Week 11-1: K-Nearest Neighbors (k-NN)

Ekarat Rattagan

April 4, 2025

Supervised Learning: Overview

- **Regression**:
  - Linear Regression (LR)
  - Cost Function: Mean Squared Error (MSE)
  - Evaluation Metric: MSE
- **Classification**:
  - Logistic Regression
  - Cost: Negative Log-Likelihood (NLL)
  - Evaluation Metric: Confusion Matrix

k-NN

Assumption: Similar inputs should have similar outputs.

- Cost Function: ?
- Evaluation Metric: Confusion Matrix

Properties:

- Non-parametric model
- Lazy learning / instance-based method
- Works for both classification and regression tasks

k-NN Algorithm

1. Find the Euclidean distance between the testing data and each training data point.
2. Sort the distances from minimum to maximum
3. Pick k nearest neighbors
4. Use majority vote for classification

Euclidean Distance (L_2):

$$d(x_i, x_k) = \sqrt{\sum_{j=1}^{d} (x_{i,j} - x_{k,j})^2}$$

k-NN for Regression and Classification

Sample Data:

Sample	x_1	x_2	y_1	y_2
S1	1	5	1000	yes
S2	2	6	1200	yes
S3	3	1	1100	no
S4	2	4	2000	yes
S5	**2**	**5**	?	?

Predict: Use average y values of k-nearest neighbors to estimate for S5.

k-NN with Categorical Data

Example Dataset

Sample	Food	Chat	Fast	Price	Bar	Tip
S1	great	Y	Y	normal	no	yes
S2	g	N	Y	normal	no	yes
S3	m	Y	N	high	no	no
S4	g	Y	Y	normal	yes	yes
S5	great	no	no	normal	no	?

Distance: Hamming Distance Match $= 0$, Mismatch $= 1$

Define $k = 2$, then compute:

- $H(S5, S1)$
- $H(S5, S2)$
- $H(S5, S3)$
- $H(S5, S4)$
- Choose two nearest neighbors with lowest Hamming distances

Definition: Hamming Distance

Definition

The Hamming distance between two equal-length strings of symbols is the number of positions at which the corresponding symbols are different.

Examples

The symbols may be letters, bits, or decimal digits, among other possibilities. For example, the Hamming distance between:

- "**karolin**" and "**kathrin**" is 3.
- "**karolin**" and "**kerstin**" is 3.
- "**kathrin**" and "**kerstin**" is 4.
- **0000** and **1111** is 4.
- **2173896** and **2233796** is 3.

Ref: https://en.wikipedia.org/wiki/Hamming_distance

Tuning the Hyperparameter k

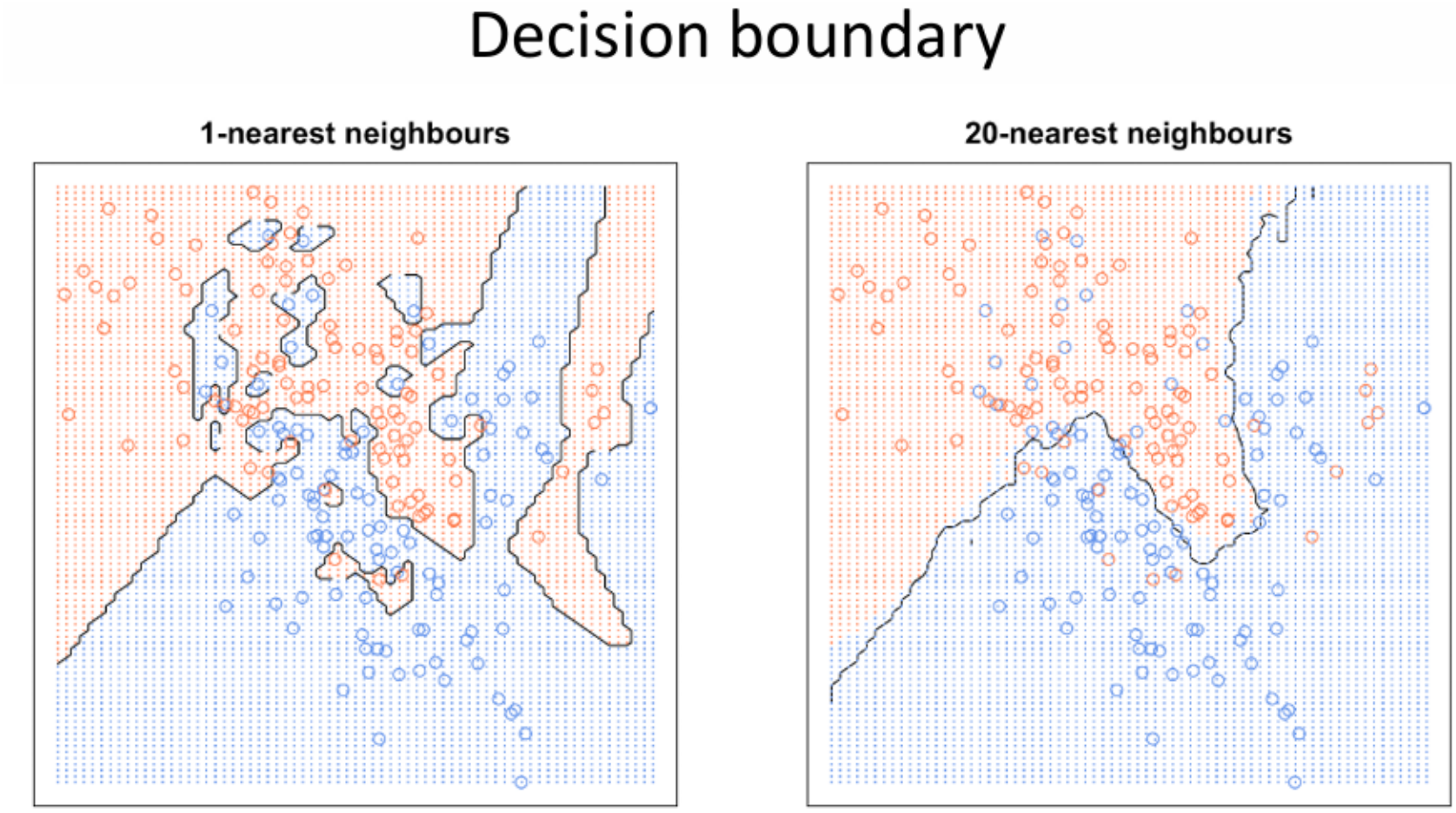
How to choose k?

- Try multiple values: $k = 1, 2, 3, \ldots, N$
- Use odd k values to avoid ties
- Perform cross-validation using `GridSearchCV`

Effect of k:

- k too small: overfitting
- k too large: underfitting

Decision Boundaries



Ref: https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

Issue 1: Feature Scaling

Problem:

- Features with large magnitude dominate distance calculations.
- Example: Population $= 120{,}000{,}000$ VS Age $= 60$

Solution: Feature Scaling

- **z-score normalization**: Maps to range $(-3, 3)$
- **Min-max scaling**: Maps to range $[0, 1]$

Other Issues in k-NN

Issue 2: Noisy Data

- Solution: Feature Selection
- Methods: Random Forest, Lasso, RFE, PCA

Issue 3: Slow Testing Time for Big Data

- Solution: Use KD-Trees

Issue 4: Storage Requirements

- Solution: Compression (e.g., ZIP)

Issue 5: Curse of Dimensionality

- Solution: ?