Nearest Neighbors

João Gama
jgama@fep.up.pt

LIAAD-INESC Porto, University of Porto, Portugal

Setembro 2020

Outline

Context

Predictive Learning:

- Given
  - examples of a function $(X, f(X))$
    $f(.)$ is unknown
  - Predict de value $f(X)$ for X, not seen before
- Two different possibilities:
  - Classification:
    $f(X) \in \{c_1, \ldots, c_n\}$
    the domain of $f(x)$ is an undordered discrete set;
  - Regression: $f(X) \in R$
    the domain of $f(x)$ is a subset of $\Re$.

Tempo	Temperatu	Humidade	vento	Joga
Sol	85	85	Não	Não
Sol	80	90	Sim	Não
Nublado	83	86	Não	Sim
Chuva	70	96	Não	Sim
Chuva	68	80	Não	Sim
Chuva	65	70	Sim	Não
Nublado	64	65	Sim	Sim
Sol	72	95	Não	Não
Sol	69	70	Não	Sim
Chuva	75	80	Não	Sim
Sol	75	70	Sim	Sim
Nublado	72	90	Sim	Sim
Nublado	81	75	Não	Sim
Chuva	71	91	Sim	Não

Peso	Distancia	Efeito
-1.48334449	1.4139718	0.8001842
0.06711704	-0.3090329	2.4637740
0.78459210	0.6591077	0.2712122
-0.55427611	1.4456181	1.1274092

Motivation

- The nearest-neighbour algorithm is one of the simplest data mining algorithms.
- Intuition:
  Objects of the same concept are similar to each other.
  Examples of the same class are close to each other.

Motivation

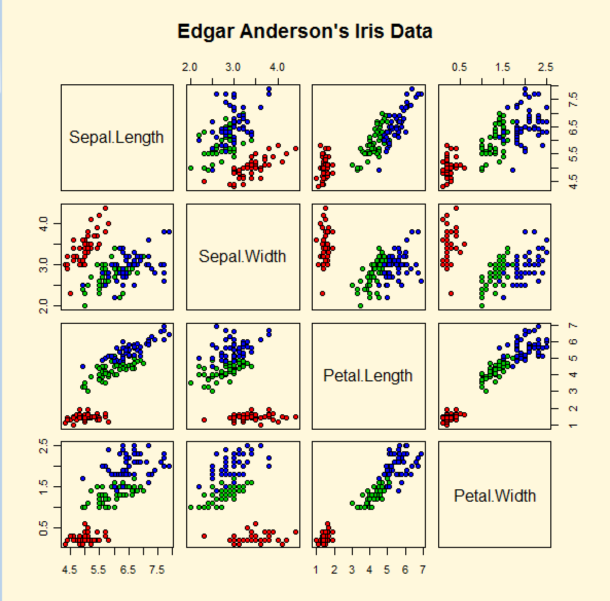
- The algorithm:
  - Each example represents a point in the space defined by the attributes;
  - classifies objects based on closeness to the examples in the training set;
- Characteristics
  - lazy algorithm. Does not learn a compact model for the training data;
  - only memorize training examples;
  - It can be used both for classification or regression.

The *Iris* dataset

```
> data(iris)
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1
>
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica
6.3	2.9	5.6	1.8	virginica

The instance Space



Outline

Base Idea

- Each example represents a point in space defined by the attributes.
- Define a metric in this space:
  - The most common metric: Euclidean distance $d(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$
- Given a test example, select the closest training example. Classify the test example in the class of the closest training example.

Metrics

- The most common metric: Euclidean distance
  $d(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$
- Proprieties:
  1. identity: $D(Q, Q) = 0$;
  2. is always non negative: $D(Q, S) \geq 0$;
  3. is symmetric: $D(Q, S) = D(S, Q)$;
  4. satisfies the triangular inequality:
     $D(Q, S) + D(S, T) \geq D(Q, T)$.
- It is additive: assumes the independence of attributes.

Distances

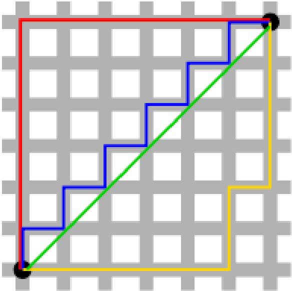
- Numeric Attributes (p-norm)

$$L^p(\vec{x},\vec{y}) = \sqrt[p]{\sum |x_i - y_i|^p}$$

Manhattan:

$$L(\vec{x},\vec{y}) = \sum |x_i - y_i|$$

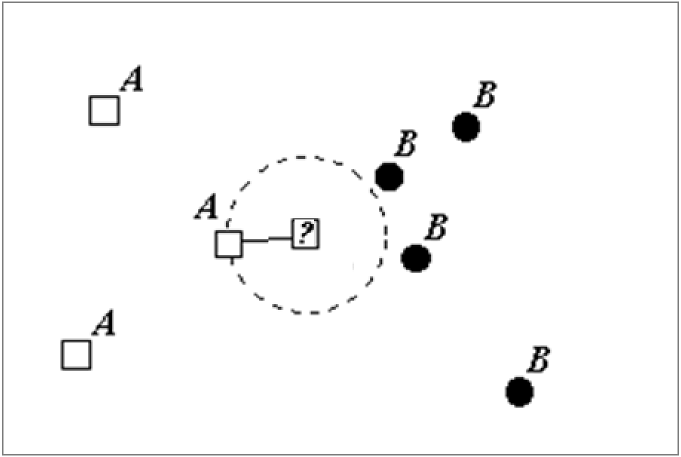
- Nominal Attributes
  Hamming Distance
  - $d(x_i, x_j) = 0$ sse $x_i = x_j$
  - $d(x_i, x_j) = 1$ sse $x_i \neq x_j$



The 1-nearest Neighbour algorithm

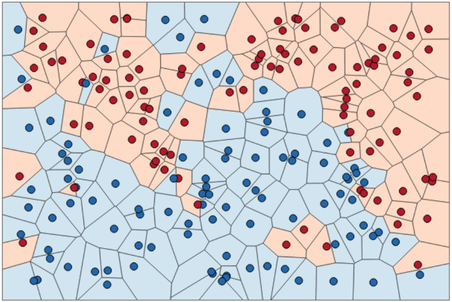
- Learning Algorithm:
  - For each training example $\{\vec{x}_i, y_i\}$
  - Memorize the example
- Applying the algorithm:
  - Given a test point $\{x_q, ?\}$:
  - Compute the distance o the point (x_q) to each training example;
  - Let $\{x_T, y_T\}$ the close training example.
  - Classify x_q: $y_q \leftarrow y_T$

Illustrative Example



The Decision Surface

The Voronoi Diagram



- Voronoi cell $x \in T$: set of points whose distance to x is less than the distance to any other point
- The decision surface is a set of convex polyhedra containing each of the training examples

Distances

What is the impact, in the distance function, of representing an attribute in cm or Km?

To avoid the impact in the distance function: normalize attributes:

- Subtract the mean and divide by the standard deviation
  all attributes with mean 0 and standard deviation 1.
- Divide attribute values by the range.

Outline

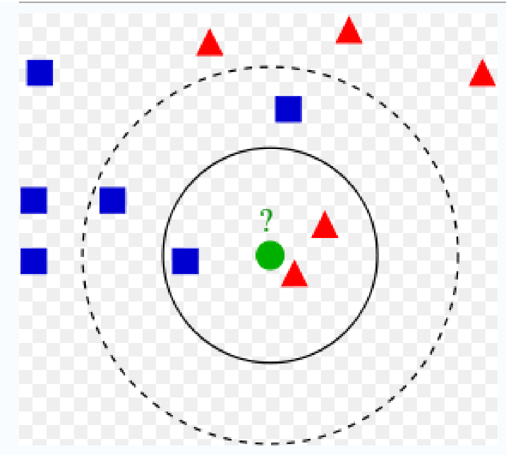
The k-nearest Neighbour algorithm

Select, from the training set, the k nearest exemplars.

- In Classification problems:
  - Each neighbour votes for one class.
  - Select the most voted class.
  - which is equivalent to:
    $f(x_T) \leftarrow moda(f(x_1), f(x_2), \ldots, f(x_k))$
  - **The constant that minimizes the 0-1 loss function is the mode.**
- In regression problems:
  - $f(x_T) \leftarrow mean(f(x_1), f(x_2), \ldots, f(x_k))$
  - **The constant that minimizes the square error is the *mean*;**
  - $f(x_T) \leftarrow median(f(x_1), f(x_2), \ldots, f(x_k))$
  - **The constant that minimizes the absolute error is the *median*.**

Illustrative Example

k = 3 e k =5



Which value for k?

- Usually small odd numbers (k=3, 5,...).
- Estimate k using cross-validation.
- Associate a weight to the vote of each neighbour
  - Weigh the contribution of each of the k neighbours inversely proportional to the distance.
  - In classification problems:
    - Weighted mode: $y_t = argmax \sum_i^k w_i \delta(c, y_i)$ with $w_i = \frac{1}{d(x_t, x_i)}$
  - In regression problems:
    - Weighted mean: $y_t = \frac{\sum_{i=1}^k w_i y_i}{\sum w_i}$ with $w_i = \frac{1}{d(x_t, x_i)}$
  - In this way, it is possible to use $k = m$ (all the training examples).

Outline

Analysis of the Algorithm

The k-nearest neighbour is one of the paradigms of inductive learning: *objects with similar characteristics belong to the same group*.

Positive

- The learning phase consists of memorizing the examples;
- Applicable even in complex problems;
- Can be used both in classification and regression;
- Naturally Incremental ;
- behaviour in the limit:
  For an infinite number of examples, the error of 1NN is bounded by twice the Bayes Optimal error.

Behaviour in the limit

Given

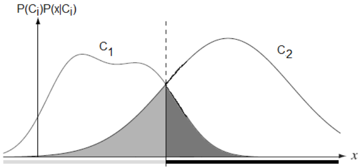
- $e(x)$: error of the optimal classifier
- $e_{1nn}(x)$: error of the 1-nearest neighbour

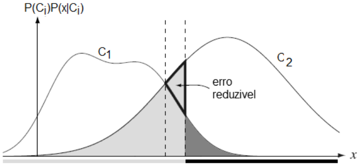
We can prove:

- Theorem: $lim_{n->\infty} e_{1nn}(x) <= 2 * e(x)$
- Theorem: $lim_{n->\infty, k->n} e_{kNN}(x) = e(x)$

For an infinite number of examples, the error of the k-NN is bounded by the Bayes Optimal error.

Bayes Optimal error





Analysis of the Algorithm

Negative

- Do not get a compact representation of examples: *lazy* algorithm;
- high application time: calculates the distance between the test example and all training examples.
- is affected by the presence of redundant and irrelevant attributes;
- The course of dimensionality.

The course of dimensionality

Consider 100 points uniformly distributed:

- In a square with a side of 1 unit;
- In a cube with side 1 unit;
- ...

(The number of attributes defines the number of dimensions of space)

We compute the average distance between any two points:

Nr. Dimensions	Average Distance
2	0.494
3	0.647
4	0.7717
5	0.875
...	
10	1.28

Increasing the size to keep the average distance between the points is necessary to increase exponentially the number of points.

The course of dimensionality

Removing irrelevant attributes

- Forward selection
- Backward elimination
- Associate weights to the attributes

Outline

Developments

Long application time: calculates the distance between the test example and all training examples.

Reducing the search space:

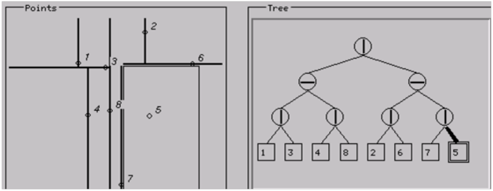
- Obtain representative examples
  - Remove redundant examples
  - Remove examples where all the neighbours are of the same class
- Remove noisy examples
  - Remove examples where all the neighbours are of other class.

Edited k-NN

- **Function Edited k-NN(*Exs*)**
  - For each example $x \in Exs$
    - If x is correctly classified by $Exs - \{\text{x}\}$
      Then remove x from Exs
- **Function Edited k-NN(*Exs*)**
  - $\text{E} = \{\}$
  - For each example $x \in Exs$
    - If x is misclassified by E
      Then Add x to E

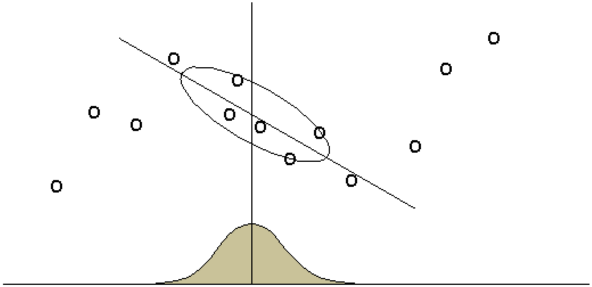
k-Dimensional Trees

kd-trees is a space-partitioning data structure for organizing points in a k-dimensional space.



Locally Weighted Regression

In locally weighted regression, points are weighted by proximity to the current x in question using a kernel. A regression is then computed using the weighted points.



C. Atkeson, A. Schaal, A. Moore; *Locally weighted learning*, AI Review, 1997 Radial basis Function Networks

Case Base Reasoning

Case-based reasoning (CBR), is the process of solving new problems based on the solutions of similar past problems.

- An auto mechanic who fixes an engine by recalling another car that exhibited similar symptoms is using case-based reasoning.
- A lawyer who advocates a particular outcome in a trial based on legal precedents or a judge who creates case law is using case-based reasoning.

A. Aamodt, E. Plazas, *Case-Based Reasoning: Foundational issues, methodological variations, and system approaches*, AI Communications Vol. 7(1), 1994

Bibliography

- D.Aha, D.Kibler,M.Albert, *Instance-based learning algorithms*, Machine Learning, Vol.6, 1991
- C. Atkeson, A. Schaal, A. Moore; *Locally weighted learning*, AI Review, 1997 Radial basis Function Networks
- M. Powell, *Radial basis functions for multivariate interpolation: a review*, in Algorithms for approximation, Clarendon Press, 1987
- A. Aamodt, E. Plazas, *Case-Based Reasoning: Foundational issues, methodological variations, and system approaches*, AI Communications Vol. 7(1), 1994