Machine Learning in High Dimension
IA317
Nearest Neighbors

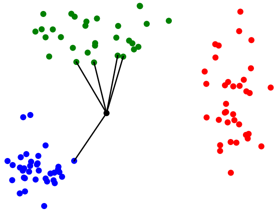
Thomas Bonald

2023 – 2024

Nearest neighbors

A set of **methods** for

- Classification
- Regression
- Clustering
- Anomaly detection



Advantages

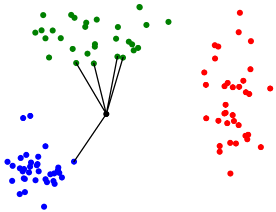
- Simple
- Explainable

Issues

- Choice of distance
- Complexity

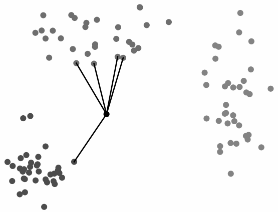
Classification

Use **majority vote** of k nearest neighbors in the training set



Regression

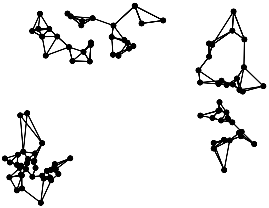
Use **(weighted) average** of k nearest neighbors in the training set



Clustering

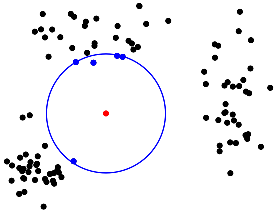
Two steps:

1. Build the **graph** of nearest neighbors
   (k nearest neighbors or distance $< d$)
2. **Cluster** the graph (e.g., through Louvain)



Anomaly detection

Detection of isolated samples by the estimation of **local density**



Outline

1. Review of **distances**
   What is meant by nearest neighbors?
2. **Search** algorithms
   How to find the nearest neighbors?

Review of distances

Distances in **vector** spaces
$\rightarrow$ numerical feat.

- Euclidean
- Manhattan
- Cosine similarity

Distances between **probability distributions**
$\rightarrow$ positive feat. $+$ normalization, numerical feat. $+$ softmax

- Hellinger distance
- Jensen-Shannon divergence

Distances between **sets**
$\rightarrow$ binary / categorical features, numerical feat. $+$ threshold

- Hamming distance
- Jaccard index

Norm distances

Let $x, y \in \mathbb{R}^d$

Euclidean distance

$$d(x, y) = ||x - y||$$

where $|| \cdot ||$ refers to the L2 norm.

Manhattan distance

$$d(x, y) = |x - y|$$

where $| \cdot |$ refers to the L1 norm.

Cosine similarity

Let $x, y \in \mathbb{R}^d \setminus \{0\}$

Cosine similarity

$$s(x, y) = \cos(x, y) = \frac{x.y}{\|x\|\|y\|} \in [-1, 1]$$

Equivalent to the **Euclidean distance** on the unit sphere:

$$\|\bar{x} - \bar{y}\|^2 = 2(1 - s(x, y)) \in [0, 2]$$

with $\bar{x}, \bar{y}$ the projections of x, y on the unit sphere:

$$\bar{x} = \frac{x}{\|x\|} \quad \bar{y} = \frac{y}{\|y\|}$$

Hellinger distance

Let p, q be discrete **probability distributions**.

Hellinger distance

$$d(p,q) = \frac{1}{\sqrt{2}} \|\sqrt{p} - \sqrt{q}\| \in [0,1]$$

Equivalent to the **cosine similarity** between $\sqrt{p}$ and $\sqrt{q}$:

$$d(p,q) = \sqrt{1 - \cos(\sqrt{p}, \sqrt{q})},$$

known as the **Bhattacharyya coefficient** between p and q:

$$\cos(\sqrt{p}, \sqrt{q}) = \frac{\sqrt{p}.\sqrt{q}}{||\sqrt{p}||||\sqrt{q}||} = \sum_{i=1}^{d} \sqrt{p_i q_i}.$$

Jensen-Shannon divergence

Let p, q be discrete **probability distributions**.

Jensen-Shannon divergence

$$d(p,q) = H\left(\frac{p+q}{2}\right) - \frac{H(p)+H(q)}{2} \in [0,1]$$

where H is the entropy (base 2).

We have:

$$d(p,q) = \frac{1}{2}\left(D(p||\ \frac{p+q}{2}) + D(q||\ \frac{p+q}{2})\right)$$

where D denotes the **Kullback-Leibler divergence**:

$$D(p||q) = \sum_i p_i \log \frac{p_i}{q_i} \geq 0$$

Hamming distance

Let $x, y \in \{0, 1\}^d$
Viewed as subsets A, B of $\{1, \ldots, d\}$.

Hamming distance

$$d(A, B) = |A \Delta B| \in [0, d]$$

where $A \Delta B$ is the symmetric difference between A and B

Expressed as:

$$d(x, y) = |x - y|$$

Jaccard distance

Let $x, y \in \{0,1\}^d$
Viewed as subsets A, B of $\{1, \ldots, d\}$.

Jaccard distance

$$d(A, B) = \frac{|A \Delta B|}{|A \cup B|} \in [0, 1]$$

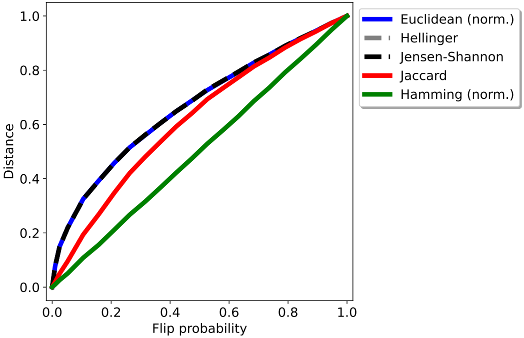
Expressed as:

$$d(x, y) = \frac{|x - y|}{|x \vee y|} \in [0, 1]$$

Example

Average distance between binary vectors $x, y \in \{0,1\}^{100}$

- $x = (1, \ldots, 1, 0, \ldots, 0)$
- $y = x$ with i.i.d. bit flips



Metric

A distance $d(x, y)$ is a **metric** if and only if:

- **Positivity & Identity**

$$d(x, y) \geq 0 \text{ and } d(x, y) = 0 \text{ if and only if } x = y$$

- **Symmetry**

$$d(x, y) = d(y, x)$$

- **Triangle inequality**

$$d(x, y) \leq d(x, z) + d(z, y)$$

Which distances are metrics?

Distance	Metric	Condition
Euclidean	✓	
Manhattan	✓	
Cosine	(✓)	$\frac{x}{\|\|x\|\|}, x \neq 0$
Hellinger	(✓)	$\sqrt{p}$
Jensen-Shannon	(✓)	$\sqrt{d(p,q)}$
Hamming	✓	
Jaccard	✓	

✓ Yes

(✓) Under condition

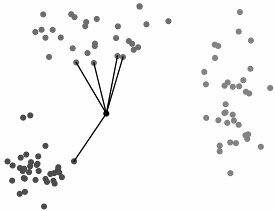
Outline

1. Review of **distances**
   What is meant by nearest neighbors?
2. **Search** algorithms
   How to find the nearest neighbors?

Nearest neighbor search

How to find the k nearest neighbors of a sample?

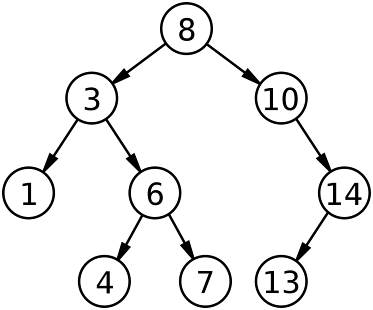
- **Sequential search** $O(n)$
- **Tree search**
  Construction $O(n \log n)$
  Search $O(\log n)$ (in best cases)



Binary tree search

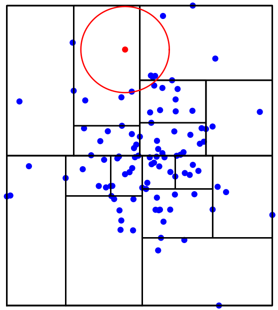
For 1-d data, e.g.,

$$\{8, 3, 1, 6, 10, 14, 4, 13, 7\}$$

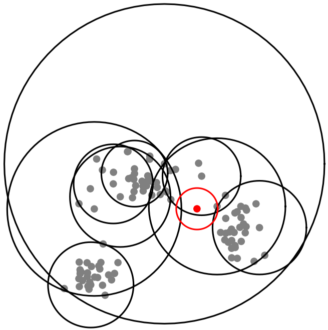


Tree search

1. KD tree

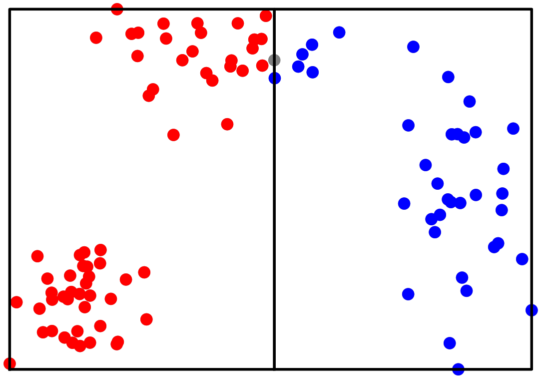


2. Ball tree



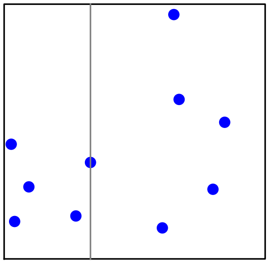
Bentley 1975

KD tree

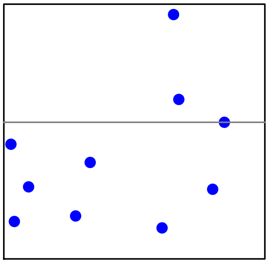


Cut strategies

1. **Max variance**
+ median point

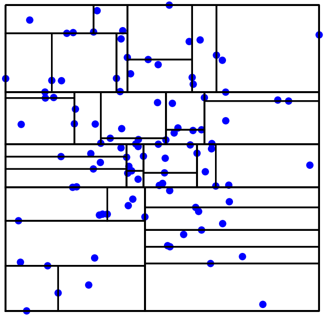


2. **Max spread**
+ middle point

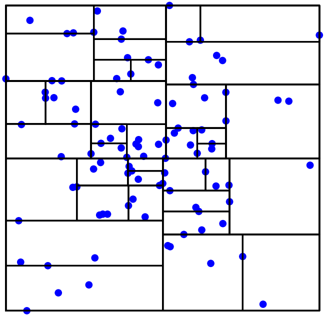


Example

1. Max variance

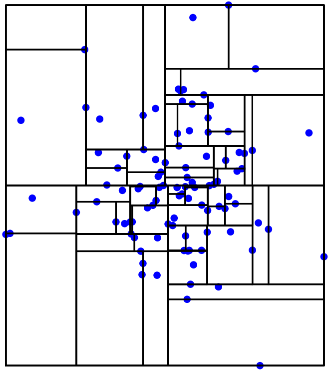


2. Max spread

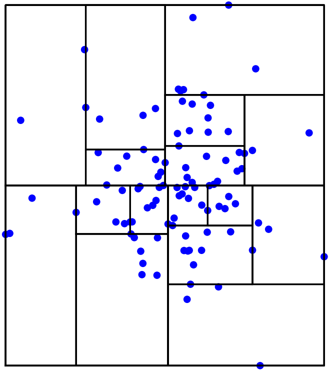


Pruning

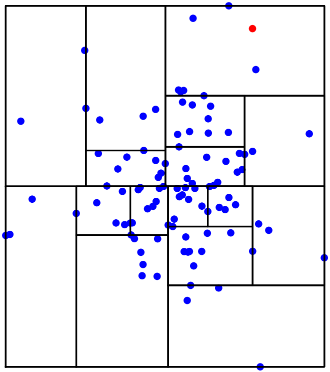
Leaf size = 1
(full tree)



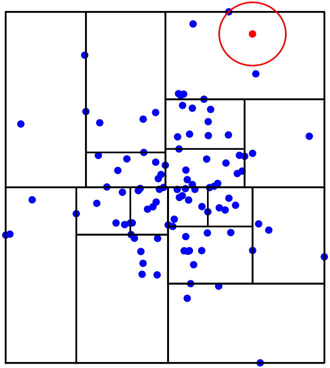
Leaf size $\leq$ 10
(pruned tree)



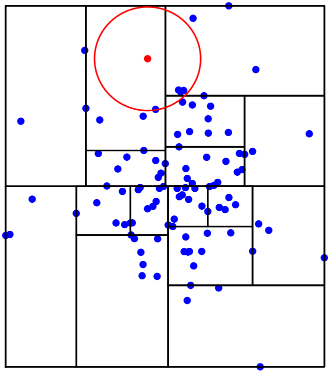
Nearest neighbor search



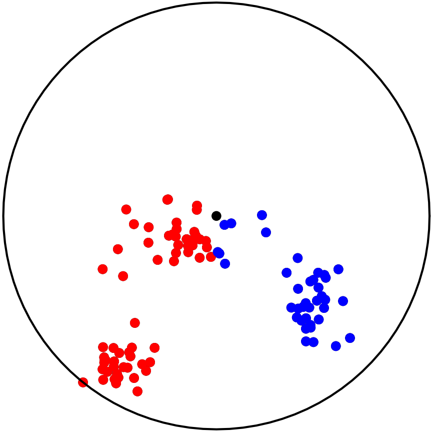
Nearest neighbor search



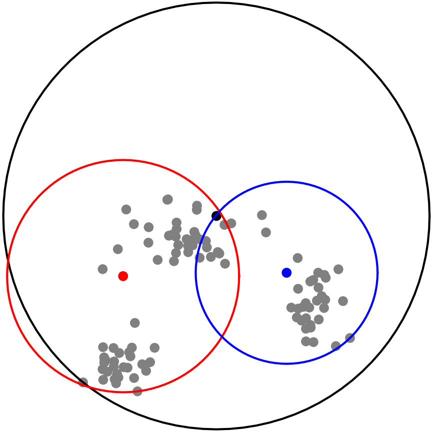
Nearest neighbor search



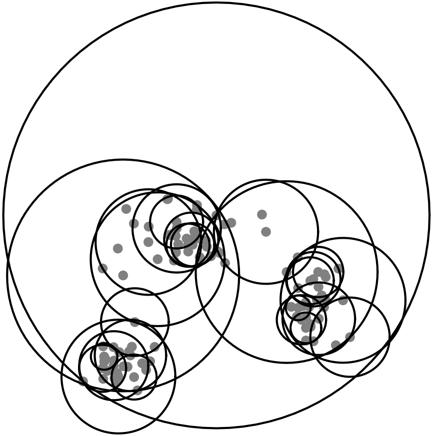
Ball tree



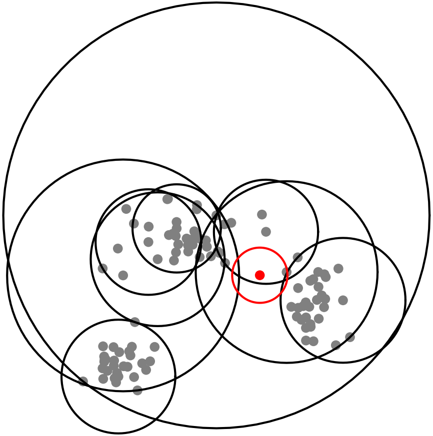
Ball tree



Ball tree

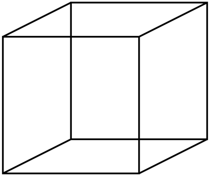


Nearest neighbor search



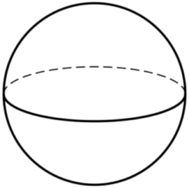
Boxes or balls?

Unit box



volume $= 1$

Unit ball



volume $= \dfrac{\pi^p}{p!}$ for $d = 2p$

Ball trees are more efficient than KD trees in high dimension

Complexity

Construction

- $O(n \log n)$ for both KD trees and Ball trees

Search

- $O(\log n)$ with Ball trees
- $O(\log n)$ (low dimension) up to $O(n)$ (high dimension) for KD trees

Comments

- Need for a **metric**
- Importance of **pruning**

Summary

Nearest neighbors

- A good **baseline**
  Efficient in high dimension
  Explainable
- **Applications**
  Classification, regression, clustering, anomaly detection
- **Distances**
  for numerical, categorical or binary features
  $\rightarrow$ importance of pre-processing / scaling
- **Search**
  Sequential search or tree search