

8. Nearest neighbors

Chloé-Agathe Azencott

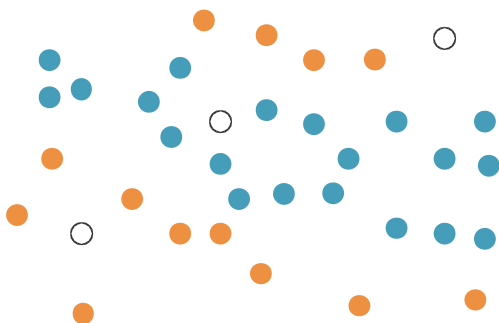
Centre for Computational Biology, Mines ParisTech
chloe-agathe.azencott@mines-paristech.fr



Learning objectives

- Implement the **nearest-neighbor** and **k-nearest-neighbors** algorithms.
- Compute **distances** between **real-valued vectors** as well as objects represented by **categorical features**.
- Define the **decision boundary** of the nearest-neighbor algorithm.
- Explain why kNN might not work well in **high dimension**.

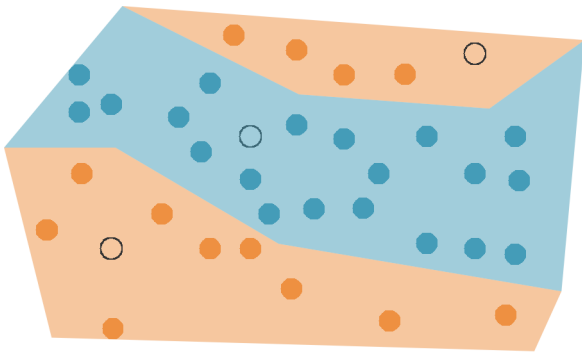
Nearest neighbors



- How would you color the blank circles?

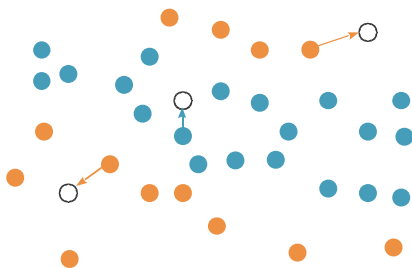
Partitioning the space

The training data partitions the entire space



Nearest neighbor

- **Learning:**
 - Store all the training examples
- **Prediction:**
 - For x : the label of the training example closest to it



k nearest neighbors

- **Learning:**
 - Store all the training examples
- **Prediction:**
 - Find the k training examples closest to x
 - **Classification**
 - **Regression**

Choice of k

- **Small k:** noisy
 - The idea behind using more than 1 neighbor is to average out the noise
- **Large k:** computationally intensive
 - If $k=n$:
 - Classification:
 - Regression:
- Set k by **cross-validation**
- Heuristic: $k \approx \sqrt{n}$

Non-parametric learning

Non-parametric learning algorithm:

- **the complexity of the decision function grows with the number of data points.**
- contrast with linear regression (\approx as many parameters as features).
- Usually: decision function is expressed directly in terms of the training examples.
- Examples:
 - kNN (this chapter)
 - tree-based methods (Chap. 9)
 - SVM (Chap. 10)

Instance-based learning

- **Learning:**
 - Storing training instances.
- **Predicting:**
 - Compute the label for a new instance based on its **similarity** with the stored instances.
- Also called **lazy learning**.
- Similar to **case-based reasoning**
 - Doctors treating a patient based on how patients with similar symptoms were treated,
 - Judges ruling court cases based on legal precedent.

Computing distances & similarities

- Distance

$$d : \mathcal{X} \rightarrow \mathbb{R}_+$$

- Euclidean distance

$$d(\mathbf{x}^1, \mathbf{x}^2) = \|\mathbf{x}^1 - \mathbf{x}^2\|_2 = \sqrt{\sum_{j=1}^p (x_j^1 - x_j^2)^2}$$

- Manhattan distance

$$d(\mathbf{x}^1, \mathbf{x}^2) = \|\mathbf{x}^1 - \mathbf{x}^2\|_1 = \sum_{j=1}^p |x_j^1 - x_j^2|$$

- Lq-norm

$$d(\mathbf{x}^1, \mathbf{x}^2) = \|\mathbf{x}^1 - \mathbf{x}^2\|_q = \left(\sum_{j=1}^p |x_j^1 - x_j^2|^q \right)^{1/q}$$

- L1 = Manhattan.

- L2 = Euclidean.

- L_∞

$$\mathbf{x} \in \mathbb{R}^p$$

Similarity between instances

$$s = \frac{1}{1 + d}$$

- Pearson's correlation

$$\rho(\mathbf{x}, \mathbf{z}) = \frac{\sum_{j=1}^p (x_j - \bar{x})(z_j - \bar{z})}{\sqrt{\sum_{j=1}^p (x_j - \bar{x})^2} \sqrt{\sum_{j=1}^p (z_j - \bar{z})^2}}$$

- Assuming the data is centered

$$\rho(\mathbf{x}, \mathbf{z}) = \frac{\sum_{j=1}^p x_j z_j}{\sqrt{\sum_{j=1}^p x_j^2} \sqrt{\sum_{j=1}^p z_j^2}}$$

$$\bar{x} = \frac{1}{p} \sum_{j=1}^p x_j$$

Geometric interpretation?

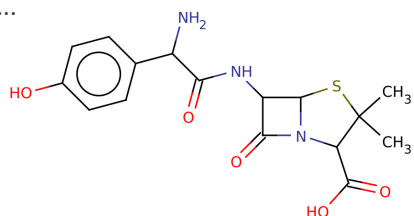
Categorical features

- Represent object as the list of **presence/absence** (or counts) of **features that appear in it**.

- **Example:** small molecules

features = atoms and bonds of a certain type

- C, H, S, O, N...
- O-H, O=C, C-N....



Binary representation

0 1 1 0 0 1 0 0 0 1 0 1 0 0 1

no occurrence
of the 1st feature

1+ occurrences
of the 10th feature

- **Hamming distance**

Number of bits that are different

$$d(\mathbf{x}^1, \mathbf{x}^2) = \sum_{j=1}^p (x_j^1 \text{ XOR } x_j^2)$$

Equivalent to

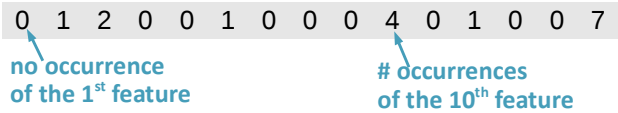
0 1 1 0 0 1 0 0 0 1 0 1 0 0 1

- **Tanimoto similarity**

Number of shared features (normalized)

$$s(\mathbf{x}^1, \mathbf{x}^2) = \frac{\sum_{j=1}^p (x_j^1 \text{ AND } x_j^2)}{\sum_{j=1}^p (x_j^1 \text{ OR } x_j^2)}$$

Counts representation



- MinMax similarity**

Number of shared features (normalized)

$$s(\mathbf{x}^1, \mathbf{x}^2) = \frac{\sum_{j=1}^p \min(x_j^1, x_j^2)}{\sum_{j=1}^p \max(x_j^1, x_j^2)}$$

If x is binary, MinMax and Tanimoto are equivalent

$$s(\mathbf{x}^1, \mathbf{x}^2) = \frac{\sum_{j=1}^p (x_j^1 \text{ AND } x_j^2)}{\sum_{j=1}^p (x_j^1 \text{ OR } x_j^2)}$$

Categorical features

- Features



- Compute the Hamming distance and Tanimoto and MinMax similarities between these objects:

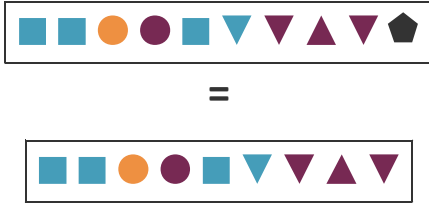


Categorical features

- Features

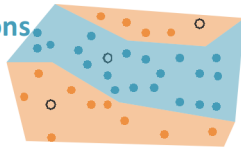


- When new data has unknown features: ignore them.



Advantages of kNN

- **Training is very fast**
 - Just store the training examples.
 - Can use smart indexing procedures to speed-up testing (slower training).
- **Keeps the training data**
 - Useful if we want to do something else with it.
- Rather robust to **noisy data** (averaging k votes)
- Can learn **complex functions**.

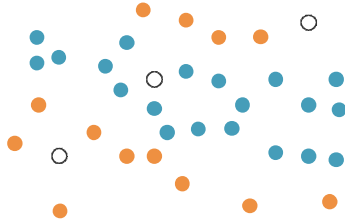


Drawbacks of kNN

- **Memory** requirements
- Prediction can be **slow**.
 - Complexity of labeling 1 new data point:
But kNN works best with lots of samples...
 - Efficient data structures (**k-D trees**)
 - Approximate solutions based on **hashing**
- kNN are fooled by **irrelevant attributes**.
 - E.g. $p=1000$, only 10 features are relevant; distances become meaningless.

Decision boundary of kNN

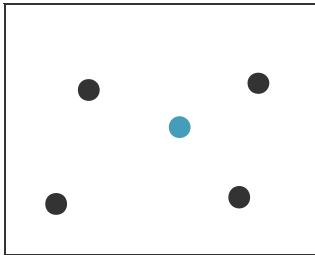
- Classification
- **Decision boundary:** Line separating the positive from negative regions.
- **What decision boundary is the kNN building?**



Voronoi tessellation

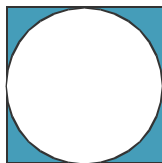
- **Voronoi cell** of x :
 - set of all points of the space closer to x than any other point of the training set
 - polyhedron
- **Voronoi tessellation** of the space: union of all Voronoi cells.

Draw the Voronoi cell of the blue dot.



Curse of dimensionality

- Remember from Chap 3
- When $p \nearrow$ the proportion of a hypercube outside of its inscribed hypersphere approaches 1.
- Volume of a p -sphere: $\frac{2r^2 \pi^{p/2}}{p \Gamma(p/2)}$



- What this means:
 - hyperspace is very big
 - all points are far apart
 - dimensionality reduction needed.

kNN variants

- **ε-ball neighbors**

- Instead of using the k nearest neighbors, use **all points within a distance ε** of the test point.
- What if there are no such points?

kNN variants

- **Weighted kNN**

- **Weigh the vote of each neighbor** according to the distance to the test point.

$$w_l = \exp\left(-\frac{1}{2}d(\mathbf{x}, \mathbf{x}^l)\right)$$

- Variant: **learn** the optimal weights [e.g. Swamidass, Azencott et al. 2009, **Influence Relevance Voter**]

Collaborative filtering

- **Collaborative filtering**: recommend items that similar users have liked in the past

similar users = users with similar tastes

- **item-based kNN**

- similarity between items: **adjusted cosine similarity**

Sum over the users that rated both item A and item B

$$s(A, B) = \frac{\sum_u (R(u, A) - \bar{R}(u))(R(u, B) - \bar{R}(u))}{\sqrt{\sum_u (R(u, A) - \bar{R}(u))^2 \sum_u (R(u, B) - \bar{R}(u))^2}}$$

Rating of item A by user u

Average rating by user u

– score of item A for user u:

$$S(u, A) = \frac{\sum_{B \in \mathcal{N}_u^k(A)} s(A, B) R(u, B)}{\sum_{B \in \mathcal{N}_u^k(A)} |s(A, B)|}$$

k nearest neighbors of A
according to s
among the items rated by user u

Summary

- **kNN**
 - very simple training
 - prediction can be expensive
- Relies on a “good” **distance/similarity** between instances
- **Decision boundary = Voronoi tessellation**
- **Curse of dimensionality:** the hyperspace is very big.

References

- *A Course in Machine Learning.*
http://ciml.info/dl/v0_99/ciml-v0_99-all.pdf
 - **kNN:** Chap 3.2 — 3.3
 - **Categorical variables:** Chap 3.1
 - **Curse of dimensionality:** Chap 3.5
- More on
 - **Kd-trees**
https://www.ricmu.edu/pub_files/pub1/moore_andrew_1991_1/moore_andrew_1991_1.pdf
 - **Voronoi tessellation**
<http://philogb.github.io/blog/2010/02/12/voronoi-tessellation/>