Pattern Recognition 2015
Introduction

Ad Feelders

Universiteit Utrecht

About the Course

- Lecturers:
  - Statistical Pattern Recognition: Ad Feelders
  - Geometrical Pattern Recognition: Marc van Kreveld
- Required Literature
  - Statistical Pattern Recognition:
    1. Christopher M. Bishop, Pattern Recognition and Machine Learning.
    2. Gareth James et al., An Introduction to Statistical Learning.
    3. Lecture Slides.
  - Geometrical Pattern Recognition: TBA.

About the Course

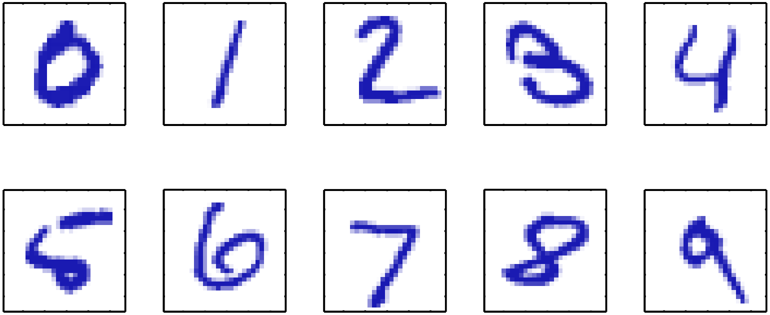
- Two lectures each week (Wednesday 13-15, Friday 9-11).
- Two practical assignments:
    - SPR: 4 computer lab sessions with R (weeks 46 and 48-50, Friday 11-13).
    - GPR.
- Grading:
    - Two assignments (25% each)
    - Written exam (50%)

What is statistical pattern recognition?

> *The field of pattern recognition/machine learning is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories.*

(Bishop, page 1)

Example: Handwritten Digit Recognition



28 $\times$ 28 pixel images

Machine Learning Approach

Use *training data*

$$\mathcal{D} = \{(\mathbf{x}_1, \mathbf{t}_1), \ldots, (\mathbf{x}_N, \mathbf{t}_N)\}$$

of N labeled examples, and fit a model to the training data.

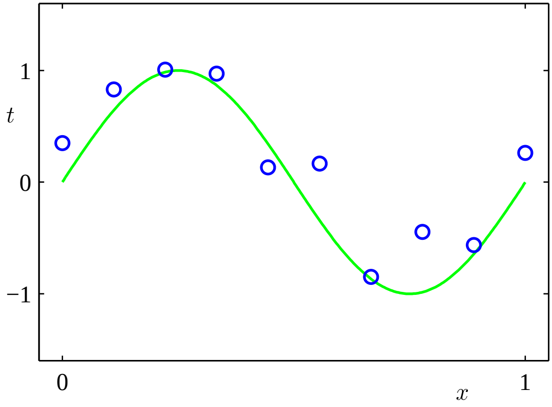
This model can subsequently be used to predict the class (digit) for new input vectors $\mathbf{x}$.

The ability to categorize correctly new examples is called *generalization*.

Types of Learning Problems

- Supervised Learning
  - Numeric target: regression.
  - Discrete unordered target: classification.
  - Discrete ordered target: ordinal classification/regression; ranking.
- Unsupervised Learning
  - Clustering.
  - Density estimation.

Example: Polynomial Curve Fitting



$t = \sin(2\pi x) + \varepsilon$, with $\varepsilon \sim \mathcal{N}(\mu = 0, \sigma = 0.3)$.

Polynomial Curve Fitting

Fit a model:

$$\begin{aligned} y(x, \mathbf{w}) &= w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M \\ &= \sum_{j=0}^{M} w_j x^j \end{aligned} \tag{1.1}$$

Linear function of the coefficients $\mathbf{w} = w_0, w_1, \ldots, w_M$.

The coefficients $\mathbf{w}$ are “learned” or “estimated” from the data.

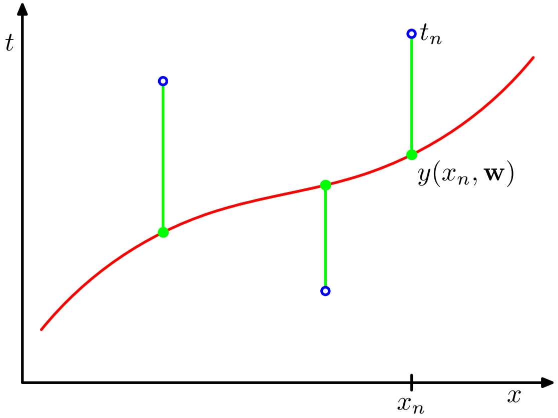
Error Function

We choose those values for **w** that minimize the sum of squared errors

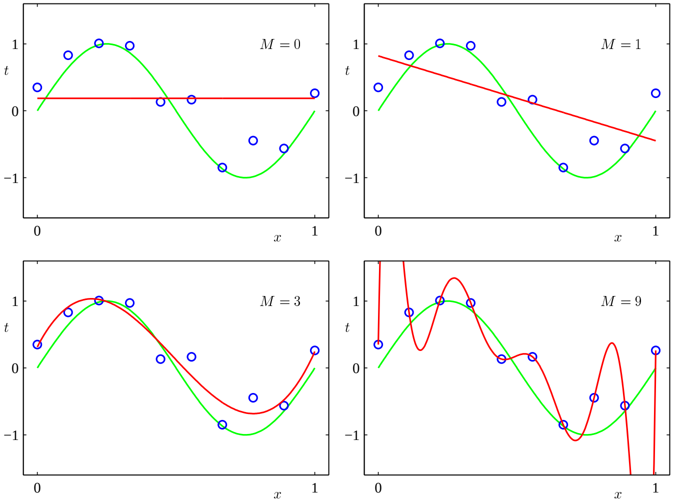
$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2 \tag{1.2}$$

Why *square* the difference between predicted and true value?

Error Function



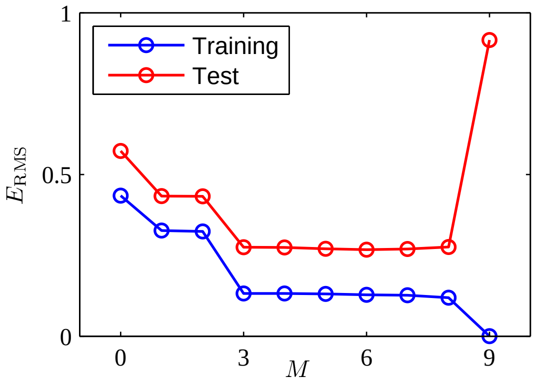
Curves Fitted with Least Squares (in red)



Magnitude of Coefficients

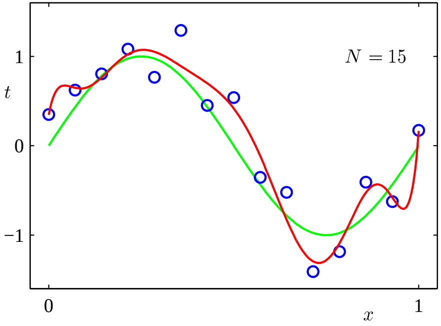
	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

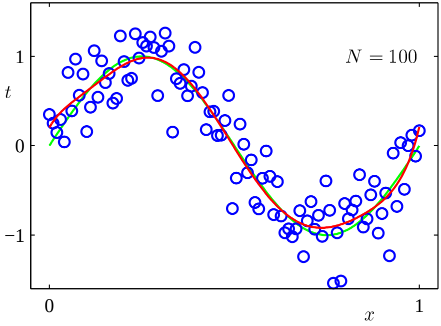
Training and Test Error



$$E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N} \tag{1.3}$$

Overfitting and Sample Size





Red curve ($M = 9$) is much more *smooth* for $N = 100$ than for $N = 15$. Also, it is closer to the true (green) curve.

Regularization

Adjusted error function

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2 \tag{1.4}$$

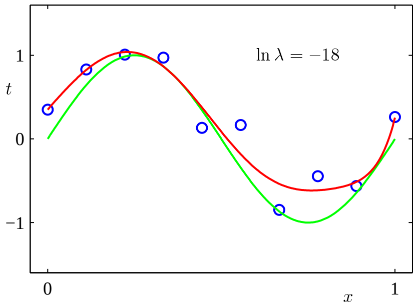
with $\|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w} = w_0^2 + w_1^2 + \ldots + w_M^2$.

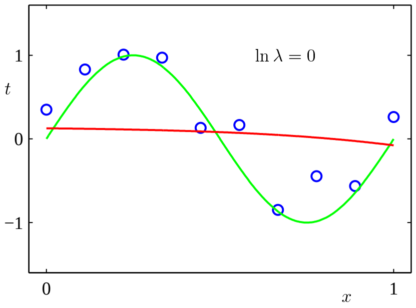
- Ridge regression
- Neural networks: weight decay

Magnitude of Coefficients ($M = 9$)

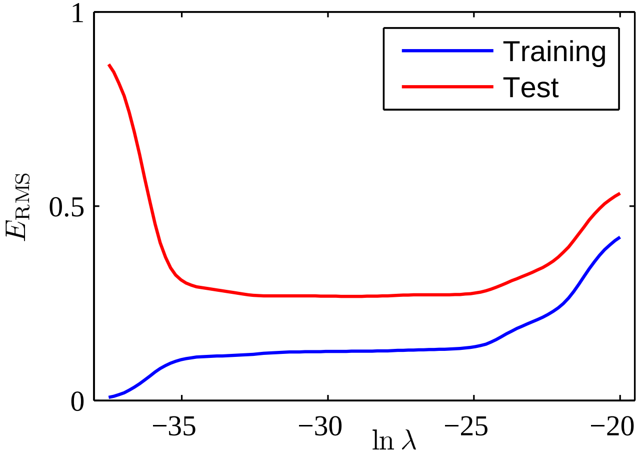
	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Fitted Curves for $M = 9$, $\lambda \approx 10^{-8}$, $\lambda = 1$.





RMSE versus $\ln \lambda$ for $M = 9$



Probability distribution and likelihood function

Binomial distribution with parameters N and π:

$$p(t) = \binom{N}{t} \pi^t (1-\pi)^{N-t}$$

Binomial distribution with $N = 10$ and $\pi = 0.7$:

$$p(t) = \binom{10}{t} 0.7^t \, 0.3^{10-t}$$

Probability of observing $t = 8$:

$$\binom{10}{8} 0.7^8 0.3^2 \approx 0.234$$

Likelihood function if we observe 7 heads in 10 trials:

$$L(\pi \mid t = 7) = \binom{10}{7} \pi^7 (1-\pi)^3$$

Probability and Likelihood

t	π				
	0.1	0.3	0.5	0.7	0.9
0	.349	.028	.001		
1	.387	.121	.01		
2	.194	.234	.044	.002	
3	.057	.267	.117	.009	
4	.011	.2	.205	.036	
5	.002	.103	.246	.103	.002
6		.036	.205	.2	.011
7		.009	.117	.267	.057
8		.002	.044	.234	.194
9			.01	.121	.387
10			.001	.028	.349
	1	1	1	1	1

Probability distribution for $\pi = 0.7$ and $N = 10$.

Probability and Likelihood

t	π				
	0.1	0.3	0.5	0.7	0.9
0	.349	.028	.001		
1	.387	.121	.01		
2	.194	.234	.044	.002	
3	.057	.267	.117	.009	
4	.011	.2	.205	.036	
5	.002	.103	.246	.103	.002
6		.036	.205	.2	.011
7		.009	.117	.267	.057
8		.002	.044	.234	.194
9			.01	.121	.387
10			.001	.028	.349
	1	1	1	1	1

Likelihood function for observing $t = 7$ in 10 trials.

Likelihood function

Let

$$\mathbf{t} = (t_1, \ldots, t_N)$$

be N independent observations, all from the same probability density

$$p(t \mid \theta),$$

where θ is the parameter vector of p (e.g. $\theta = (\mu, \sigma)$ for normal distribution), then

$$L(\theta \mid \mathbf{t}) \propto \prod_{n=1}^{N} p(t_n | \theta)$$

is the likelihood function for $\mathbf{t}$.

Maximum likelihood estimation:
Find that particular value θ_{ML} which maximizes L, i.e. that θ_{ML} such that the observed $\mathbf{t}$ are more likely to have come from $p(t \mid \theta_{\text{ML}})$ than from $p(t \mid \theta)$ for any other value of θ.

Maximum Likelihood Estimation

Take the derivatives of L with respect to the components of θ and equate them to zero (*normal equations*)

$$\frac{\partial L}{\partial \theta_j} = 0$$

Solve for the θ_j.
Maximizing the loglikelihood function is often easier

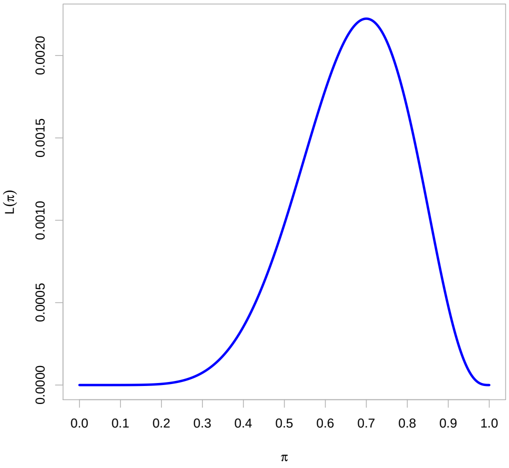
$$\begin{aligned}\mathcal{L}(\theta \,|\, \mathbf{t}) = \ln\{L(\theta \,|\, \mathbf{t})\} &= \ln\left\{\prod_{n=1}^{N} p(t_n \,|\, \theta)\right\} \\ &= \sum_{n=1}^{N} \ln p(t_n \,|\, \theta)\end{aligned}$$

since $\ln ab = \ln a + \ln b$.

This is allowed because the ln function is strictly increasing on $(0, \infty)$.

Likelihood function

Likelihood function for 7 heads out of 10:



Example: coin flipping

Random variable t with $t = 1$ if heads comes up, and $t = 0$ if tails comes up. Probability distribution for one coin flip

$$p(t) = \pi^t (1-\pi)^{1-t}$$

Sequence of N coin flips

$$p(\mathbf{t}) = p(t_1, t_2, \ldots, t_N) = \prod_{n=1}^{N} \pi^{t_n} (1-\pi)^{1-t_n}$$

which defines the likelihood when viewed as a function of π. The loglikelihood function consequently becomes

$$\mathcal{L}(\pi \mid \mathbf{t}) = \sum_{n=1}^{N} t_n \ln(\pi) + (1-t_n)\ln(1-\pi)$$

since $\ln a^b = b \ln a$.

Example: coin flipping (continued)

In a sequence of 10 coin flips with seven times heads coming up, we obtain

$$\mathcal{L}(\pi) = \ln(\pi^7(1-\pi)^3) = 7\ln\pi + 3\ln(1-\pi)$$

To determine the maximum we take the derivative with respect to π, equate to zero, and solve for π:

$$\frac{d\mathcal{L}}{d\pi} = \frac{7}{\pi} - \frac{3}{1-\pi} = 0$$

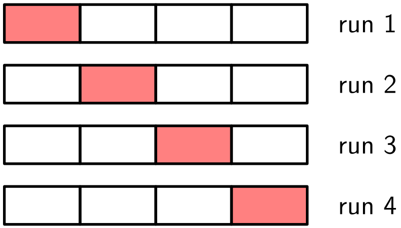
which yields maximum likelihood estimate $\pi_{\mathrm{ML}} = 0.7$.

Note:

$$\frac{d\ln x}{dx} = \frac{1}{x}$$

Model Selection

- Cross-Validation

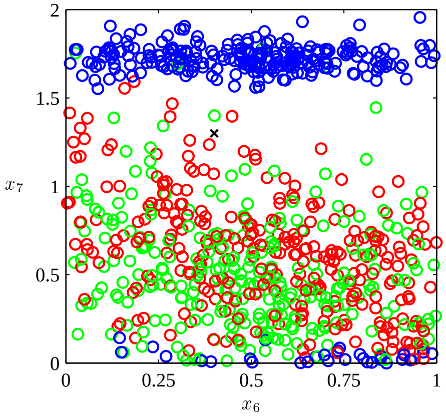


- Score = Quality of Fit − Complexity Penalty

  For example: $\text{AIC} = \ln p(\mathcal{D}|\mathbf{w}_{\text{ML}}) - M$

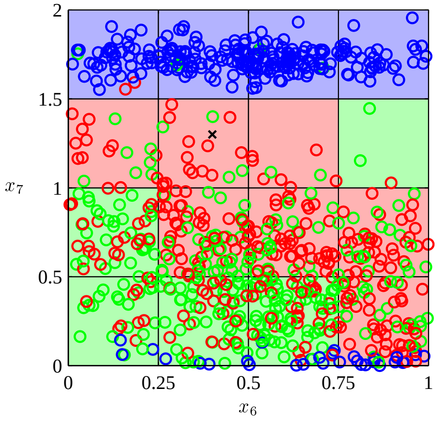
where $\ln p(\mathcal{D}|\mathbf{w}_{\text{ML}})$ is the maximized loglikelihood and M is the number of parameters of the fitted model.

The Curse of Dimensionality



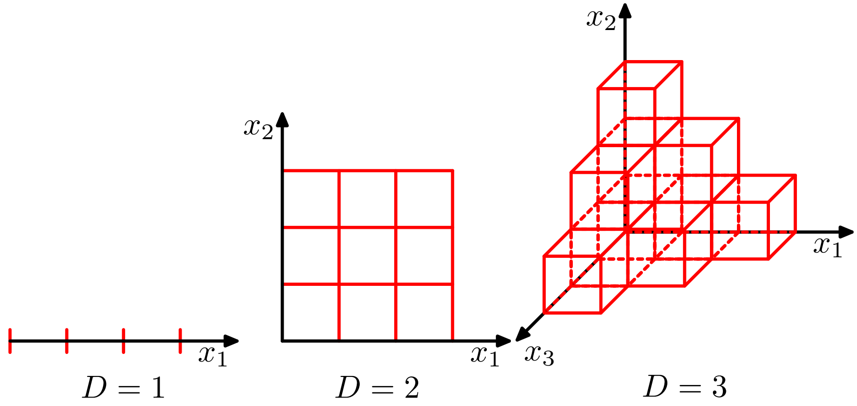
Predict class of ×.

The Curse of Dimensionality



Predict class of ×.

The Curse of Dimensionality



Decision Theory

Suppose we have to make a decision in a situation involving uncertainty. Two steps

1. *Inference*: Learn $p(\mathbf{x}, t)$ from data. This problem is the main subject of this course.
2. *Decision*: Given this estimate of $p(\mathbf{x}, t)$, determine the optimal decision. Relatively straightforward.

Decision Theory: Example

Predict whether patient has cancer from X-ray image.

Let $t = 1$ denote that cancer is present.
Then knowledge of

$$p(t = 1|\mathbf{x}) = \frac{p(\mathbf{x}|t = 1)p(t = 1)}{p(\mathbf{x})}$$

would allow us to make optimal predictions of t from $\mathbf{x}$ (given an appropriate loss function).

Loss Functions for Classification

Suppose we know $p(\mathbf{x}, t)$.
Task: given a value for $\mathbf{x}$, predict the class label t.
$y(\mathbf{x})$: predicted value for t at $\mathbf{x}$.
$L(t, y(\mathbf{x}))$: loss of predicting the value $y(\mathbf{x})$ when the true value is t.

We want to choose $y(\mathbf{x})$ so as to minimize the expected loss

$$\begin{aligned}\mathbb{E}[L] &= \int_{\mathbf{x}}\int_t L(t, y(\mathbf{x}))p(\mathbf{x}, t)\mathrm{d}\mathbf{x}\mathrm{d}t \\ &= \int_{\mathbf{x}}\int_t L(t, y(\mathbf{x}))p(t|\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x}\mathrm{d}t\end{aligned}$$

Hence we have

$$\mathbb{E}[L] = \int_{\mathbf{x}} \left(\sum_{k=1}^{K} L(\mathcal{C}_k, y(\mathbf{x}))p(\mathcal{C}_k|\mathbf{x})\right) p(\mathbf{x})\mathrm{d}\mathbf{x}$$

since t is a categorical variable with possible values $\mathcal{C}_1, \ldots, \mathcal{C}_K$.

Loss Functions for Classification

It suffices to minimize

$$\sum_{k=1}^{K} L(\mathcal{C}_k, y(\mathbf{x}))p(\mathcal{C}_k|\mathbf{x})$$

for each point $\mathbf{x}$. Thus the decision rule that minimizes the expected loss is the one that assigns each point $\mathbf{x}$ to the class j for which the quantity

$$\sum_{k=1}^{K} L_{kj}p(\mathcal{C}_k|\mathbf{x}) \tag{1.81}$$

is a minimum, where L_{kj} is the loss incurred when we predict class $\mathcal{C}_j$ when the true class is $\mathcal{C}_k$.

Minimizing the Misclassification Rate

To minimize the misclassification rate, we take

$$L_{kj} = \begin{cases} 0 & \text{if } j = k \\ 1 & \text{otherwise} \end{cases}$$

The minimum of

$$\sum_{k=1}^{K} L_{kj} p(\mathcal{C}_k|\mathbf{x}) = \sum_{k \neq j} p(\mathcal{C}_k|\mathbf{x}) = 1 - p(\mathcal{C}_j|\mathbf{x})$$

is now achieved if we assign to the class j for which $p(\mathcal{C}_j|\mathbf{x})$ is maximum.

Minimizing Expected Loss

Example loss matrix for prediction of cancer:

k/j	0	1
0	0	1
1	10	0

Suppose $p(t=0) = 0.8$ and $p(t=1) = 0.2$.

The expected loss of predicting "no cancer present" is:

$$L_{00} \times p(t=0) + L_{10} \times p(t=1) = 0 \times 0.8 + 10 \times 0.2 = 2$$

The expected loss of predicting "cancer present" is:

$$L_{01} \times p(t=0) + L_{11} \times p(t=1) = 1 \times 0.8 + 10 \times 0 = 0.8$$

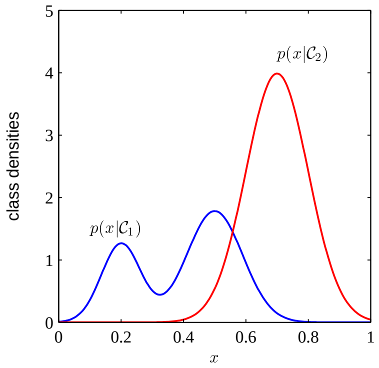
Even though the probability of cancer is "only" 0.2, loss is minimized if we predict (act as if) cancer is present.

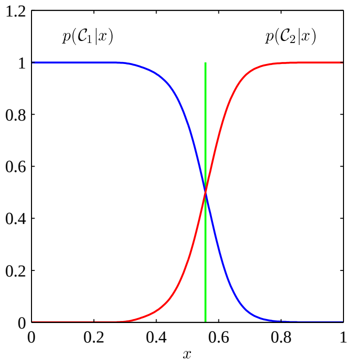
Inference and Decision

Approaches to solving decision problems

1. Generative models: learn $p(\mathbf{x}|\mathcal{C}_k)$ and $p(\mathcal{C}_k)$ from data and use Bayes' rule
$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})} \tag{1.82}$$
to find the posterior class probabilities $p(\mathcal{C}_k|\mathbf{x})$.
2. Discriminative models: model $p(\mathcal{C}_k|\mathbf{x})$ directly.
3. Discriminant functions: map $\mathbf{x}$ directly onto a class label without making probability estimates.

Inference and Decision





Loss Functions for Regression

Expected loss

$$\begin{aligned}\mathbb{E}[L] &= \int_x \int_t L(t, y(x))p(x,t)\mathrm{d}x\mathrm{d}t \\ &= \int_x \left(\int_t L(t, y(x))p(t|x)\mathrm{d}t \right) p(x)\mathrm{d}x \qquad (1.86)\end{aligned}$$

For squared error loss

$$\mathbb{E}[L] = \int_x \underbrace{\left(\int_t \{y(x) - t\}^2 p(t|x)\mathrm{d}t \right)}_{\mathbb{E}[\{y(x)-t\}^2]} p(x)\mathrm{d}x \qquad (1.87)$$

Properties of Expectation and Variance

Some useful properties:

1. $\mathbb{E}[c] = c$ for constant c.
2. $\mathbb{E}[cx] = c\mathbb{E}[x]$.
3. $\mathbb{E}[x \pm y] = \mathbb{E}[x] \pm \mathbb{E}[y]$.
4. $\text{var}[c] = 0$ for constant c.
5. $\text{var}[cx] = c^2\text{var}[x]$.
6. $\text{var}[x \pm y] = \text{var}[x] + \text{var}[y]$ if x and y independent.

Minimizing expected squared prediction error

Given a random draw from $p(t|x_0)$ predict the value of t by some number $y(x_0)$. We have

$$\begin{aligned}\mathbb{E}[\{y(x_0)-t\}^2] &= \mathbb{E}[y(x_0)^2 - 2y(x_0)t + t^2] \\ &= y(x_0)^2 - 2y(x_0)\mathbb{E}[t|x_0] + \mathbb{E}[t^2|x_0]\end{aligned}$$

To minimize this expression we solve

$$\frac{d}{dy(x_0)} = 2y(x_0) - 2\mathbb{E}[t|x_0] = 0$$

which gives $y(x_0) = \mathbb{E}[t|x_0]$.

Minimizing expected squared prediction error

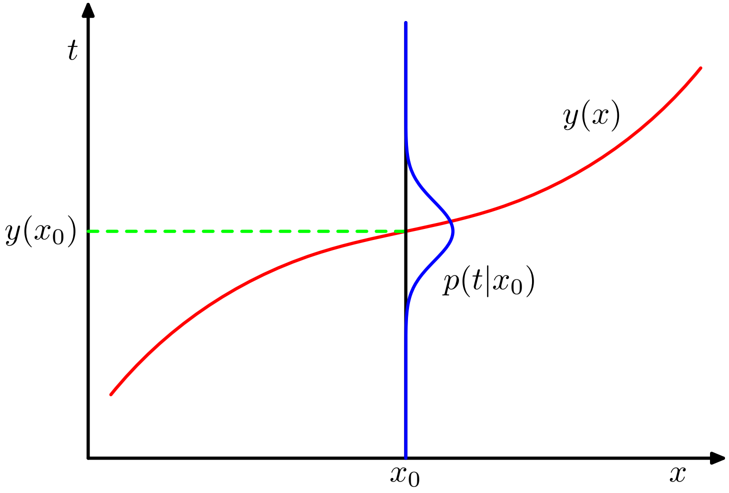
Since this reasoning applies to any value of x we might pick, we have that

$$y(x) = \mathbb{E}_t[t|x] \tag{1.89}$$

minimizes the expected squared prediction error.

The function $\mathbb{E}_t[t|x]$ is called the (population) regression function.

Population Regression Function



Question

We have derived the result that

$$y(x) = \mathbb{E}_t[t|x] \tag{1.89}$$

minimizes the expected squared prediction error.

How could we use this result to construct a prediction rule $y(x)$ from a finite data sample

$$\mathcal{D} = \{(x_1, t_1), \ldots, (x_N, t_N)\}?$$

Nearest-neighbor functions

Consider a regression problem with input variable x and the output variable t:

- for each input value x, we define a *neighborhood* $N_k(x)$ containing the indices n of the k points (x_n, t_n) from the training data that are the *closest* to x;
- from the neighborhood function $N_k(x)$, we construct the function

$$y_k(x) = \frac{1}{k} \sum_{n \in N_k(x)} t_n$$

The function $y_k(x)$ is called the *k-nearest neighbor function*.

An example learning problem

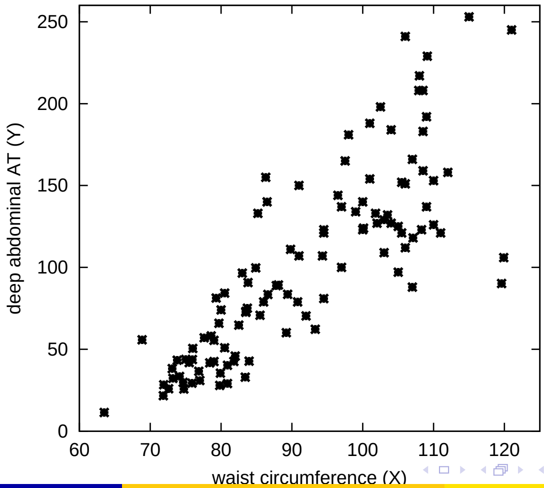
In a clinical study of risk factors for cardiovascular disease,

- the *independent variable* x is a patient's waist circumference;
- the *dependent variable* t is a patient's deep abdominal adipose tissue.

The researchers want to *predict* the amount of deep abdominal adipose tissue from a simple measurement of waist circumference.

Scatterplot of the data

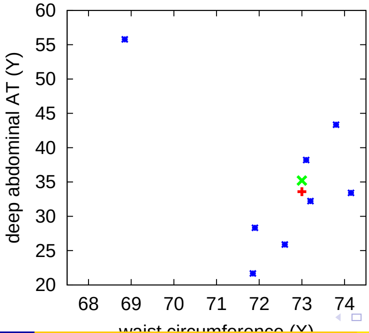
For *learning* the relationship between x and t, measurements (x_n, t_n) on 109 men between 18 and 42 years of age, are available:



An example

We consider *eight* (consecutive) points (x_n, t_n) from the clinical study of risk factors for cardiovascular disease:

1.(68.85, 55.78)	5.(73.10, 38.21)
2.(71.85, 21.68)	6.(73.20, 32.22)
3.(71.90, 28.32)	7.(73.80, 43.35)
4.(72.60, 25.89)	8.(74.15, 33.41)



The example (continued)

With $k = 2$, the neighborhood of $x = 73.00$ equals

$$N_2(x = 73.00) = \{5, 6\}$$

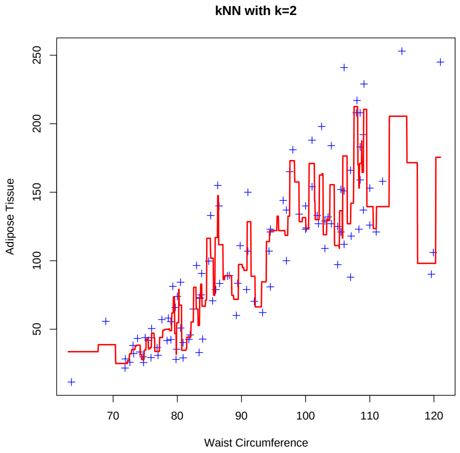
and we find

$$y_2(x = 73.00) = \frac{38.21 + 32.22}{2} = 35.215$$

With $k = 5$, we find $y_5(x = 73.00) = 33.598$.

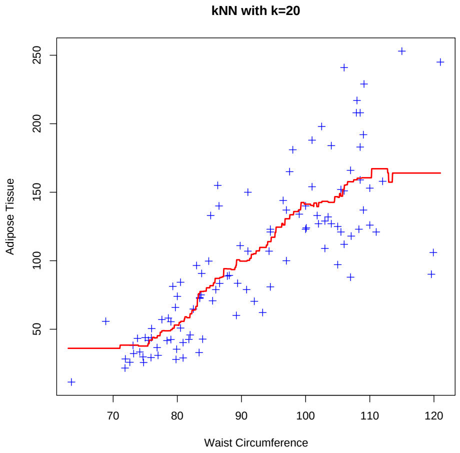
The example continued

With $k = 2$ and *Euclidean distance*, the following k-nearest neighbor function is constructed from the training data:

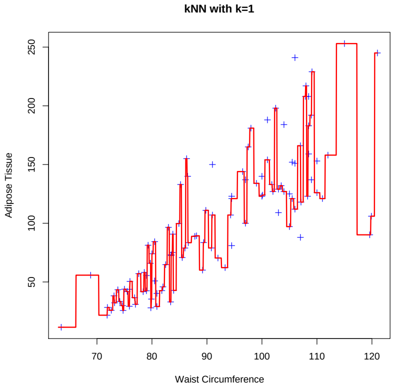


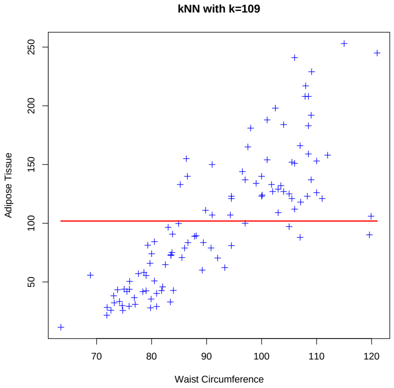
The example continued

With $k = 20$ and *Euclidean distance*, the following k-nearest neighbor function is constructed from the training data:



kNN: going to the extremes





The idea of k-nearest neighbor

We recall that, for a regression problem, the *best prediction* for the output variable t at the input value x is the *mean* $\mathbb{E}[t|x]$:

- the nearest-neighbor function *approximates* the mean by *averaging* over the training data;
- the nearest-neighbor function *relaxes* conditioning at a *specific input value* to the *neighborhood* of that value.

The nearest-neighbor function thus *implements* the idea of selecting the means for prediction *directly*.