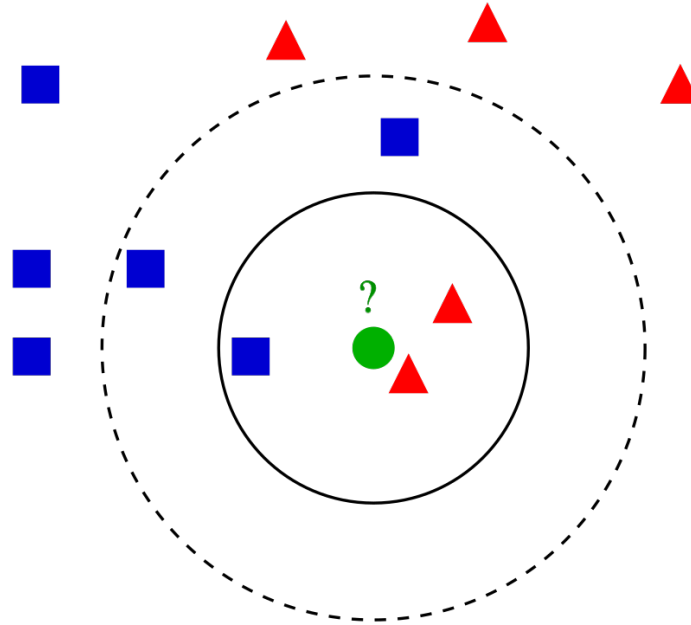


III. Nearest Neighbor Learning



*"If something is similar in some respects,
it's likely to be similar in other respects."*

-Patrick Winston

Introduction

Motivation



Introduction

Motivation

The image is a screenshot of a Google search interface. At the top, the Google logo is on the left, and the search bar contains the text "flower rose history meaning". To the right of the search bar are icons for clearing the search, voice search, image search, and a magnifying glass. Below the search bar, the section "Ähnliche Fragen" (Similar Questions) is displayed. It contains four questions, each with a downward arrow to its right: "What is the meaning of rose in history?", "What does the flower rose symbolize?", "What is the true meaning of roses?", and "What is the origin of rose flower?". Below these questions is a link that says "Feedback geben". Underneath the questions, there are two search results, each enclosed in a red rectangular box. The first result is from Wikipedia, titled "Rose symbolism - Wikipedia", and includes a brief description of rose meanings and links to related topics. The second result is from "www.freshflowers.com.au", titled "The Meaning, Symbolism and History of Red Roses", and includes a date and a brief description. Below these two results, there is a third search result from "www.flowerglossary.com", titled "History and Meaning of Red Roses - Flower Glossary", which includes a brief description.

Google

flower rose history meaning

Ähnliche Fragen

What is the meaning of rose in history?

What does the flower rose symbolize?

What is the true meaning of roses?

What is the origin of rose flower?

Feedback geben

https://en.wikipedia.org/wiki/Rose_symbolism · Diese Seite übersetzen

Rose symbolism - Wikipedia

Examples of common meanings of different coloured roses are: true love (red), mystery (blue), innocence or purity (white), death (black), friendship (yellow), ...

[In religion](#) · [In Europe](#) · [In North America](#) · [United States](#)

<https://www.freshflowers.com.au> · Diese Seite übersetzen

The Meaning, Symbolism and History of Red Roses

25.10.2018 — Deep red roses convey deep emotions, a bright red rose means romance & a red rosebud symbolises young love. Know the different meanings ...

<https://www.flowerglossary.com> · Diese Seite übersetzen

History and Meaning of Red Roses - Flower Glossary

In Greek mythology, roses were created by Aphrodite, the Goddess of Love. Greeks believed

Nearest Neighbor Learning

Principle

General properties:

- ❑ *Lazy learning*
The majority of computation is deferred to the prediction phase
- ❑ *Instance-based prediction*
There is no explicit model the prediction is based on; instead, prediction is based directly on the instances in the training data set
- ❑ *Non-parametric*
No assumption about analytical form of $f(\mathbf{x})$.
- ❑ *From a Bayesian perspective: Discriminative model*
Models posterior $p(f(\mathbf{x}) = c|\mathbf{x})$ directly.

Common Use-cases:

- ❑ *Recommender Systems: Collaborative Filtering*
- ❑ *Outlier Detection: Nearest Neighbor far away?*

(1-)Nearest Neighbor Algorithm

Pseudocode

Algorithm: *NN* Nearest Neighbor.

Input: D Training examples of the form (\mathbf{x}, y) with target value y .

d Distance measure in feature space; $d : X \times X \rightarrow \mathbb{R}^+$.

\mathbf{q} Query data point.

Output: $h(\mathbf{q})$ Prediction for \mathbf{q} .

NN – *Training*(D)

1. **FOR** i **IN** $1, \dots, n$ **DO** $// \quad |D| = n$
2. *store_training_example* $((\mathbf{x}, y)_i)$

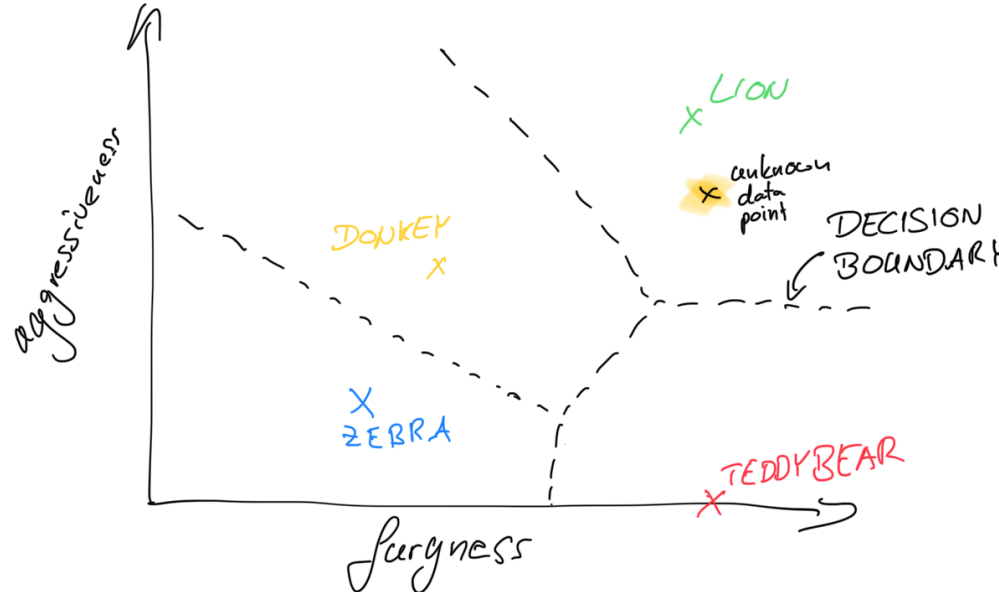
NN – *Prediction*(D, d, \mathbf{q})

1. $\text{closest_index} = \emptyset, \text{closest_distance} = \infty$
2. **FOR** i **IN** $1, \dots, n$ **DO**
3. $\text{distance} = d(\mathbf{q}, \mathbf{x}_i)$
4. **IF** $\text{distance} < \text{closest_distance}$ **DO**
5. $\text{closest_distance} = \text{distance}$
6. $\text{closest_index} = i$
7. **RETURN** y_i

Nearest Neighbor Learning

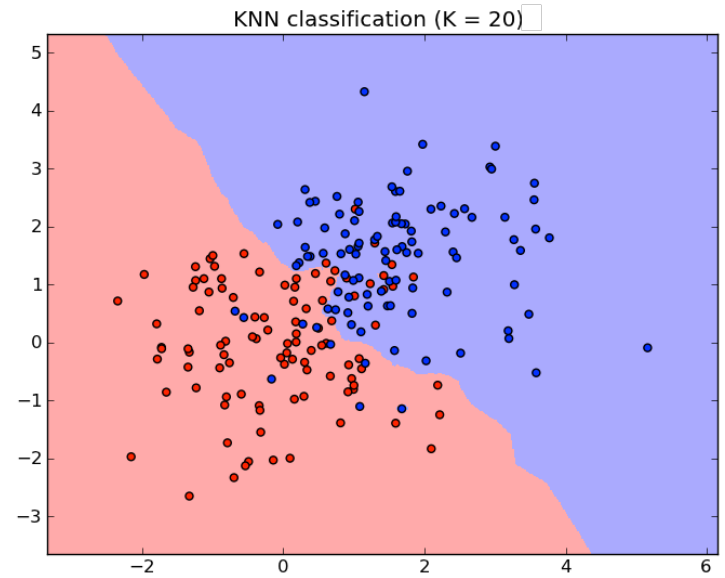
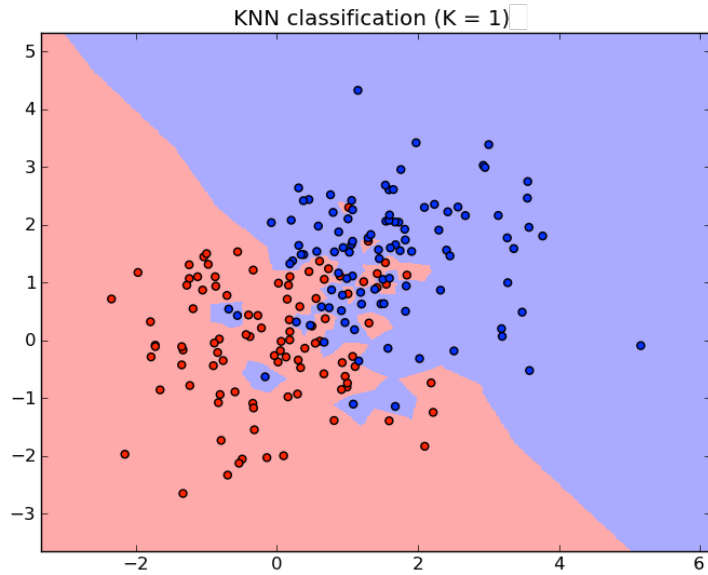
Illustration of Decision Boundary (1)

- ❑ Assuming Euclidean distance: $d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{k=1}^d (a_k - b_k)^2}$
- ❑ Decision boundary between any two training examples is a straight line.
A point on the line is *equidistant* from both training examples.
- ❑ Decision boundary globally is a set of connected convex polyhedra.
In 2-d also called: "Voronoi Tessellation"
Obtained by removing line segments that separate examples from the same class.



Nearest Neighbor Learning

Illustration of decision boundary (2)

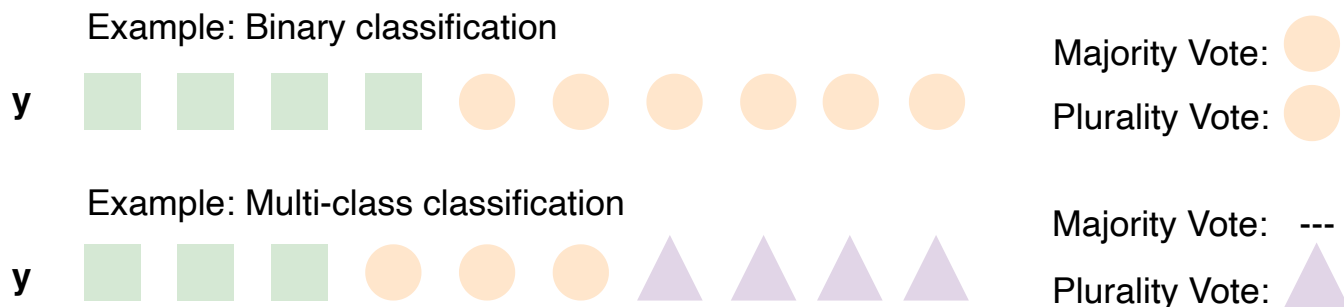


k -Nearest Neighbor Learning

Classification

The k -NNs prediction is based on the k most similar (\approx nearest) training points. The predicted class label is obtained via plurality voting among k neighbors.

Oftentimes called "majority" voting. But no absolute "majority" required in multi-class settings.



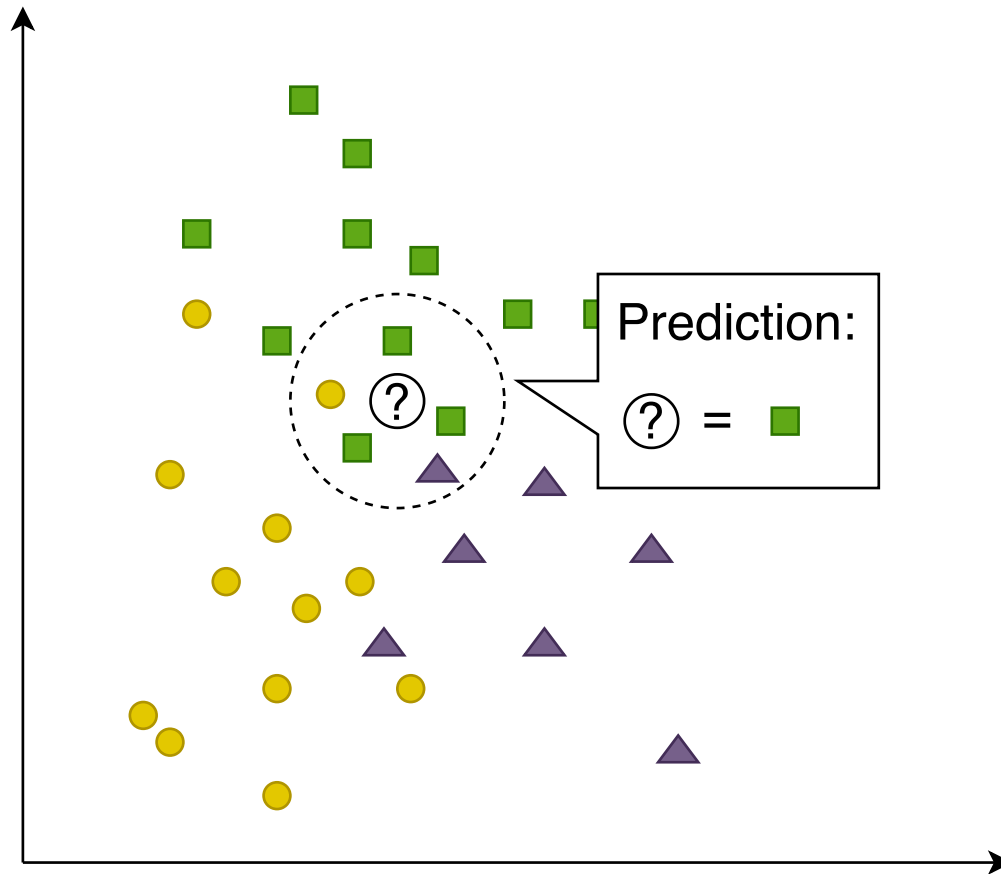
Given a target function $f : \mathbb{R}^d \rightarrow \{1, \dots, C\}$, that assigns a class label $y \in \{1, \dots, C\}$ to every data point \mathbf{x} : $f(\mathbf{x}) = y$, the set of k nearest neighbors is denoted by $D_{\mathbf{q}} = \{(\mathbf{x}_i, y_i)\}_{i=1, \dots, k}$.

For a given query point \mathbf{q} , the plurality vote of a k -NN classifier is

$$h(\mathbf{q}) = \underset{c \in \{1, \dots, C\}}{\operatorname{argmin}} \sum_{i=1}^k \mathbf{1}[c = f(\mathbf{x}_i)]$$

k -Nearest Neighbor Learning

Classification: Plurality voting



5-Nearest Neighbor plurality voting in a 3-class problem.

k -Nearest Neighbor Learning

Regression

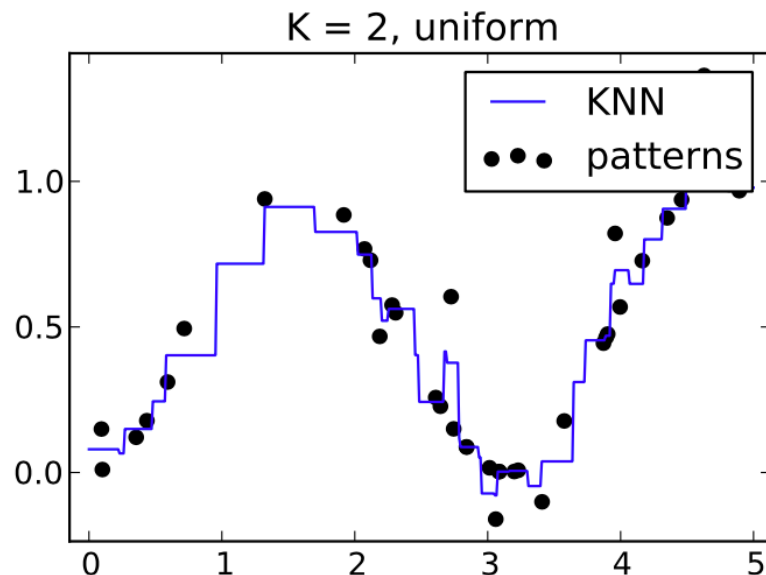
In a regression setting, the target function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ maps to real values. A common approach for combining the target values of the k neighbors is to compute the mean

$$h(\mathbf{q}) = \frac{1}{k} \sum_{i=1}^k f(\mathbf{x}_i)$$

Alternative aggregation functions:

- ❑ Median: Less sensitive to outliers
- ❑ Weighted: Target values weighted by a function of the distance.
E.g. the weight of neighbor i is inversely proportional to its distance to \mathbf{q} :

$$w_i = \frac{1}{d(\mathbf{q}, \mathbf{x}_i)^2}$$



k -Nearest Neighbour (kNN) classifier

Pseudocode

Algorithm: *KNN* k -Nearest Neighbor.

Input: D Training examples of the form $(\mathbf{x}, f(\mathbf{x}))$ with class value y_i for \mathbf{x}_i .

k Number of nearest neighbors to consider.

d A distance measure in feature space; $d : X \times X \rightarrow \mathbb{R}^+$.

\mathbf{q} A query data point to classify.

Output: $h(\mathbf{q})$ Predicted class of \mathbf{q} .

$KNN(D, k, d, \mathbf{q})$

1. $U = \emptyset, \mathcal{N}_k = \emptyset$
2. **FOR** (\mathbf{x}_i, y_i) **IN** D **DO**
3. $U = U \cup \{(d(\mathbf{q}, \mathbf{x}_i), y_i)\}$
4. **ENDDO**
5. $U = \text{sort_distances_ascending}(U)$
6. $\mathcal{N}_k = \text{select_top_k}(U, k)$
7. $h(\mathbf{q}) = \text{find_most_frequent_class}(\mathcal{N}_k)$
8. **RETURN** $h(\mathbf{q})$

k -Nearest Neighbour (kNN) classifier

Pseudocode

Algorithm: *KNN* k -Nearest Neighbor.

Input: D Training examples of the form $(\mathbf{x}, f(\mathbf{x}))$ with class value y_i for \mathbf{x}_i .

k Number of nearest neighbors to consider.

d A distance measure in feature space; $d : X \times X \rightarrow \mathbb{R}^+$.

\mathbf{q} A query data point to classify.

Output: $h(\mathbf{q})$ Predicted class of \mathbf{q} .

$KNN(D, k, d, \mathbf{q})$

1. $U = \emptyset, \mathcal{N}_k = \emptyset$
2. **FOR** (\mathbf{x}_i, y_i) **IN** D **DO**
3. $U = U \cup \{(d(\mathbf{q}, \mathbf{x}_i), y_i)\}$
4. **ENDDO**
5. $U = \text{sort_distances_ascending}(U)$
6. $\mathcal{N}_k = \text{select_top_k}(U, k)$
7. $h(\mathbf{q}) = \text{find_most_frequent_class}(\mathcal{N}_k)$
8. **RETURN** $h(\mathbf{q})$

Issues/Bottlenecks?

Improving Computational Performance

Runtime Complexity

The Landau-notation ("Big- \mathcal{O} ") provides a formalism to express an upper bound (worst case) on the asymptotic growth rate of a function.

- ❑ In computer science, the Big- \mathcal{O} notation is used to quantify runtime (time complexity) or memory usage (space complexity) of an algorithm.
- ❑ In a naive implementation, the k -NN prediction phase has to execute $n \cdot d$ comparisons of constant cost. Since typically $n \gg d$, the complexity of the nearest neighbor algorithm is $\mathcal{O}(n)$.
- ❑ *...for every single query point!*

Strategies for faster prediction:

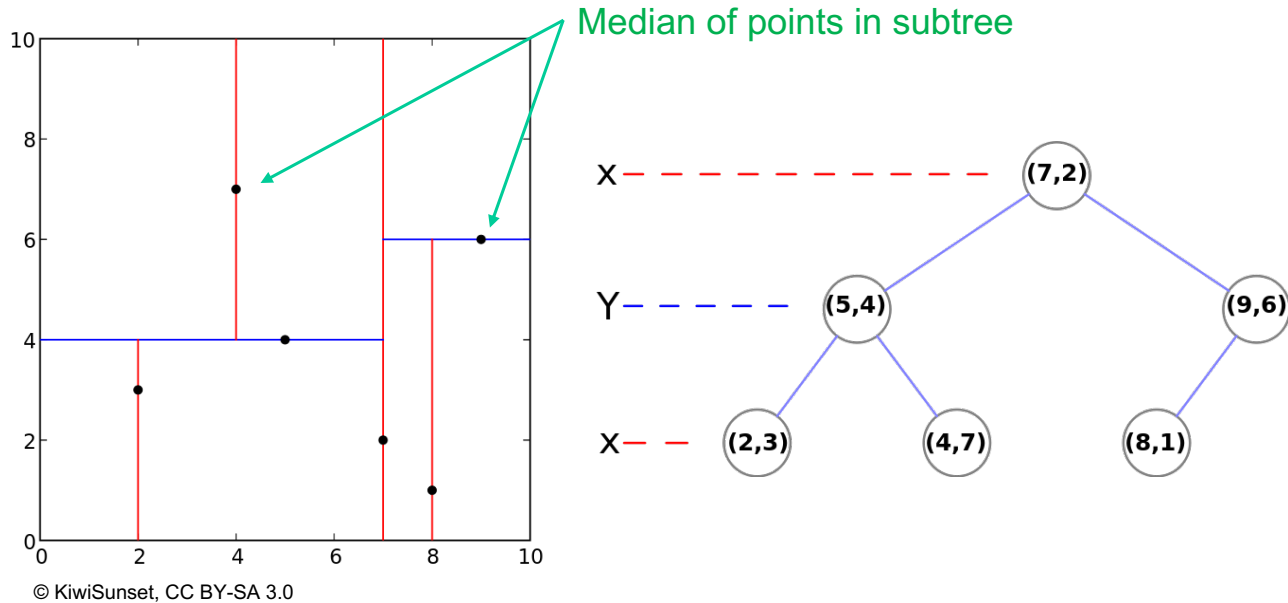
- ❑ Simple tweak to avoid sorting: Priority Queue
- ❑ Store training examples in range based data structures such as e.g. KD-Trees or Ball-Trees
- ❑ Faster distance metric or heuristic
- ❑ Pruning
- ❑ Parallelization

Improving Computational Performance

Optimized Storage of Training Examples

Use a k-d tree to index regions in feature space:

- ❑ Multi-dimensional Binary Search Tree (BST)
- ❑ Maintain leaf counts on nodes



Improving Computational Performance

Pruning

Editing:

- ❑ Permanently remove training examples that do not affect decision boundary.
- ❑ Single examples surrounded by examples from a different class don't affect decision boundary in plurality voting.

Prototypes:

- ❑ Replace selected examples by "prototypes" that summarize multiple examples in dense regions.
- ❑ E.g. k -Means clustering to obtain "prototypes"

Improving Predictive Performance

Model Selection

There are a couple of ways to tune the predictive performance of the k -NN algorithm:

- ❑ Choosing the value of k
- ❑ Scaling feature axes
- ❑ Choice of distance measure
- ❑ Weighting scheme for distance measure

→ Model selection (aka hyper-parameter tuning) using cross-validation

Distance Measures

Consider two points \mathbf{u}, \mathbf{v} in feature space $X \subset \mathbb{R}^l$:

- Manhattan distance (L_1 -Norm)

$$d(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^l |u_i - v_i| \quad (1)$$

- Euclidean distance (L_2 -Norm)

$$d(\mathbf{u}, \mathbf{v}) = \sqrt{\sum_{i=1}^l (u_i - v_i)^2} \quad (2)$$

- Maximum distance (L_∞ -Norm)

$$d(\mathbf{u}, \mathbf{v}) = \max\{|u_i - v_i|\}_{i=1}^l \quad (3)$$

- Minkowski distance (L_p -Norm): A generalized version of L -Norms.

$$d(\mathbf{u}, \mathbf{v}) = \sqrt[p]{\sum_{i=1}^l |u_i - v_i|^p} \quad (4)$$

Distance Measures

- ❑ Cosine similarity¹

$$\cos(\theta) = \frac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|} \quad (5)$$

- ❑ Jaccard Index
- ❑ Hamming distance
- ❑ Mahalanobis distance

¹Not a proper distance metric; Violates triangle inequality.

k -Nearest Neighbor Learning

Probabilistic interpretation

- The k -Nearest Neighbor algorithm estimates the conditional probabilities $p(c|\mathbf{x})$ by counting the data points in class c in the neighbourhood $\mathcal{N}_k(\mathbf{x})$ of \mathbf{x} :

$$p(c|\mathbf{x}) = \frac{|\{\mathbf{x}' \in \mathcal{N}_k(\mathbf{x}) | f(\mathbf{x}') = c\}|}{k} \quad (6)$$

- Bias-variance trade-off
 - Small k leads to large variance in test errors and unreliable classification
 - Large k leads to large bias and inaccurate classification

Challenges

- ❑ Spread problem: Input variables have different variances.
→ Normalize values on each dimension to unit variance $x' = \frac{x}{\sigma_x}$
- ❑ Irrelevant dimension problem: The output is independent of one of the input variables.

Confusing results, since the irrelevant variable contributes to the distance measure in the same way as any relevant variable.

→ Exclude irrelevant variable

- ❑ Irrelevant data problem: The output does not depend on the input variables at all.

Can we predict whether a person is going to go bankrupt based on their physical appearance?

→ "No cake without flour".

Summary

- ❑ k NN is a **lazy, instance-based, non-parametric** supervised learning algorithm

Non-parametric: No assumptions about the analytical form of $f(x) = y$.

- ❑ Able to represent highly non-linear decision boundaries
- ❑ Efficient implementation requires special data structures
- ❑ Distance metric needs to be chosen carefully (validated)
- ❑ Simple and (surprisingly) great for small training data sets

→ kNN in *scikit-learn*: <https://scikit-learn.org/stable/modules/neighbors.html>