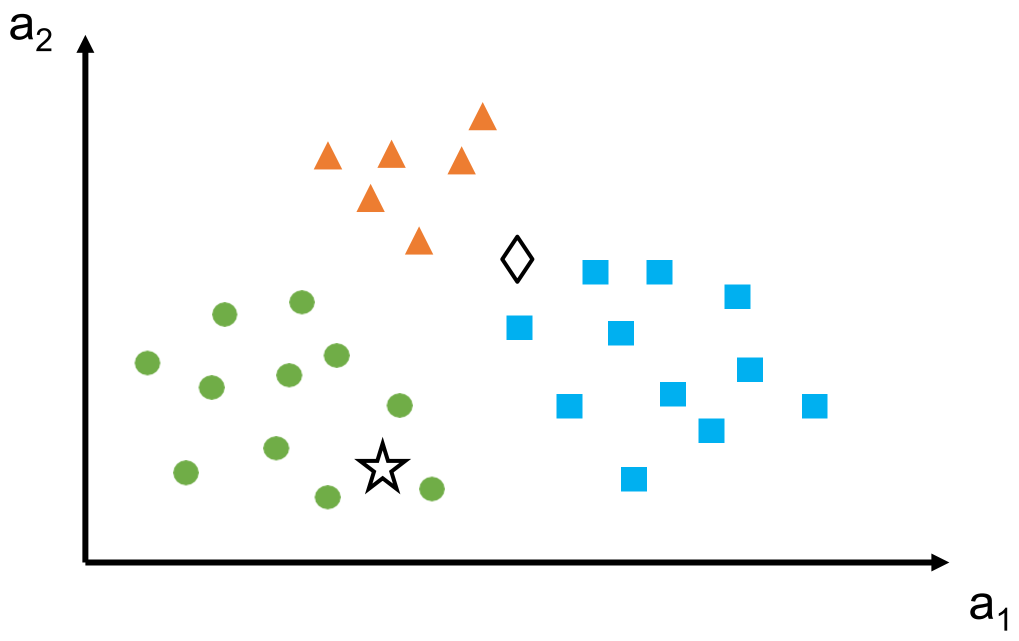
Instance-based Learning: k-Nearest Neighbor Algorithm

COMPCSI 361

Instructor: Thomas Lacombe
Based on slides from Meng-Fen Chiang

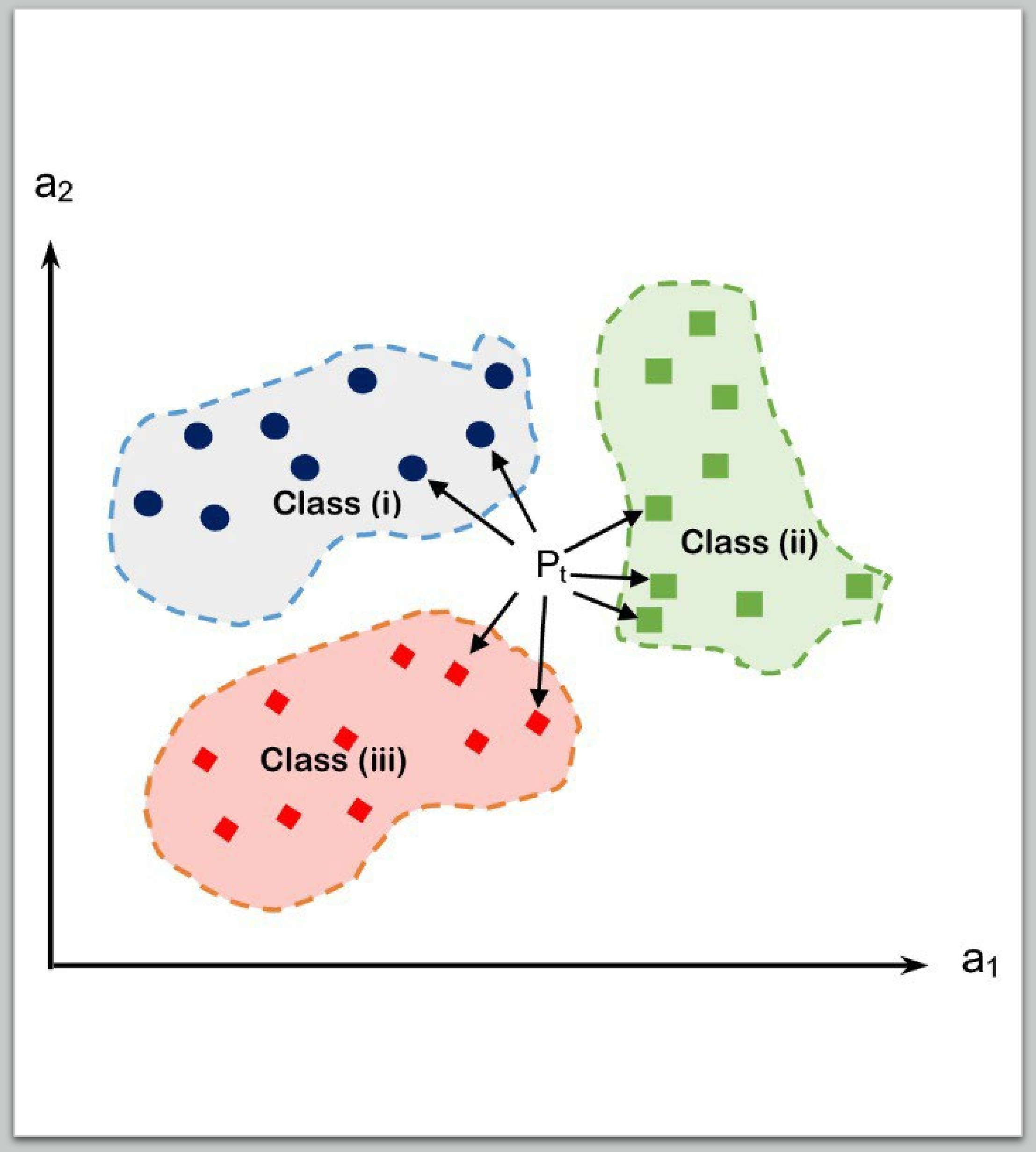
WEEK 9

How would you classify these examples?



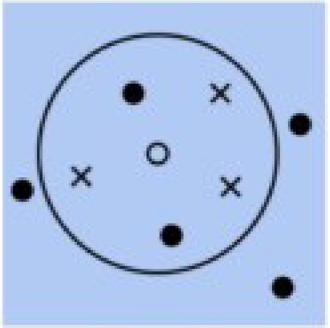
OUTLINE

- k-Nearest Neighbor Algorithm
- Example with Jupyter
- Summary



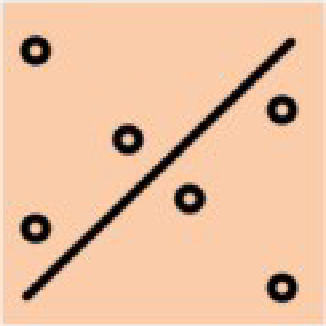
Machine Learning Systems

- Instance-based Learning
  - Compare new data points to known data points
  - Non-parametric approaches
  - Memory-based approaches
  - Prediction can be expensive



use the entire dataset as a model (e.g., k-NN)

- Model-based Learning
  - Detect a pattern in the training data
  - Build a predictive model
  - Prediction is extremely fast



use the training data to create a model that has parameters learned from the training datasets (e.g., SVM)

Instance-Based Learning

- Construct hypotheses directly from the training instances themselves
- The hypothesis complexity can grow with the data
- Example: a hypothesis is a list of n training items and the computational complexity of classifying a single new instance is $O(n)$.

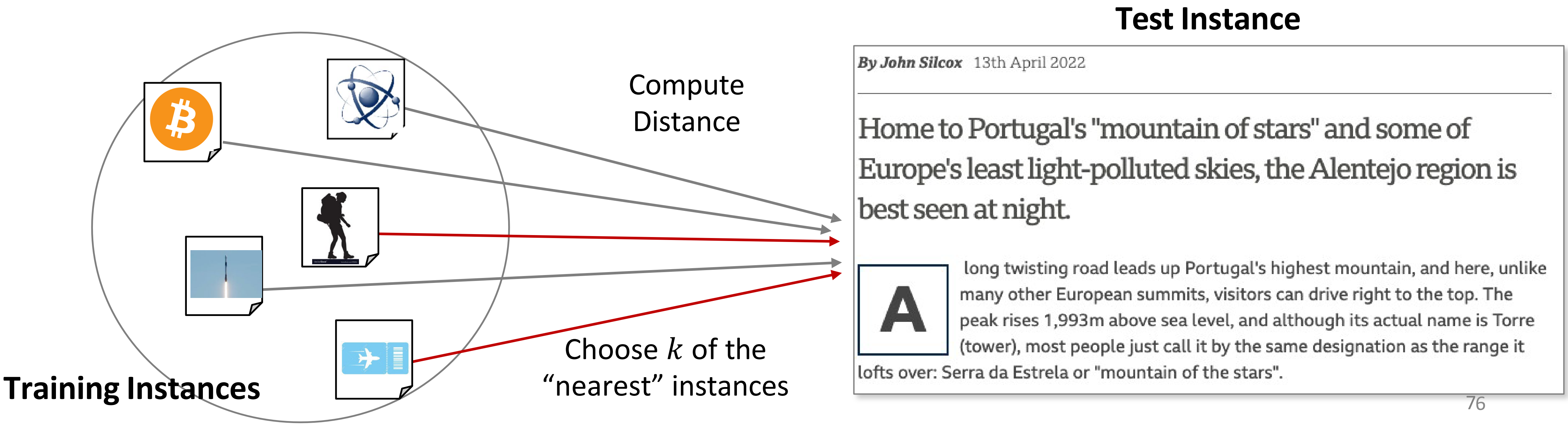
Classification: A Mathematical Formulation

- Given a set of training data $S = ((x_1, y_1), \ldots, (x_n, y_n)), y_i \in \{C_1, \ldots, C_k\}$
- Goal: The classification is to learn a function $f: X \rightarrow Y$ to predict labels for unknown data x'
  - Methods: **k-NN, SVM**, **NN**, Logistic Regression, Probabilistic Classifiers, etc.

- Example: News Article Classification (3-way Classification, i.e., n=3)

Nearest-Neighbor Classifiers

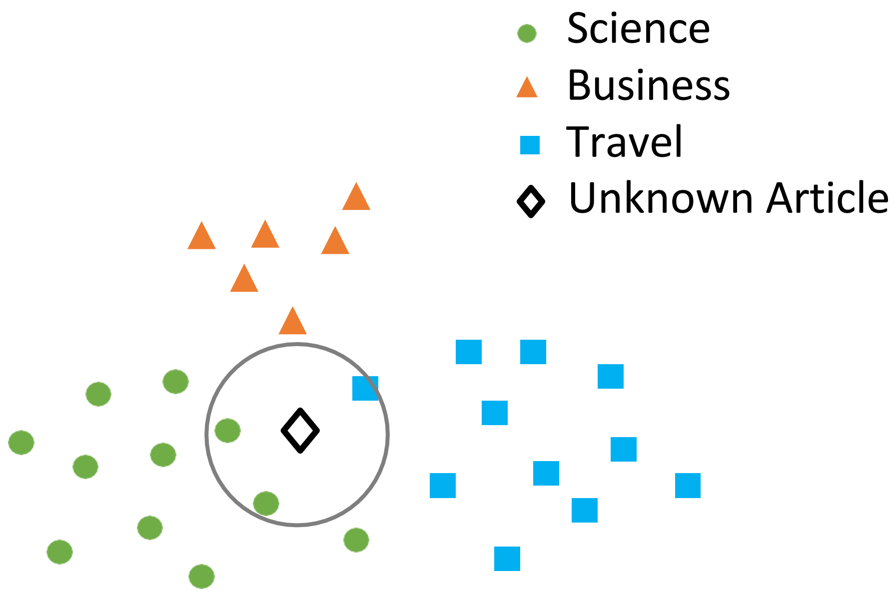
- **Basic idea.** If it walks like a duck, quacks like a duck, then it's probably a duck
- ***k*-Nearest Neighbors (*k*-NN)**. Uses k "closest" points (nearest neighbors) for performing classification



Nearest-Neighbor Classifiers

Requires three things

- The set of stored training instances
- Distance metric to compute distance between instances
- The value of k, the number of nearest neighbors to retrieve

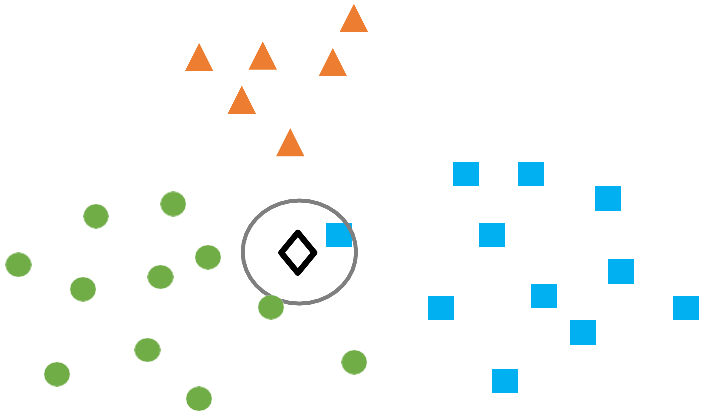


To classify an unknown instance

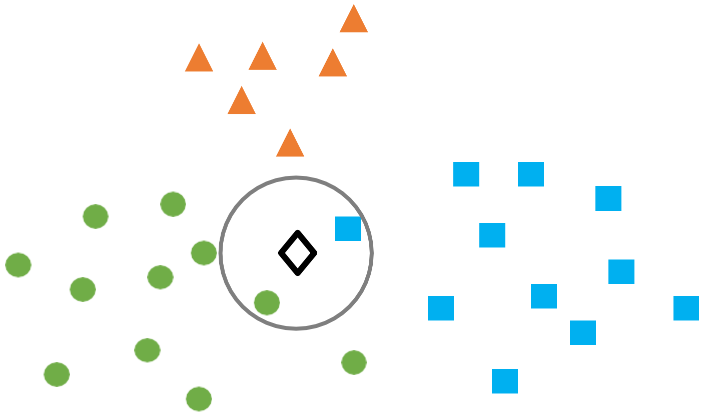
- Compute distance to other training records
- Identify k nearest neighbors
- Use class labels of the nearest neighbors to determine the class label of the unknown instance (e.g., by taking majority vote)

Definition of Nearest Neighbor

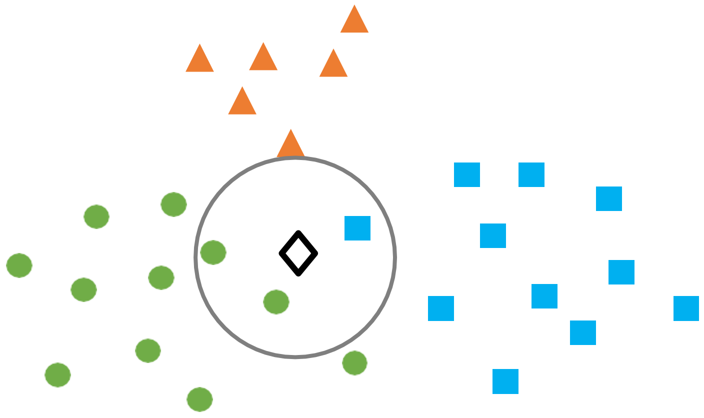
k-nearest neighbors of an instance x' are data points that have the k smallest distance to x'



1-nearest neighbor



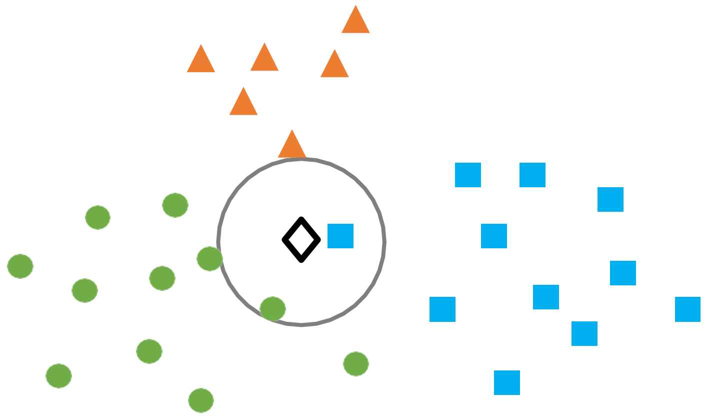
2-nearest neighbor



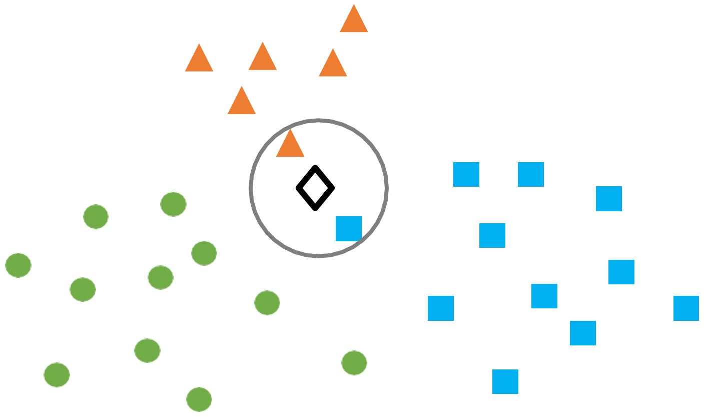
3-nearest neighbor

Breaking ties

What if you encounter one of the following situations?



2-nearest neighbor
No majority amount neighbors



1-nearest neighbor
2 data points at the same distance

Need to break ties:

- Choose an odd k value (does not solve every possible ties).
- Randomly select between tied neighbors.
- Weight the vote by distance.

The k-Nearest Neighbor Algorithm

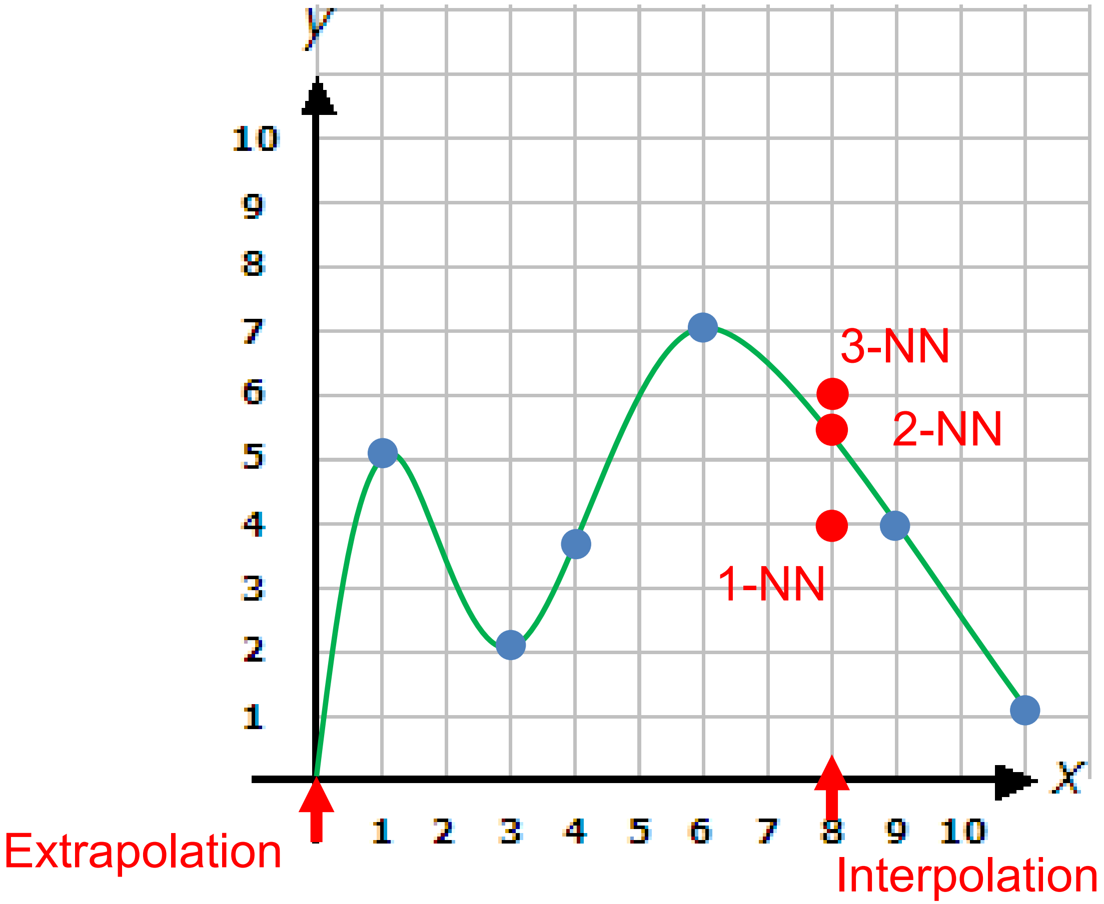
- All instances correspond to points in the D-dim space

- The nearest neighbors are defined based on Euclidean distance:

$$d_E(x, y) = \sqrt{\sum_{1}^{D} (x_i - y_i)^2}$$

- Target function could be discrete- or real- valued
  - Discrete-valued (Classification): k-NN returns the most common value among the k training examples nearest to x'
  - Real-valued (Regression): k-NN returns the mean values among the k training examples nearest to x'

k-Nearest Neighbor for Regression in 1D



Prediction: Average? Majority? Why?

- k-NN for **real-valued** prediction for a given unknown tuple ➔ Regression Problem
  - Returns the mean values of the k nearest neighbors

- **Distance-weighted** nearest neighbor algorithm
  - Weight the contribution of each of the k neighbors according to their distance to the query x'
  - Give greater weight to closer neighbors: $$w = \frac{1}{d(x', x_i)}$$

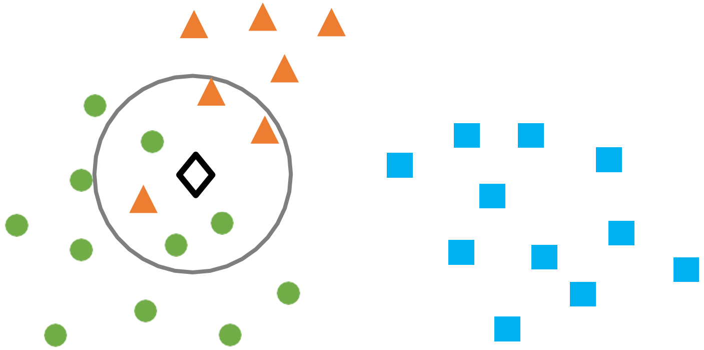
- **Robust** to noisy data by averaging k-nearest neighbors

- **Curse of dimensionality**: distance between neighbors could be dominated by irrelevant attributes
  - To overcome it, axes stretch or elimination of the least relevant attributes

Weighted k-NN and noise

Classification:

→ Higher effect of closest points in the vote



Regression:

→ Higher effect of closest points in the mean

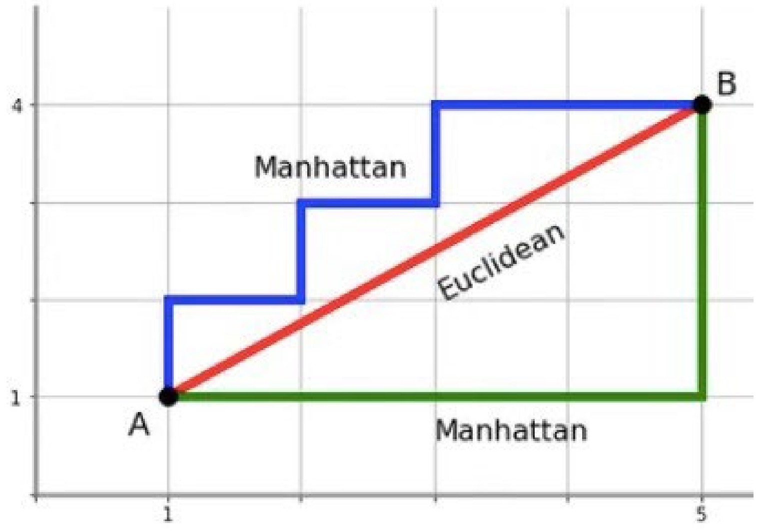
Hyperparameters of k-NN

- **Number of Neighbors (k)**: This is the k value in the k-NN algorithm.
- **Distance Metric**: Distance metric to be used to compute distances between samples.
  - **Euclidean distance**: root of square difference between co-ordinates of pair of points x and y on D-dimensional plane.

$$d_E(x,y) = \sqrt{\sum_{1}^{D}(x_i - y_i)^2}$$

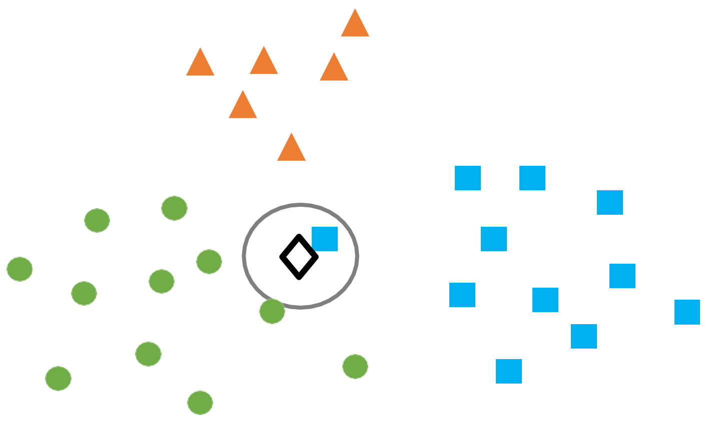
  - **Manhattan distance**: absolute differences between coordinates of pair of points x and y on D-dimensional plane

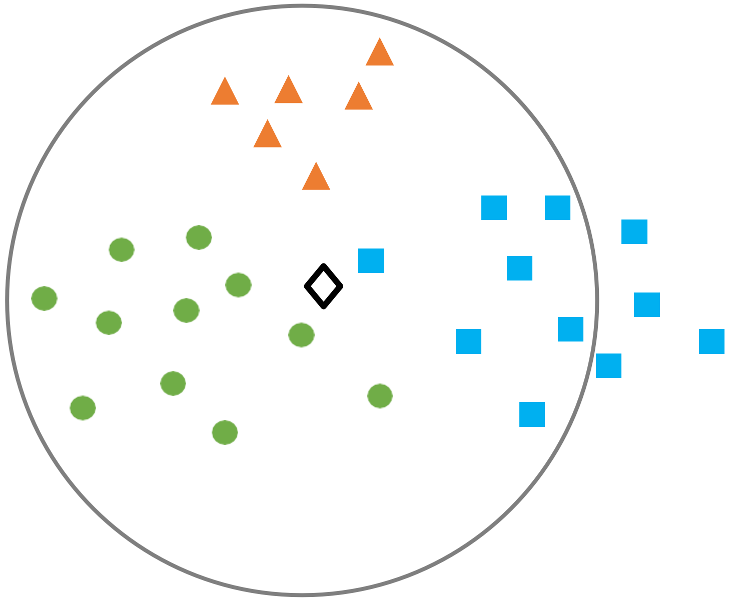
$$d_M(x,y) = \sum_{1}^{D} |(x_i - y_i)|$$



Choice of k: Bias v.s. Variance

- If k is too small, sensitive to noise points
- If k is too large, neighborhood may include points from other classes



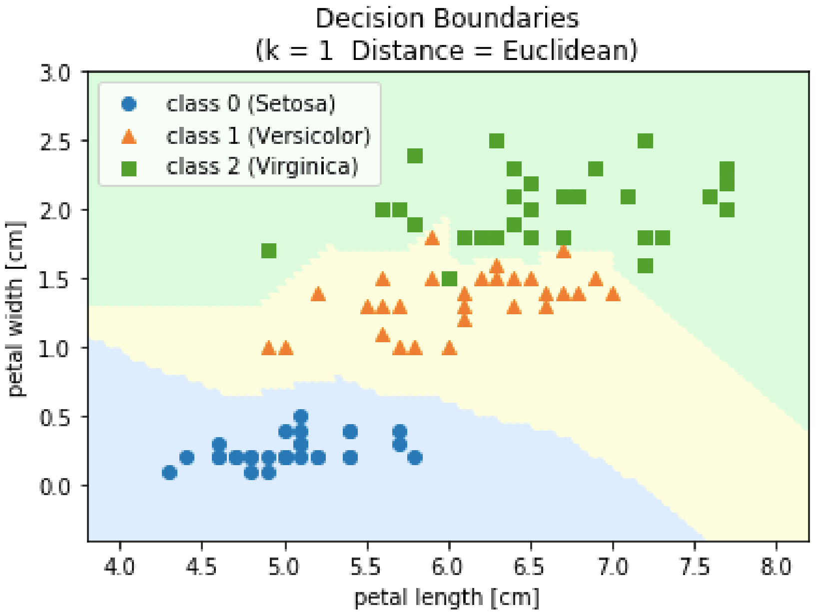


Euclidean Distance: $$d_E(x, y) = \sqrt{\sum_1^D (x_i - y_i)^2}$$

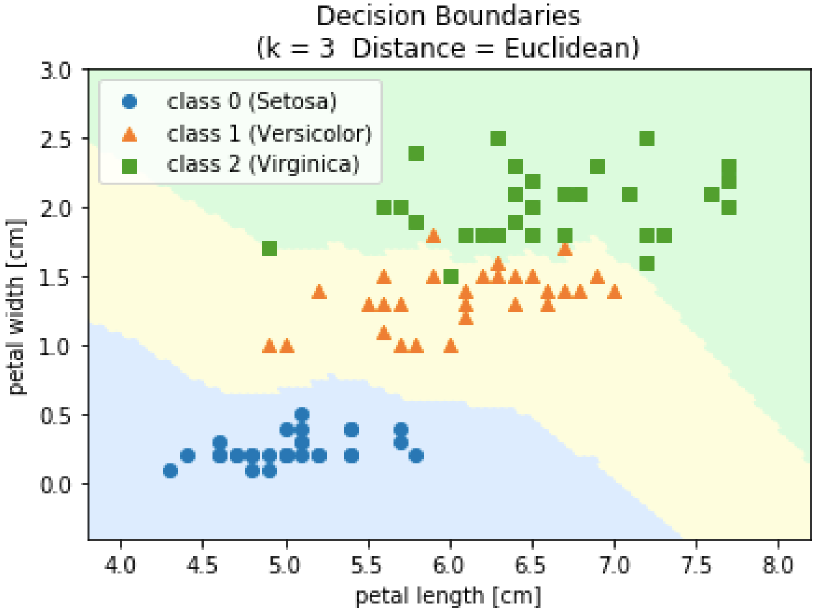
- Boundary becomes jagged
- Sensitive to Noise
- Overfitting

Question: What's the impact of k?

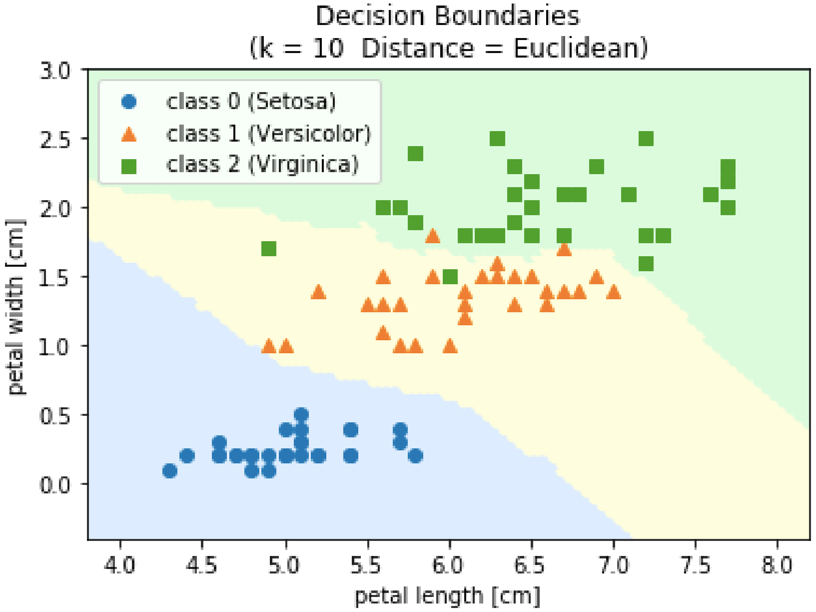
- Boundary becomes smoother
- Affected by irrelevant classes
- Underfitting



(a)



(b)



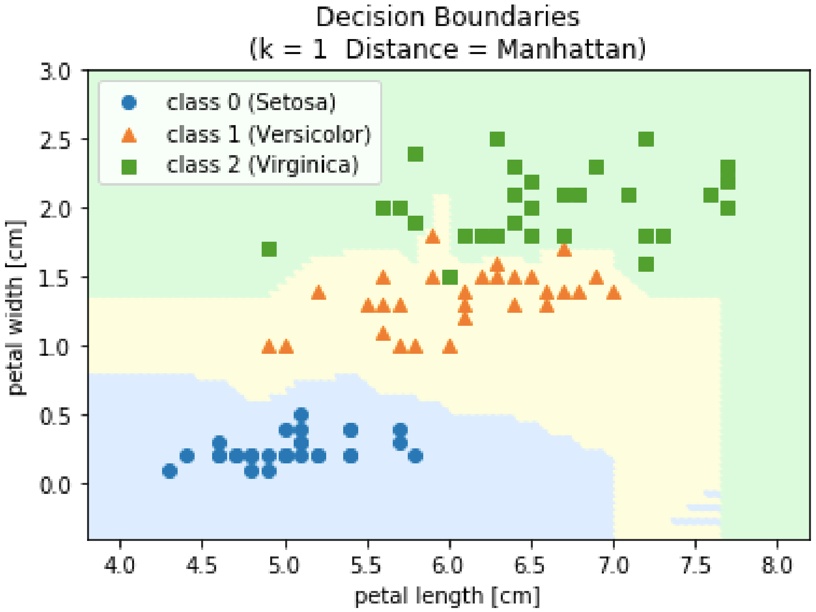
(c)

Manhattan Distance: $d_M(x,y) = \sum_{1}^{D} |(x_i - y_i)|$

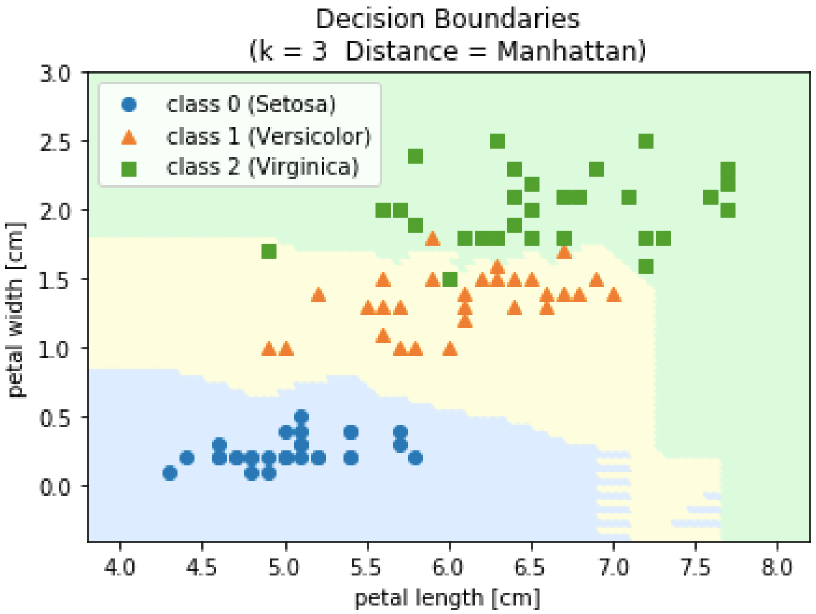
- Boundary becomes jagged
- Sensitive to Noise
- Overfitting

Question: What's the impact of k?

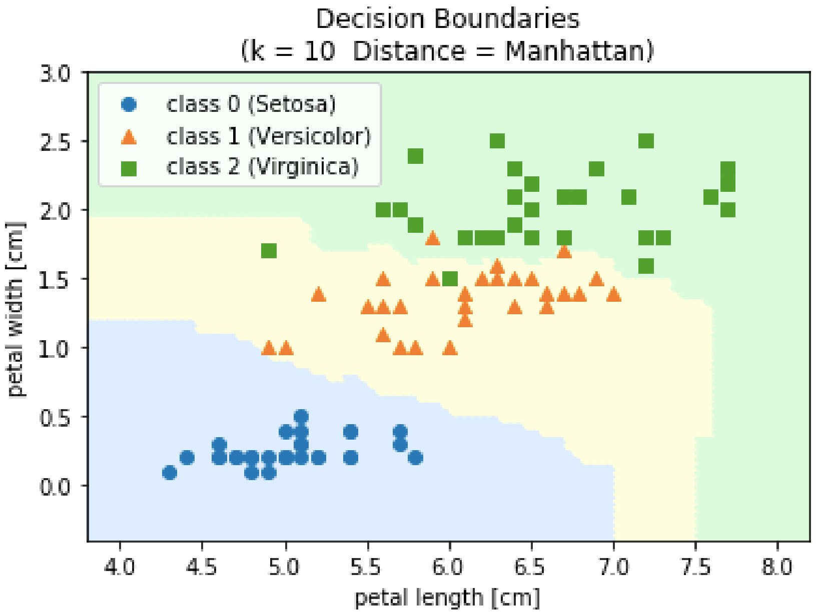
- Boundary becomes smoother
- Affected by irrelevant classes



(a)



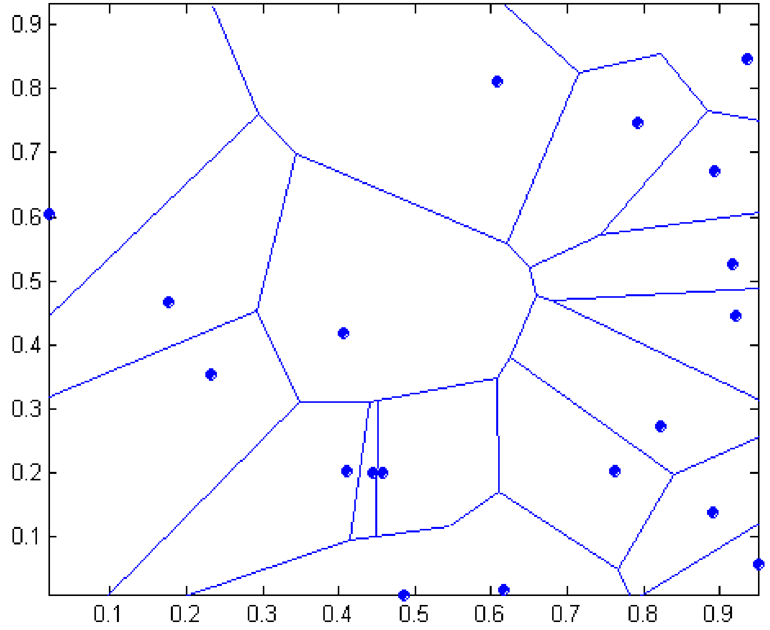
(b)



(c)

Decision Boundaries

- k-NN produces decision boundaries of arbitrary shape
- Provide more flexibility compared to rule-based classifiers
- High variety because decision boundaries depends on training samples in the local neighborhood
- Vonoroi diagram (1-NN): the decision surface induced by 1-NN for a typical set of training examples



k-NN: Other Issues

- **Scaling Issue.** Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes, e.g.,
  - height of a person may vary from 1.5m to 1.8m
  - weight of a person may vary from 90lb to 300lb
  - income of a person may vary from \$10K to \$1M

- **Irrelevant and Redundant Attributes Issue**
  - Irrelevant attributes add noise to the proximity measure
  - Redundant attributes bias the proximity measure towards certain attributes

Advantages v.s. Disadvantages

Advantages

- Easy to implement
- Incremental addition of training data trivial
- k -NN classifiers are local classifiers
- k -NN classifiers can produce decision boundaries of arbitrary shapes

Disadvantages

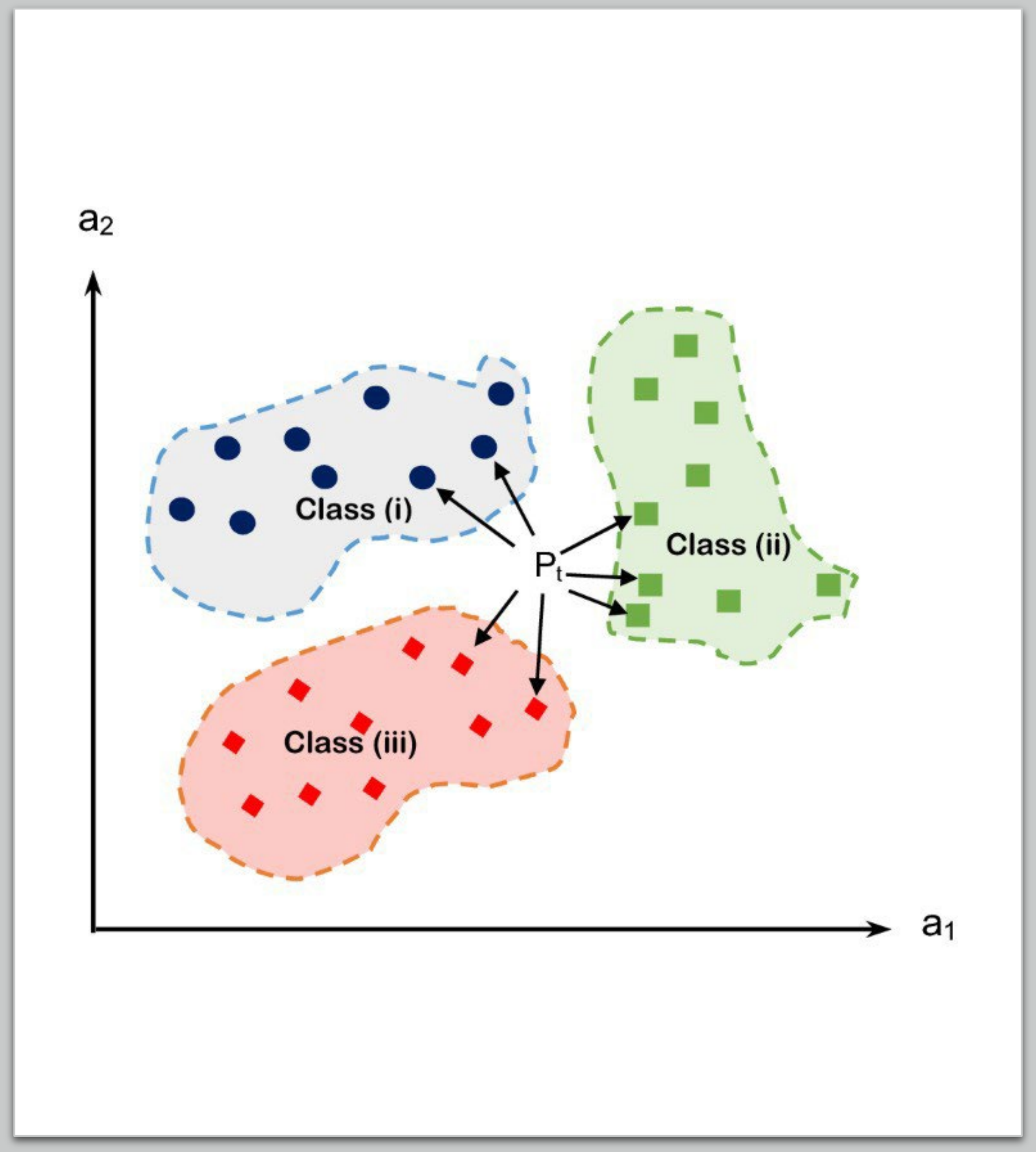
- k -NN classifiers are lazy learners, which do not build models explicitly. This can be relatively more **expensive** than eager learners (such as decision tree) when classifying a test/unknown instance
- Unlike decision tree that attempts to find a global model that fits the entire input space, nearest neighbor classifiers make the prediction based on local information, which can be more **susceptible to noise**

Jupyter Notebook

k-NN Coding Example

SUMMARY

- k-Nearest Neighbor Algorithm
  - Definition of Nearest Neighbor
  - Classification vs Regression
  - Bias v.s. Variance Tradeoff : Impact of k



Resources

- Coding Library
  - **Scikit-Learn**: a set of supervised neighbors-based learning comes in two flavors: classification for data with discrete labels, and regression for data with continuous labels. [link]
  - Notebook Examples: Python Data Science Handbook by Jake VanderPlas (https://github.com/jakevdp/PythonDataScienceHandbook)

- Book Chapters
  - Introduction to Data Mining [Book] by Tan, Steinbach, and Kumar. Chapter 6.3