Lecture 5: K Nearest Neighbour

Andreas Wichert
Department of Computer Science and Engineering
Técnico Lisboa

1

Parametric Models

- We select a hypothesis space and adjust a fixed set of parameters with the training data, $y = y(\boldsymbol{x}, \boldsymbol{w})$
- We assume that the parameters $\boldsymbol{w}$ sumarise the training (compression)
- This methods are called parametric models
- Example:
  - Decision Trees
  - Bayesian Learning
  - DL
- When we have a small amount of data it makes sense to have a small set of parameters (avoiding overfitting)
- When we have a large quantity of data, overfitting is less an issue

2

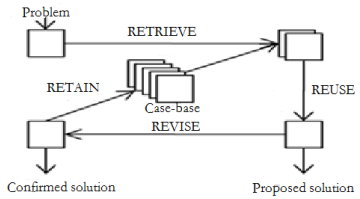
Non-Parametric Learning

- A non-parametric model is one that can not be characterized by a fixed set of parameters
- A family of non-parametric models is Instance Based Learning
- Instance Based Learning is based on the memorization of the database
- There is **no model** associated to the learned concepts
- The classification is obtained by looking into the memorized examples
- When a new query instance is encountered, a set of similar related instances is retrieved from memory and used to classify the new query instance

3

Case-based reasoning

- Instance-based methods can also use more complex, symbolic representations
- In case-based learning, instances are represented in this fashion and the process for identifying neighboring instances is elaborated accordingly



4

- The cost of the learning process is *0*, all the cost is in the computation of the prediction
- This kind learning is also known as *lazy learning*

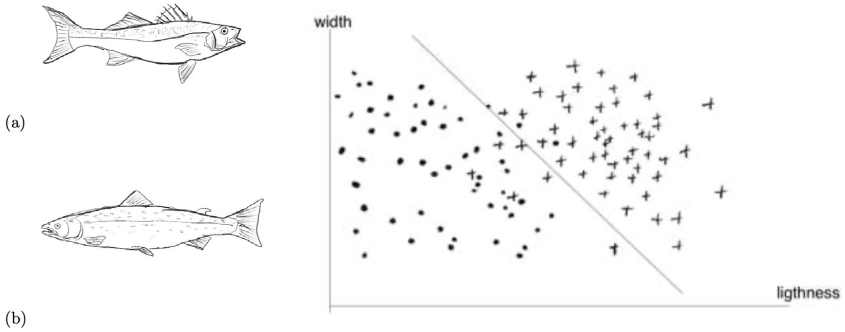
- One disadvantage of instance-based approaches is that the cost of classifying new instances can be high
  - Nearly all computation takes place at classification time rather than learning time

5

- When a new query instance is encountered, a set of similar related instances is retrieved from memory and used to classify the new query instance

- What does it mean “similar related instances”?
- How to measure if something is similar?
- If we change the distance (similarity) function, we change how examples are classified…

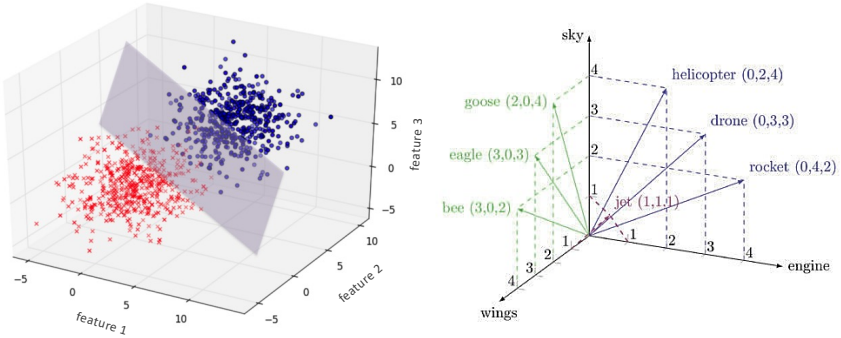
6



- In a sample, each salmon (b) is represented by a **dot** and each sea bass (a) by a **cross**

7

Multivariate feature space



8

8

Vector

- A vector **x** of dimension D is represented as

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix} = (x_1, x_2, \cdots, x_D)^T$$

- One should be careful since **x** and the transpose $\mathbf{x}^T$ is not the same

9

Norm

- A norm describes the length of a vector, it is a function given a *vector space* V that maps a vector into a real number

$$\|\mathbf{x}\| \geq 0, \quad and \quad \|\mathbf{x}\| = 0 \quad for \quad \mathbf{x} = 0 \quad only,$$

with α scalar

$$\|\alpha \cdot \mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|,$$

$$\|\mathbf{x} + \mathbf{y}\| \leq |\mathbf{x}\| + |\mathbf{y}\|.$$

10

L_p norm

- The l_p norm is defined as the following (for **p = 2** it is the Euclidean norm):

$$\|\mathbf{x}\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_D|^p)^{\frac{1}{p}}$$

- The maximum l_∞ norm is defined as

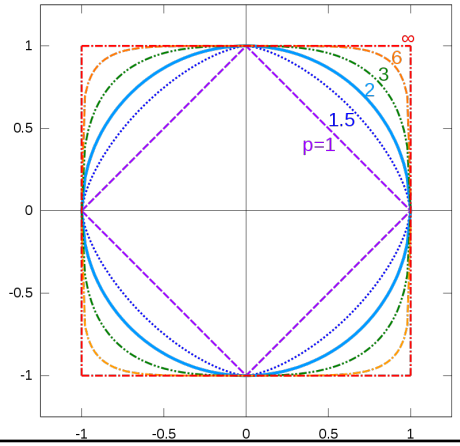
$$\|\mathbf{x}\|_\infty = max\,(|x_1|, |x_2|, \cdots, |x_D|)\,.$$

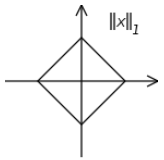
- $0 < q < p$

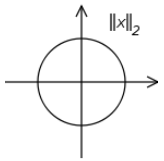
$$\|\mathbf{x}\|_p \leq \|\mathbf{x}\|_q \leq m^{\frac{1}{q}-\frac{1}{p}} \cdot \|\mathbf{x}\|_p$$

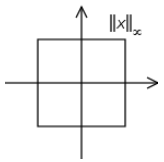
11

Unit circles









12

Metric

- A given metric *d* defines a distance between two vectors it is always positive or zero

$$d(\mathbf{x}, \mathbf{y}) \geq 0,$$

if it is symmetric

$$d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$$

and if the triangle inequality holds

$$d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y}).$$

13

- The l_p norms induce metrics that are distances between two points

$$d_p(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p = (|x_1 - y_1|^p + |x_2 - y_2|^p + \cdots + |x_D - y_D^p)^{\frac{1}{p}}$$

- The most popular metrics are the Taxicab or Manhattan metric d_1 with

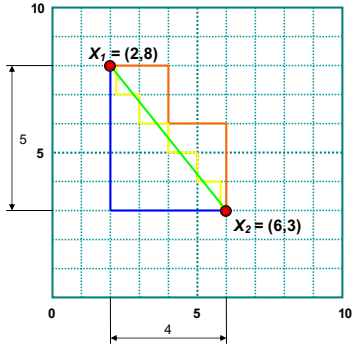
$$d_1(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_1 = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_m - y_m|$$

- and the Euclidean metric

$$d_2(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2 = \sqrt{|x_1 - y_1|^2 + |x_2 - y_2|^2 + \cdots + |x_m - y_m|^2}.$$

14

Common distance metrics
(numeric data)



2D example

x_1 = (2,8)

x_2 = (6,3)

Euclidean distance

$$d_2(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{|2-6|^2 + |8-3|^2} = \sqrt{41}$$

Manhattan distance

$$d_1(\mathbf{x}_1, \mathbf{x}_2) = |2-6| + |8-3| = 9$$

15

Hamming distance
(binary and categorical data)

- Simple: number of different features between two feature vectors

- Distance of (1011101) and (1001001) is 2
- Distance (2143896) and (2233796)
- Distance between (toned) and (roses) is 3

16

- The Euclidean norm is induced by the inner product (scalar product)

$$\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}|\mathbf{x}\rangle}$$

- and defines a **Hilbert space**, which extends the two or three dimensional Euclidean space to spaces with any finite or infinite number of dimensions

- A scalar product exists in l_2 but not in l_1 space. A normed vector space (does not need to have a scalar product) is called a **Banach space**.
  - Without a scalar product there is no orthogonality

17

David Hilbert, (1862 - 1943) one of the most famous German mathematicians, attended a banquet in 1934, and he was seated next to the new minister of education, Bernhard Rust. Rust asked, "How is mathematics in Gottingen now that it has been freed of the Jewish influence?" Hilbert replied, "Mathematics in Gottingen? There is really none any more."

Stefan Banach (1892 -1945) a Polish mathematician who is generally considered one of the world's most important and influential 20th-century mathematicians.

18

Cosine Similarity

- By normalizing the vector to the length one (unit vectors) the Euclidean distance function is constrained to the unit sphere and corresponds to the cosine of the angle ω between the vectors

$$\cos\omega = \frac{\langle \mathbf{x}|\mathbf{y}\rangle}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}$$

- with a similarity function

$$-1 \leq sim(\mathbf{x}, \mathbf{y}) = \cos\omega \leq 1$$

19

Cosine Similarity vs. Euclidean Distance

- By normalizing the vector to length one the Euclidean distance function is constrained to a unit radius ball

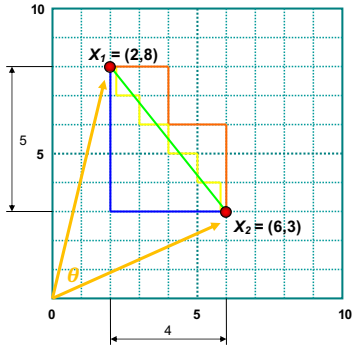
$$0 \leq d\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}, \frac{\mathbf{y}}{\|\mathbf{y}\|}\right) = \left\|\frac{\mathbf{x}}{\|\mathbf{x}\|} - \frac{\mathbf{y}}{\|\mathbf{y}\|}\right\| \leq \sqrt{2}.$$

$$\cos\omega = \frac{\langle \mathbf{x}|\mathbf{y}\rangle}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}$$

$$-1 \leq sim(\mathbf{x}, \mathbf{y}) = \cos\omega \leq 1$$

20

Cosine similarity



Euclidean distance

$$d_2(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{|2-6|^2 + |8-3|^2} = \sqrt{41}$$

Manhattan distance

$$d_1(\mathbf{x}_1, \mathbf{x}_2) = |2-6| + |8-3| = 9$$

Cosine similarity (*not a distance function*)

$$\cos(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{\|\mathbf{x}_1\| \|\mathbf{x}_2\|} = \frac{2\times6+8\times3}{\sqrt{2^2+8^2}\sqrt{6^2+3^2}} = 0.65$$

21

- If we do not normalize the vectors, then we get the simple scalar product also called the dot product

$$\langle \mathbf{x} | \mathbf{w} \rangle = \cos\omega \cdot \|\mathbf{x}\| \cdot \|\mathbf{w}\|,$$

$$net := \langle \mathbf{x} | \mathbf{w} \rangle = \sum_{i=1}^{D} w_j \cdot x_j,$$

- The scalar product is commutative

$$\langle \mathbf{x} | \mathbf{w} \rangle = \mathbf{x}^T \cdot \mathbf{w} = \mathbf{w}^T \cdot \mathbf{x} = \langle \mathbf{w} | \mathbf{x} \rangle.$$

22

- Matrix multiplication between vectors is not commutative

$$\mathbf{x}^T \cdot \mathbf{w} = \langle \mathbf{x} | \mathbf{w} \rangle = \begin{pmatrix} w_0 & w_1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = w_0 \cdot x_0 + w_1 \cdot x_1$$

is very different from

$$\mathbf{x} \cdot \mathbf{w}^T = |\mathbf{w}\rangle\langle\mathbf{x}| = \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} \cdot \begin{pmatrix} x_0 & x_1 \end{pmatrix} = \begin{pmatrix} w_0 \cdot x_0 & w_0 \cdot x_1 \\ w_1 \cdot x_0 & w_1 \cdot x_1 \end{pmatrix}.$$

23

K-Nearest Neighbor

- In nearest-neighbor learning the target function may be either discrete-valued or real valued
- Learning a discrete valued function

- $f : \Re^d \rightarrow V$, *V* is the finite set *{v1,.....,vn}*

- For discrete-valued, the *k*-NN returns the most common value among the k training examples nearest to *xq*.
- $Data = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \cdots, , (\mathbf{x}_N, t_N)\}$

$$f(\mathbf{x}_\eta) = t_\eta = v_\eta$$

24

K-Nearest Neighbor

$$Data = \{(\mathbf{x}_1, f(\mathbf{x}_1)), (\mathbf{x}_2, f(\mathbf{x}_1)), \cdots, , f(\mathbf{x}_N, (\mathbf{x}_N))\}$$

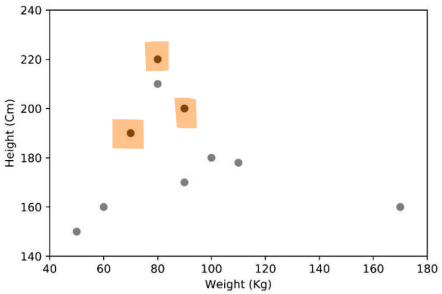
- Training algorithm
  - For each training example ***(x,f(x))*** add the example to the list
- Classification algorithm
  - Given a query instance x_q to be classified
    - Let $x_1,..,x_k$ k instances which are nearest to x_q

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\arg\max} \sum_{i=1}^{k} \delta(v, f(x_i))$$

    - Where $\delta(a,b)=1$ if $a=b$, else $\delta(a,b)=0$ (Kronecker function)

25

- X1: "Weight (Kg)"
- X2: "Height (Cm)"
- Class: "NBA"
  - (National Basketball Association) Player"

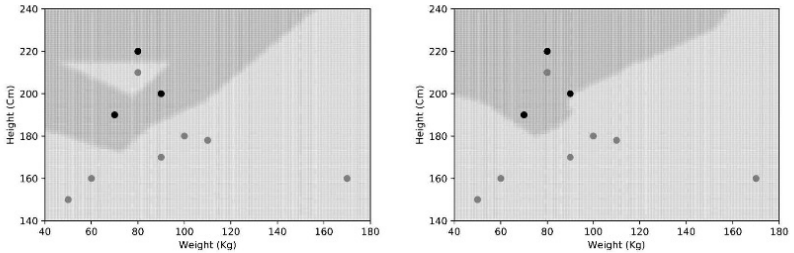


	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\mathbf{x}^{(3)}$	$\mathbf{x}^{(4)}$	$\mathbf{x}^{(5)}$	$\mathbf{x}^{(6)}$	$\mathbf{x}^{(7)}$	$\mathbf{x}^{(8)}$	$\mathbf{x}^{(9)}$	$\mathbf{x}^{(10)}$
X_1	170	80	90	60	50	70	90	100	110	80
X_2	160	220	200	160	150	190	170	180	178	210
$Class$	0	1	1	0	0	1	0	0	0	0

26

Diffferent Values of k for kNN

(a) (b)



(a) k=1 (b) k=2 *(for l_2)*

27

Which Class is $\mathbf{x} = \begin{bmatrix} 100 & 210 \end{bmatrix}^T$ in l_2 metric?

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\mathbf{x}^{(3)}$	$\mathbf{x}^{(4)}$	$\mathbf{x}^{(5)}$	$\mathbf{x}^{(6)}$	$\mathbf{x}^{(7)}$	$\mathbf{x}^{(8)}$	$\mathbf{x}^{(9)}$	$\mathbf{x}^{(10)}$
X_1	170	80	90	60	50	70	90	100	110	80
X_2	160	220	200	160	150	190	170	180	178	210
$Class$	0	1	1	0	0	1	0	0	0	0
$\left\|\mathbf{x} - \mathbf{x}^{(i)}\right\|_2$	86.0	22.4	14.1	64.0	78.1	36.1	41.2	30.0	33.5	20.0

- k=1; is $\mathbf{x}^{(3)}$: **is Class 1 (NBA)**
- k=3; is is $\mathbf{x}^{(3)}$: is Class 1, $\mathbf{x}^{(10)}$: is Class 0 , $\mathbf{x}^{(2)}$: is Class 1. **Majority of two examples have Class 1 -> Class 1**
- k=5; is is $\mathbf{x}^{(3)}$: is Class 1, $\mathbf{x}^{(10)}$: is Class 0 , $\mathbf{x}^{(2)}$: is Class 1, is is $\mathbf{x}^{(8)}$: is Class 0, $\mathbf{x}^{(9)}$: is Class 0.
  - **Majority of three examples have Class 0 -> Class 0**

28

Which Class is in $\mathbf{x} = \begin{bmatrix} 100 & 210 \end{bmatrix}^T$ l_1 metric?

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\mathbf{x}^{(3)}$	$\mathbf{x}^{(4)}$	$\mathbf{x}^{(5)}$	$\mathbf{x}^{(6)}$	$\mathbf{x}^{(7)}$	$\mathbf{x}^{(8)}$	$\mathbf{x}^{(9)}$	$\mathbf{x}^{(10)}$
X_1	170	80	90	60	50	70	90	100	110	80
X_2	160	220	200	160	150	190	170	180	178	210
$Class$	0	1	1	0	0	1	0	0	0	0
$\left\|\mathbf{x} - \mathbf{x}^{(i)}\right\|_1$	120	30	20	90	110	50	50	30	42	20

- k=1; is $\mathbf{x}^{(3)}$ $\mathbf{x}^{(10)}$: **Undecided, random choice**
- k=3 **Undecided, random choice**
  - k=3; is is $\mathbf{x}^{(3)}$: is Class 1, $\mathbf{x}^{(10)}$: is Class 0 , $\mathbf{x}^{(2)}$: is Class 1. **Majority of two examples have Class 1 -> Class 1**
  - k=3; is is $\mathbf{x}^{(3)}$: is Class 1, $\mathbf{x}^{(10)}$: is Class 0 , $\mathbf{x}^{(8)}$: is Class 0. **Majority of two examples have Class 1 -> Class 0**
- k=5; is is $\mathbf{x}^{(3)}$: is Class 1, $\mathbf{x}^{(10)}$: is Class 0 , $\mathbf{x}^{(2)}$: is Class 1, $\mathbf{x}^{(8)}$: is Class 0, $\mathbf{x}^{(9)}$: is Class 0.
  - **Majority of three examples have Class 1 -> Class 0**

29

How to determine *k*?

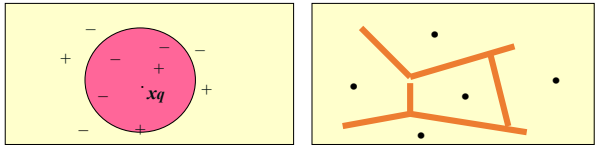
- Determined experimentally
  - use a test set to validate the error rate of the classifier
  - start with k=1 and assess the error rate
  - repeat with k=k+2
  - choose the value of k for which the error rate is minimum
- Note: *k* should be odd number to avoid ties
- More to come on
  - simple error rate = number of observations incorrectly classified. Better alternatives?
  - training *versus* testing error rates?

30

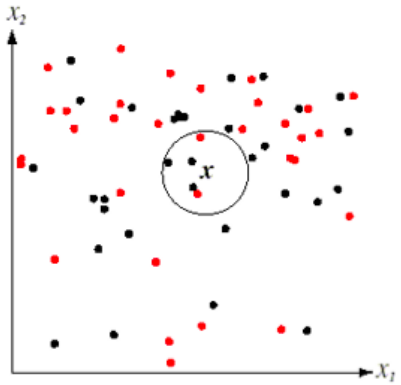
30

Definition of Voronoi diagram

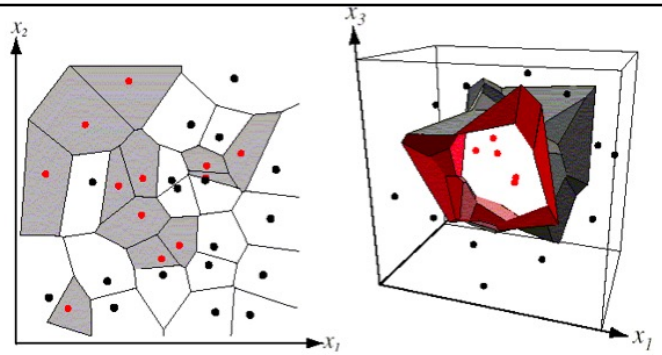
- The decision surface induced by 1-NN for a typical set of training examples.



31



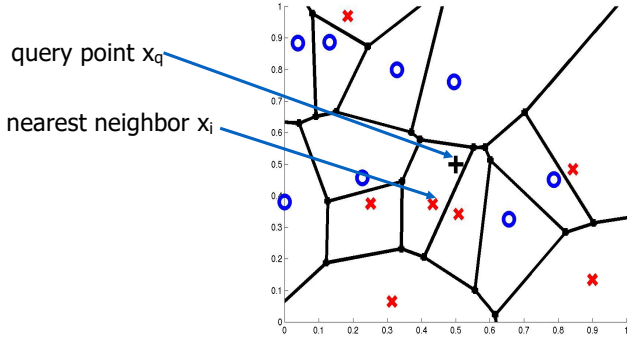
32



- kNN rule leeds to partition of the space into cells (Vornoi cells) enclosing the training points labelled as belonging to the same class
- The decision boundary in a Vornoi tessellation of the feature space resembles the surface of a crystall

33

1-Nearest Neighbor



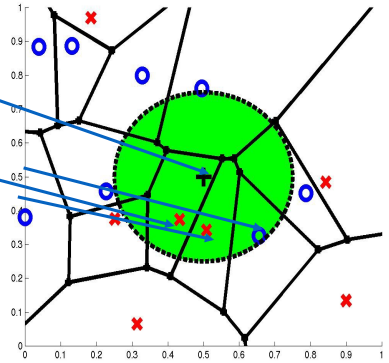
34

3-Nearest Neighbors

query point x_q

3 nearest neighbors

2x,1o



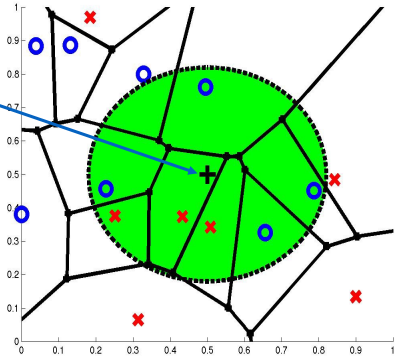
35

7-Nearest Neighbors

query point x_q

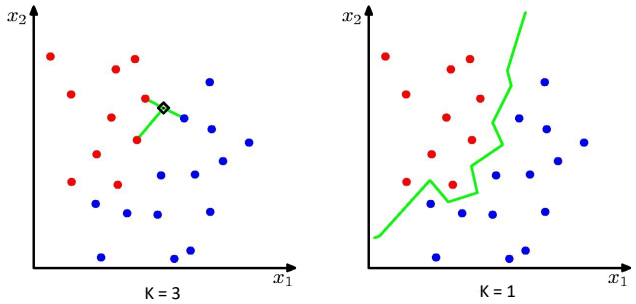
7 nearest neighbors

3x,4o



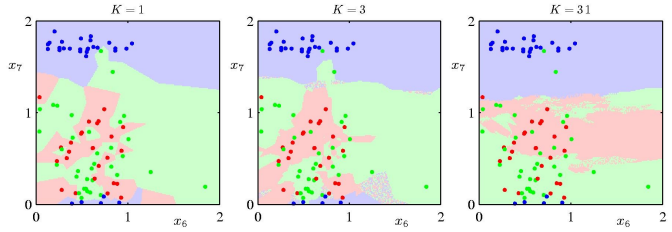
36

K-Nearest-Neighbours for Classification



37

K-Nearest-Neighbours for Classification



- K acts as a smother

38

Distance Weighted

- Refinement to kNN is to weight the contribution of each *k* neighbor according to the distance to the query point x_q
  - Greater weight to closer neighbors
  - For discrete target functions
    - With Euclidean distance *d(x,y)* we do not need to compute the square root, $d(x,y)^2$

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\arg\max} \sum_{i=1}^{k} w_i \delta(v, f(x_i))$$

$$w_i = \begin{cases} \dfrac{1}{d(x_q, x_i)^2} & if \quad x_q \neq x_i \\ 1 & else \end{cases}$$

39

Curse of Dimensionality

- Imagine instances described by **20 features** (attributes) **but only 3 are relevant** to target function
- **Curse of dimensionality**: nearest neighbor is easily misled when instance space is **high-dimensional**
- Dominated by large number of irrelevant features

Possible solutions

- Stretch j-th axis by weight z_j, where $z_1,\ldots,z_n$ chosen to minimize prediction error (weight different features differently)
- Use cross-validation to automatically choose weights $z_1,\ldots,z_n$
- Note setting z_j to zero eliminates this dimension altogether (feature subset selection)
- PCA (later)

40

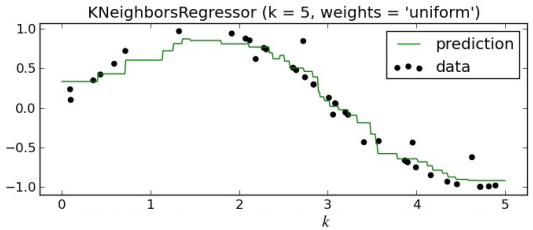
Disatvantages

- One disadvantage of instance-based approaches is that the cost of classifying new instances can be high
- How can we reduce the classification costs (time)?

- Therefore, techniques for **efficiently indexing** training examples are a significant practical issue in reducing the computation required at query time (High Dimensional Indexing, tree indexing will not work!)
- Compression, reduce the number of representatives (LVQ)
  - Two Examples…

41

Continuous-valued target functions

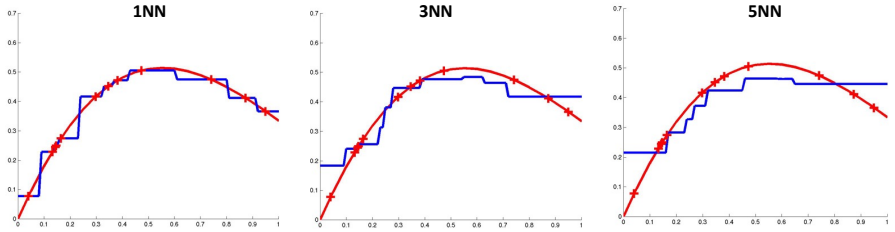


$$f : \Re^d \rightarrow \Re \qquad \hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k}$$

- kNN approximating continuous-valued target functions
  - Calculate the mean value of the *k* nearest training examples rather than calculate their most common value

42

*k*NN regressor



- in **red**: samples of a well-defined signal
- in **blue**: kNN regression function

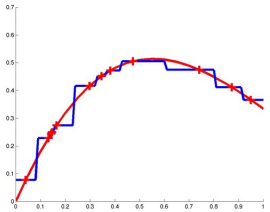
43

Distance Weighted

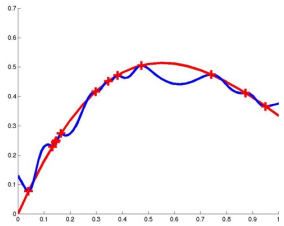
- For real valued functions
  - with Euclidean distance *d(x,y)* we do not need to compute the square root, *d(x,y)²*

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i}$$

$$w_i = \begin{cases} \dfrac{1}{d(x_q, x_i)^2} & if \quad x_q \neq x_i \\ 1 & else \end{cases}$$



1-NN mean value



1-NN distance weighted

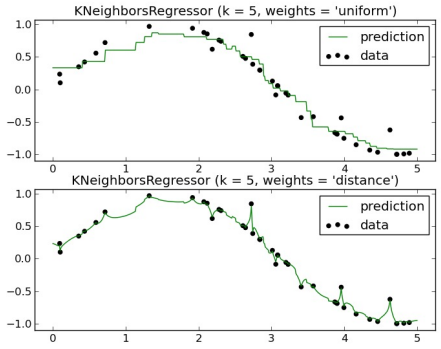
44

Distance weighted kNN

- In **regression** (numeric targets):

$$f(\mathbf{x}_{new}) \leftarrow \frac{\sum_{i=1}^{k} w_i f(\mathbf{x}_i)}{\sum_{i=1}^{k} w_i}$$

$$w_i(\mathbf{x}_{new}) = \begin{cases} \dfrac{1}{d(\mathbf{x}_{new}, \mathbf{x}_i)} & \text{if } \mathbf{x}_{new} \neq \mathbf{x}_i \\ 1 & \text{else} \end{cases}$$

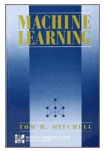


45

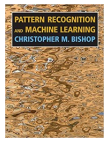
- Can fit low dimensional, very complex, functions very accurately
- Training, adding new data, is almost free
- Doesn't forget old training data
- Lazy: wait for query before generalizing
- Lazy learner can create local approximations

46

Literature



- Tom M. Mitchell, Machine Learning, McGraw-Hill; 1st edition (October 1, 1997)
  - Chapter 8



- Christopher M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer 2006
  - Section 2,5

47