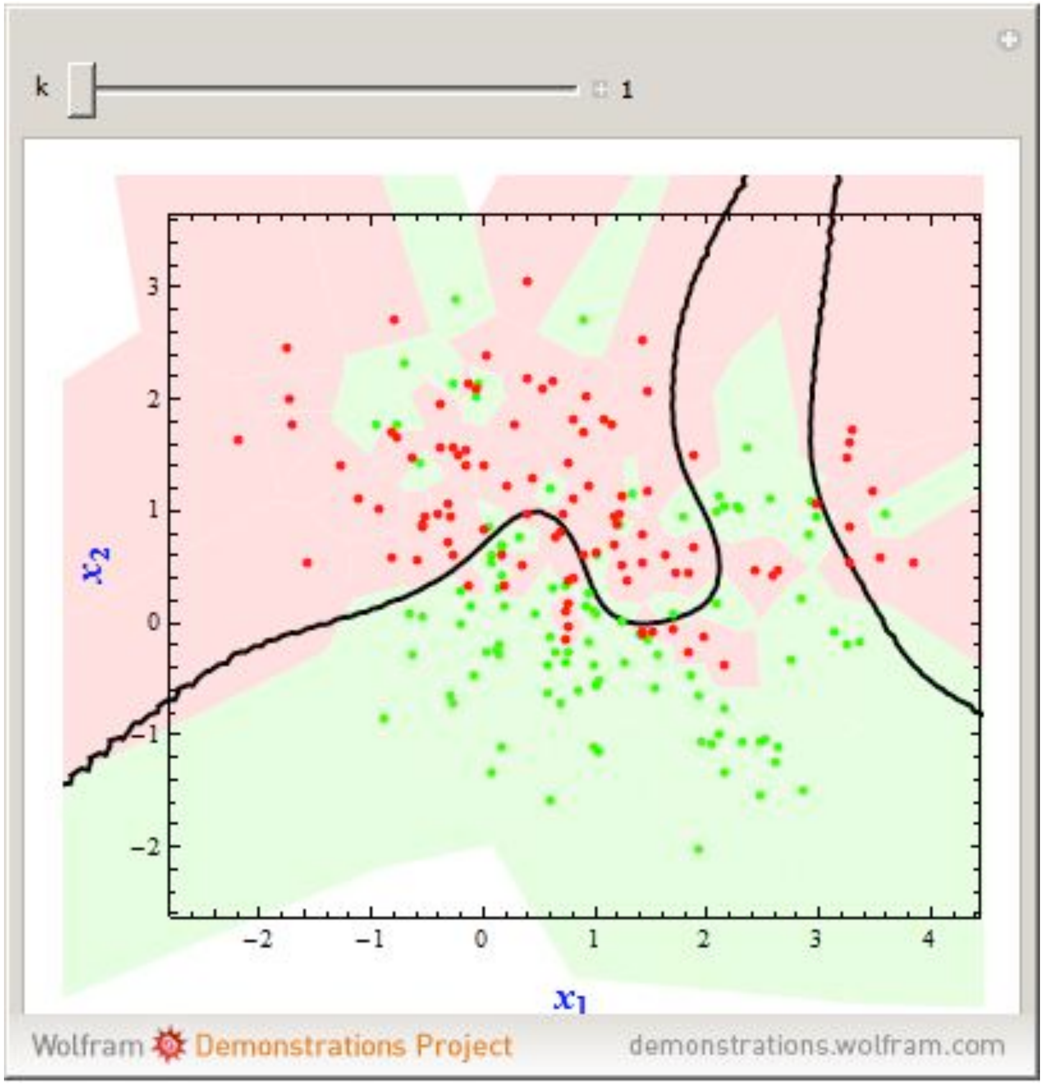


K-Nearest-Neighbours for Classification

- K acts as a smoother
- As $N \to \infty$, the error rate of the 1-nearest-neighbour classifier is never more than twice the optimal error (obtained from the true conditional class distributions).



KNN Example


k
1
3
2
1
0
−1
−2
x_2
−2
−1
0
1
2
3
4
x_1
Wolfram Demonstrations Project
demonstrations.wolfram.com

Naïve Bayes Classifiers

- All of these nonparametric methods require lots of data to work. If $\mathcal{O}(N)$ training points are required for accurate estimation in 1 dimension, then $\mathcal{O}(N^D)$ points are required for D-dimensional input vectors.
- It may sometimes be possible to assume that the individual dimensions of the feature vector are conditionally independent. Then we have

$$p(\underline{x} \mid \omega_i) = \prod_{j=1}^{D} p(x_j \mid \omega_i)$$

- This reduces the data requirements to $\mathcal{O}(DN)$.

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. Nonparametric Density Estimation
6. **Training and Evaluation Methods**

Machine Learning System Design

- The process of solving a particular classification or regression problem typically involves the following sequence of steps:
  1. **Design and code** promising candidate systems
  2. **Train** each of the candidate systems (i.e., learn the parameters)
  3. **Evaluate** each of the candidate systems
  4. **Select and deploy** the best of these candidate systems

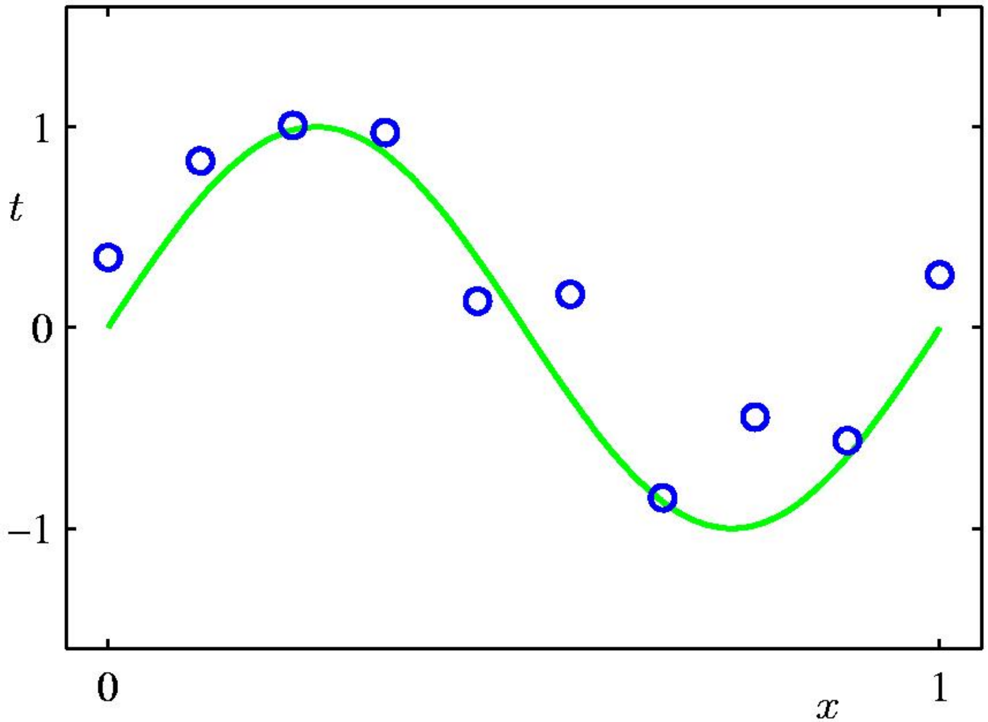
Using Your Training Data

- You will always have a finite amount of data on which to train and evaluate your systems.
- The performance of a classification system is often **data-limited**: if we only had more data, we could make the system better.
- Thus it is important to use your finite data set wisely.

Overfitting

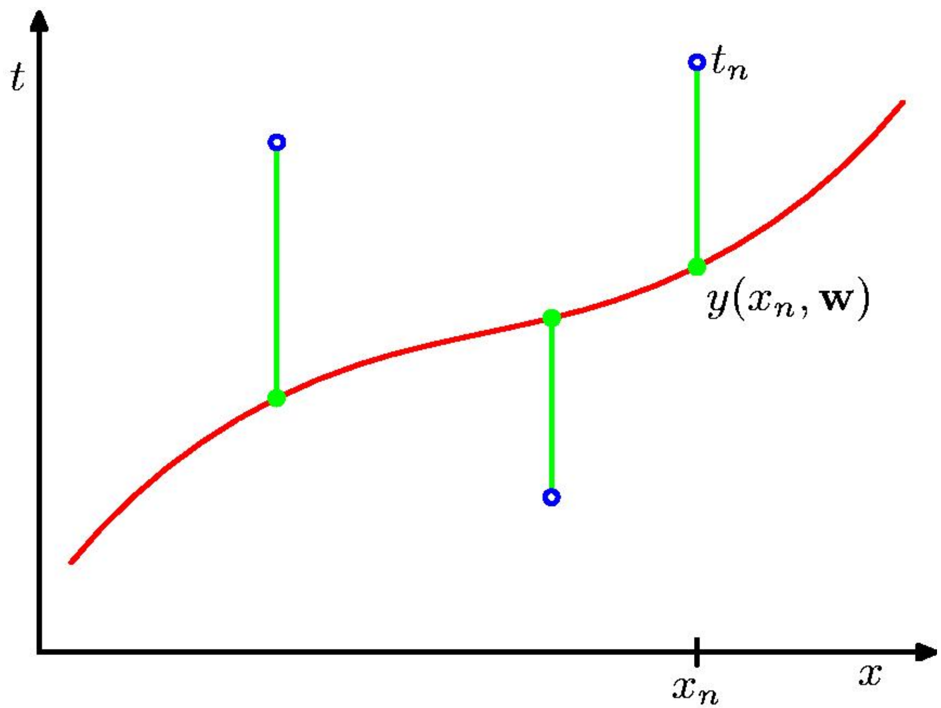
- Given that learning is often data-limited, it is tempting to use all of your data to estimate the parameters of your models, and then select the model with the lowest error on your training data.
- Unfortunately, this leads to a notorious problem called **over-fitting**.

Example: Polynomial Curve Fitting



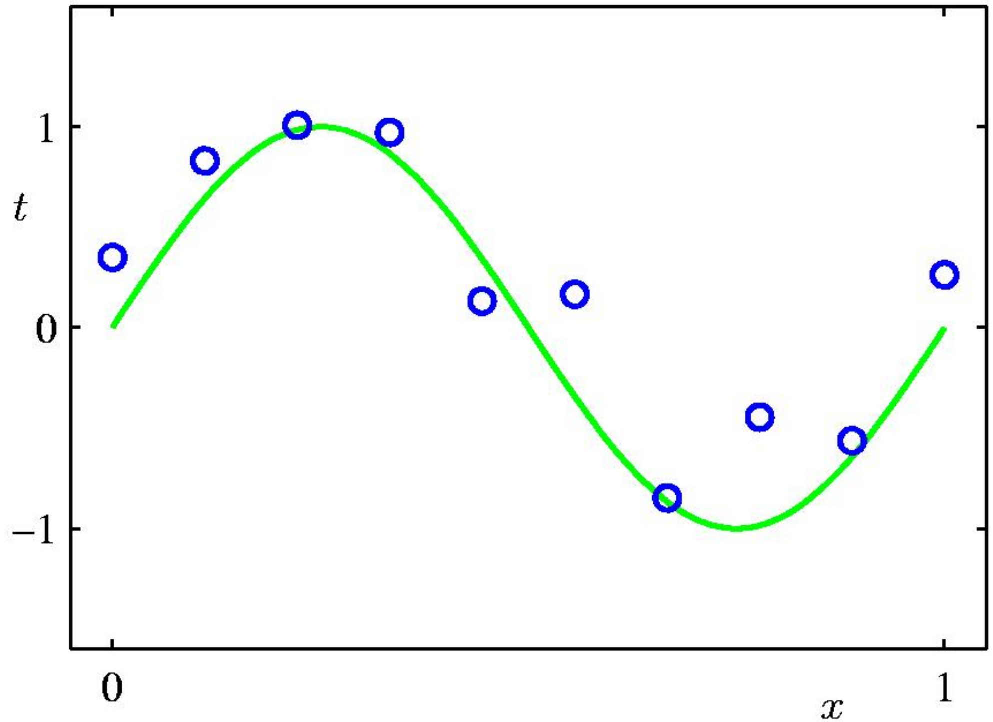
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Sum-of-Squares Error Function



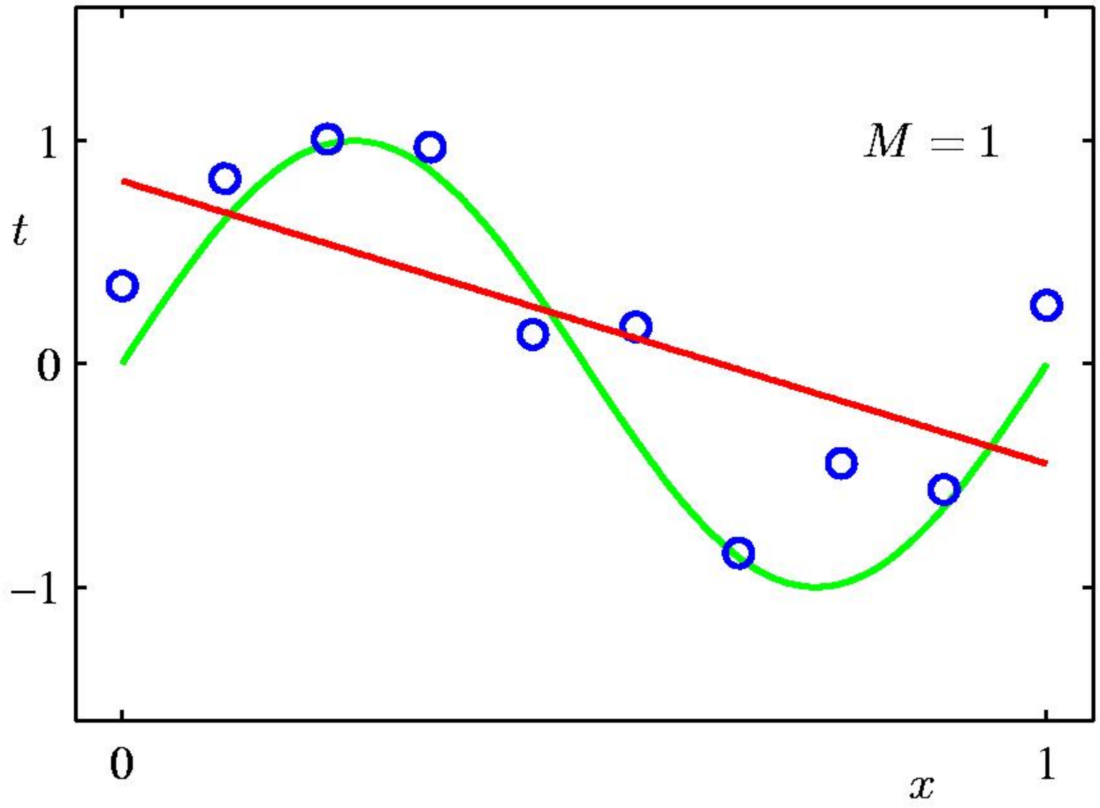
$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

How do we choose *M*, the order of the model?

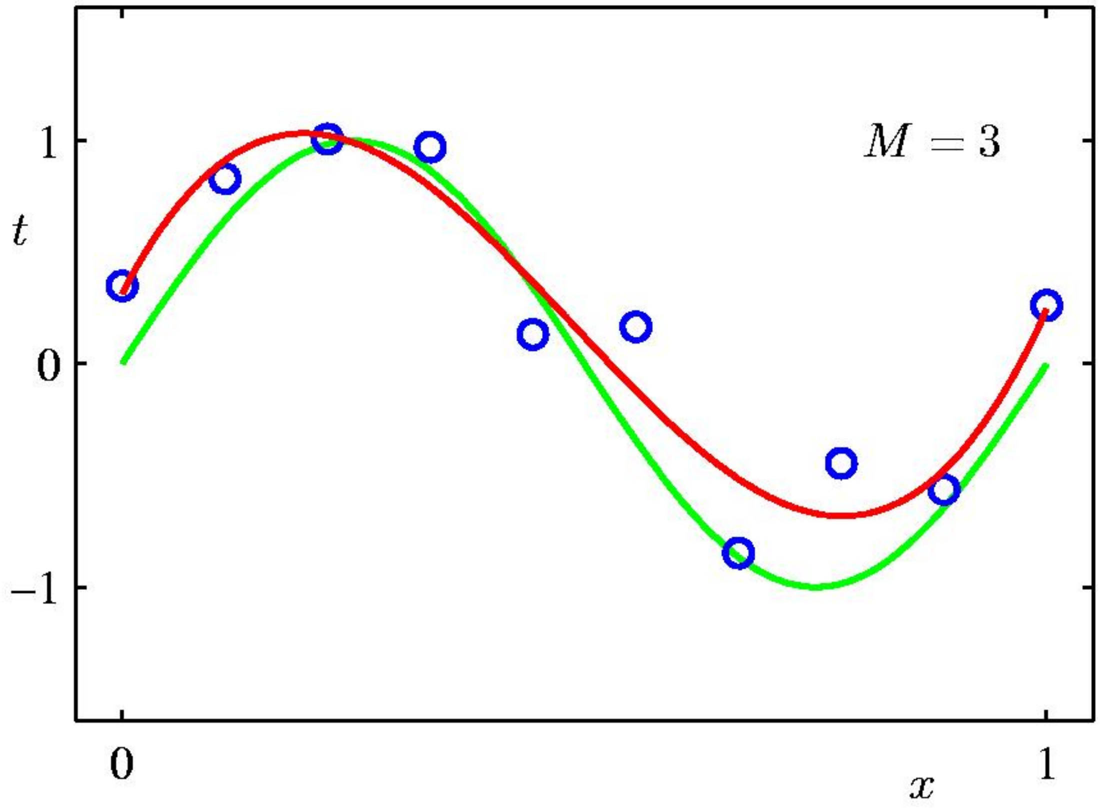


$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

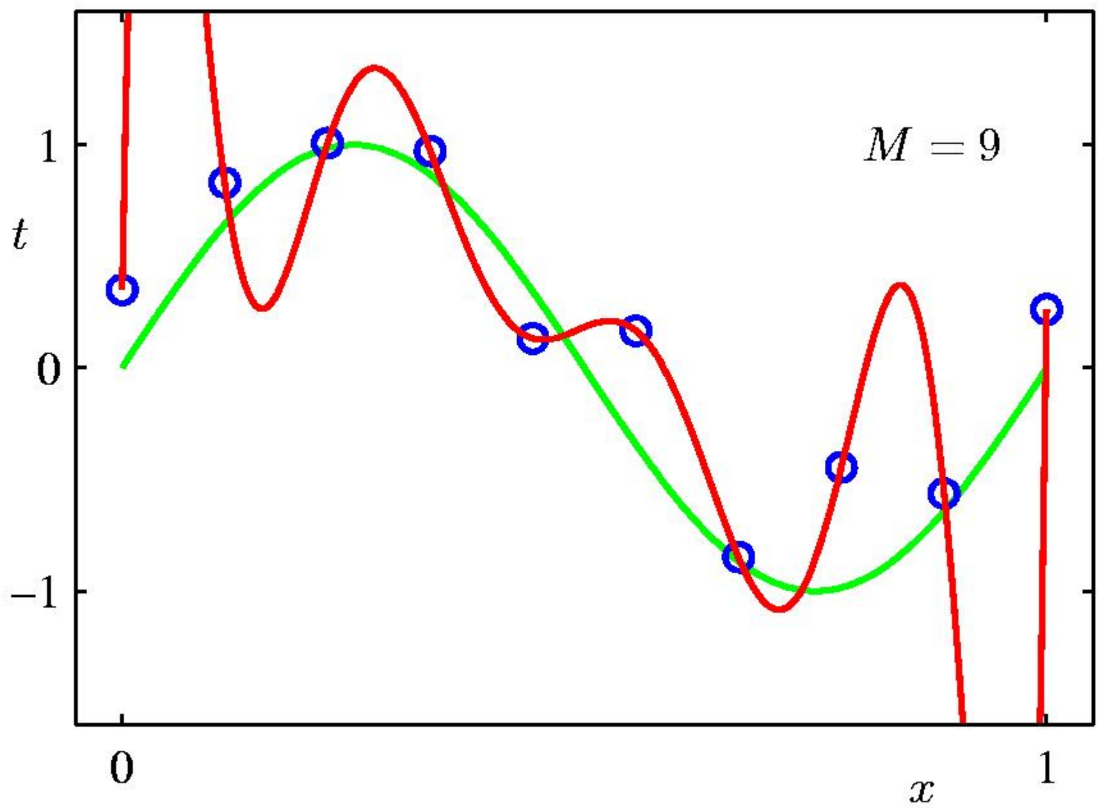
1 st Order Polynomial



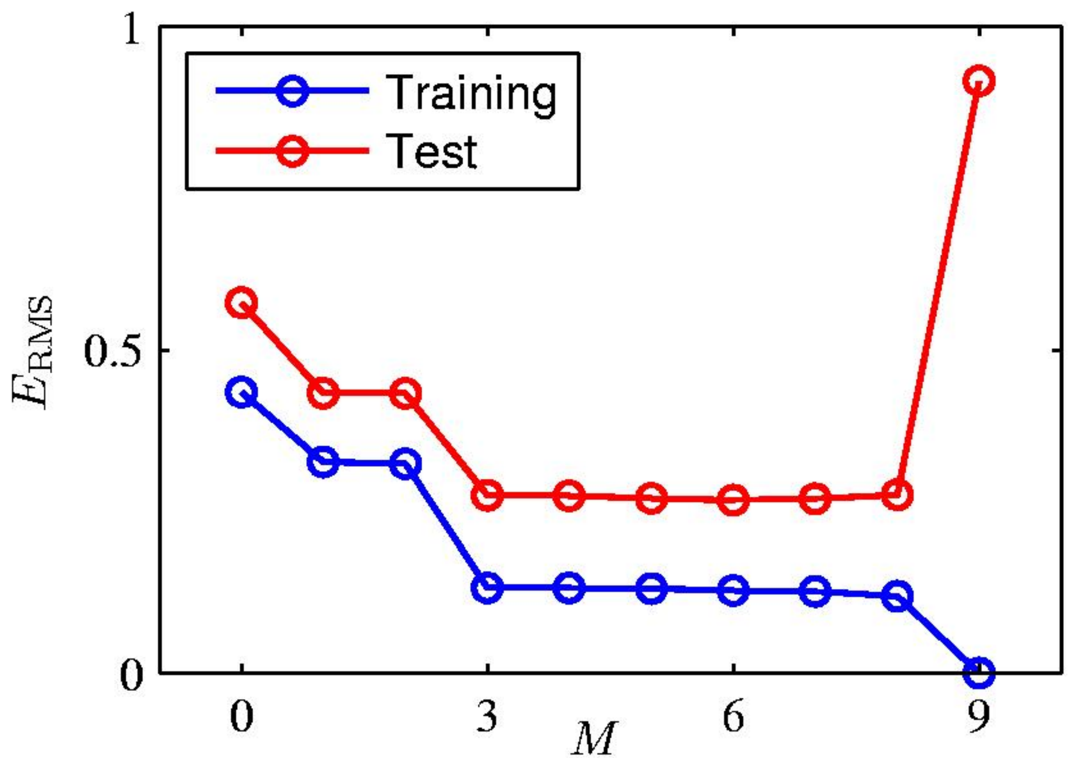
3rd Order Polynomial



9^{th} Order Polynomial



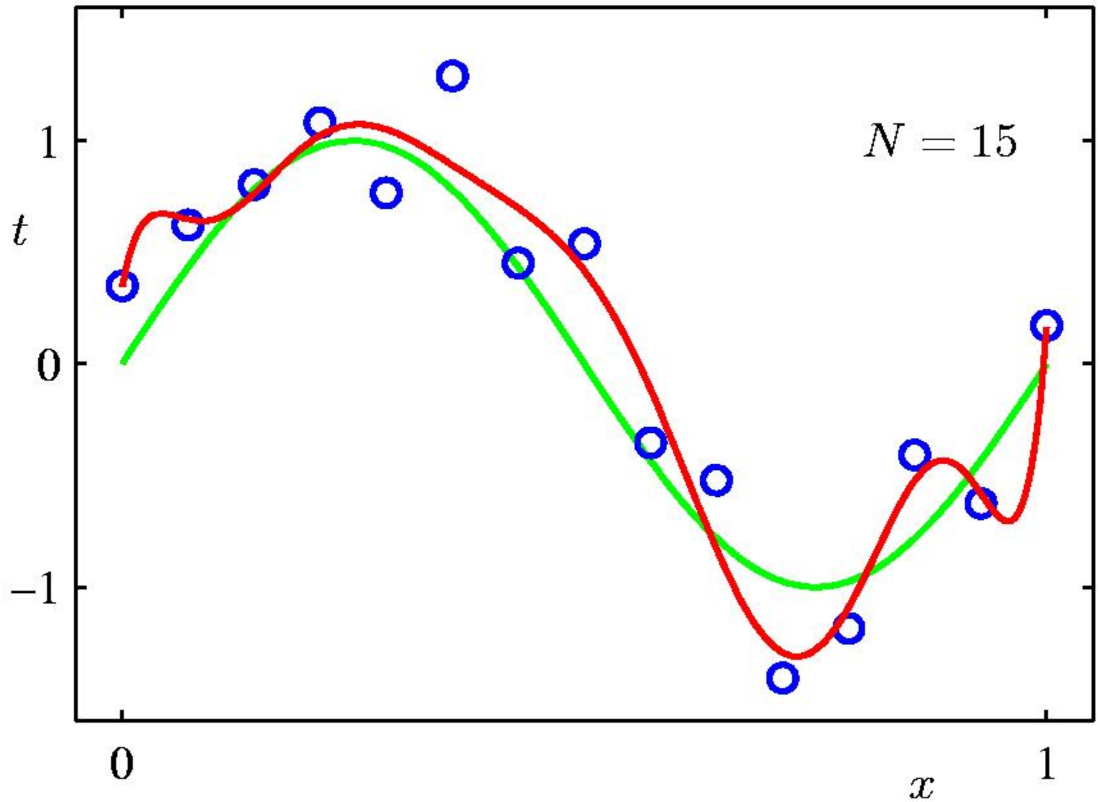
Over-fitting



Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$

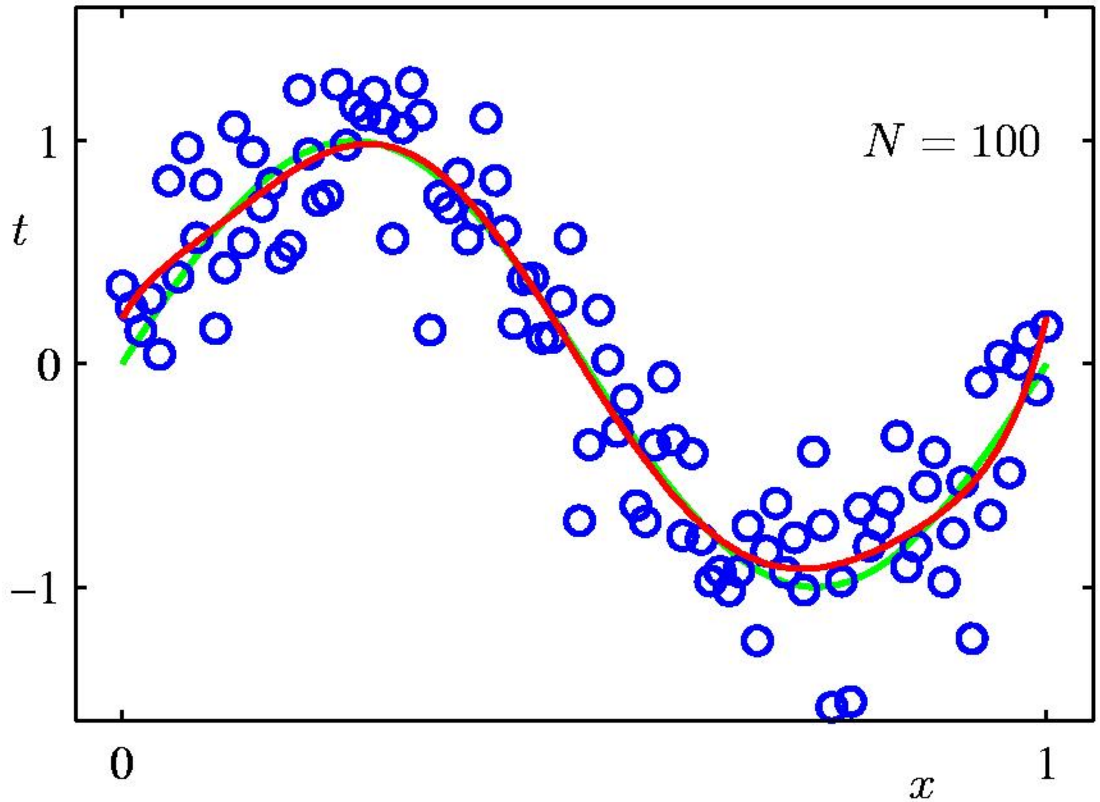
Overfitting and Sample Size

9th Order Polynomial



Over-fitting and Sample Size

9^{th} Order Polynomial

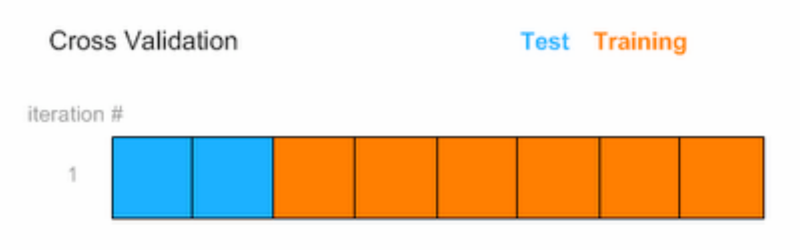


Methods for Preventing Over-Fitting

- Bayesian parameter estimation
  - Application of prior knowledge regarding the probable values of unknown parameters can often limit over-fitting of a model
- Model selection criteria
  - Methods exist for comparing models of differing complexity (i.e., with different types and numbers of parameters)
    - Bayesian Information Criterion (BIC)
    - Akaike Information Criterion (AIC)
- Cross-validation
  - This is a very simple method that is universally applicable.

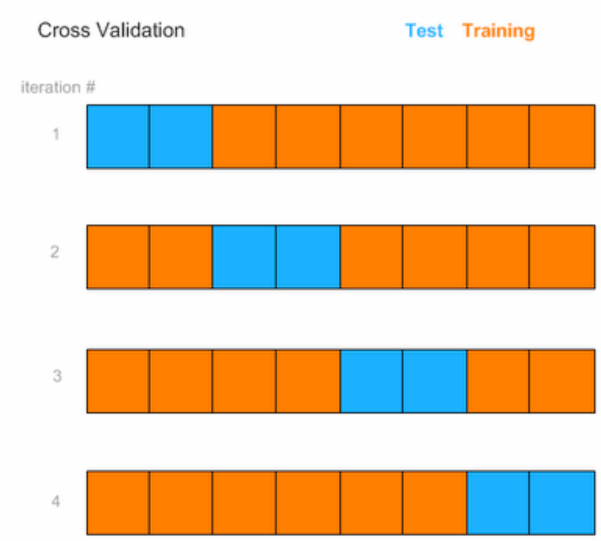
Cross-Validation

- The available data are partitioned into disjoint training and test subsets.
- Parameters are learned on the training sets.
- Performance of the model is then evaluated on the test set.
- Since the test set is independent of the training set, the evaluation is fair: models that overlearn the noise in the training set will perform poorly on the test set.



Cross-Validation: Choosing the Partition

- What is the best way to partition the data?
  - A larger training set will lead to more accurate parameter estimation.
  - However a small test set will lead to a noisy performance score.
  - If you can afford the computation time, repeat the training/test cycle on complementary partitions and then average the results. This gives you the best of all worlds: accurate parameter estimation and accurate evaluation.
  - In the limit: the **leave-one-out method**



A useful MATLAB function

- randperm(n)
  - Generates a random permutation of the integers from 1 to n
  - The result can be used to select random subsets from your data

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. Nonparametric Density Estimation
6. Training and Evaluation Methods