K-Nearest Neighbors

Guowei Wei
Department of Mathematics
Michigan State University

References:
Duc D. Nguyen's lecture notes
Wikipedia

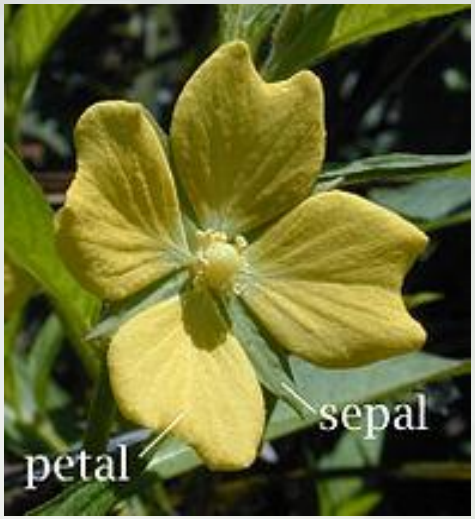
Introduction

- *K*-Nearest Neighbors (*k*-NNs) have been used for statistical estimation and pattern recognition since the 1970s
- *K*-NN is a non-parametric technique (non-assumption on data distribution)
- It is still one of the top 10 data mining algorithms.
- It can be used for both classification and regression

An example problem

- Classification
- We consider the **iris** dataset
  - Include three types of iris plant:
    - iris setosa,
    - iris versicolour
    - iris virginica
  - 4 features:
    - sepal length in cm
    - sepal width in cm
    - petal length in cm
    - petal width in cm
  - 150 samples (50 in each of three classes)

Iris setosa

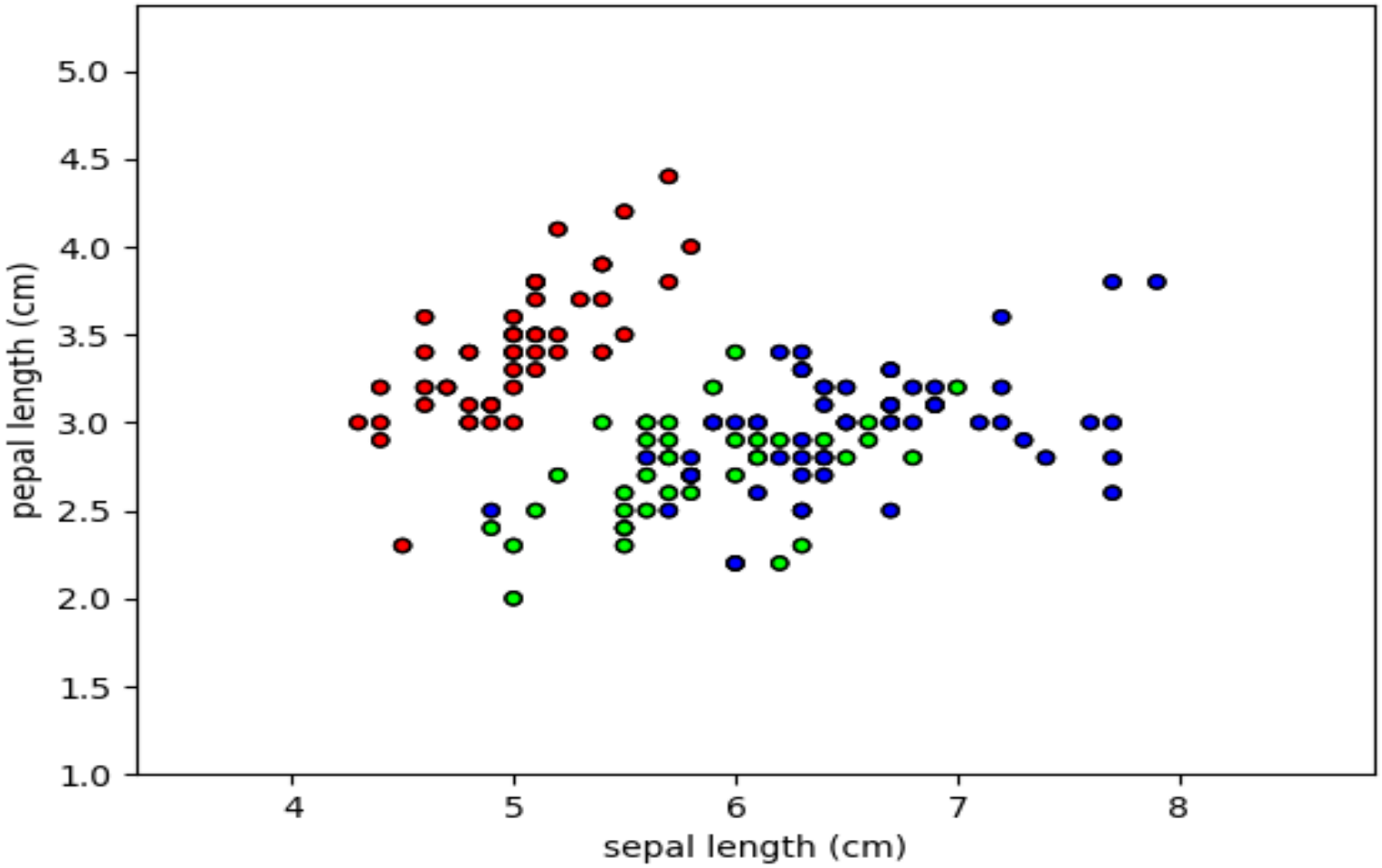


Iris versicolour

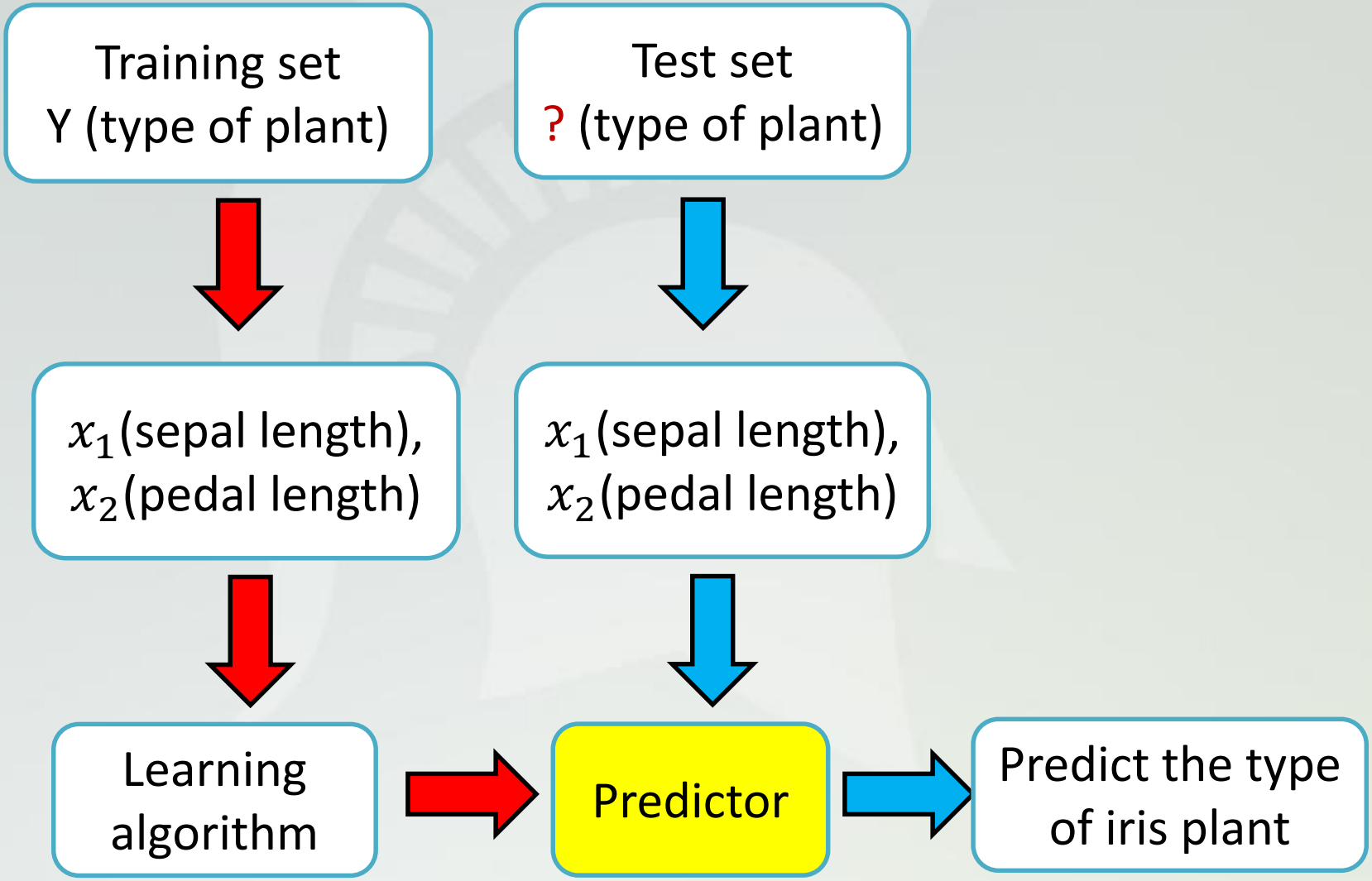
Iris virginica

Example

Two-parameter classification



Model Representation



Predictor Construction

- Construct a predictor:

$$p_{\mathbf{c}}(\mathbf{x}) = ?$$

- No explicit formulation for $p_{\mathbf{c}}(\mathbf{x})$ and no parameters $\mathbf{c}$

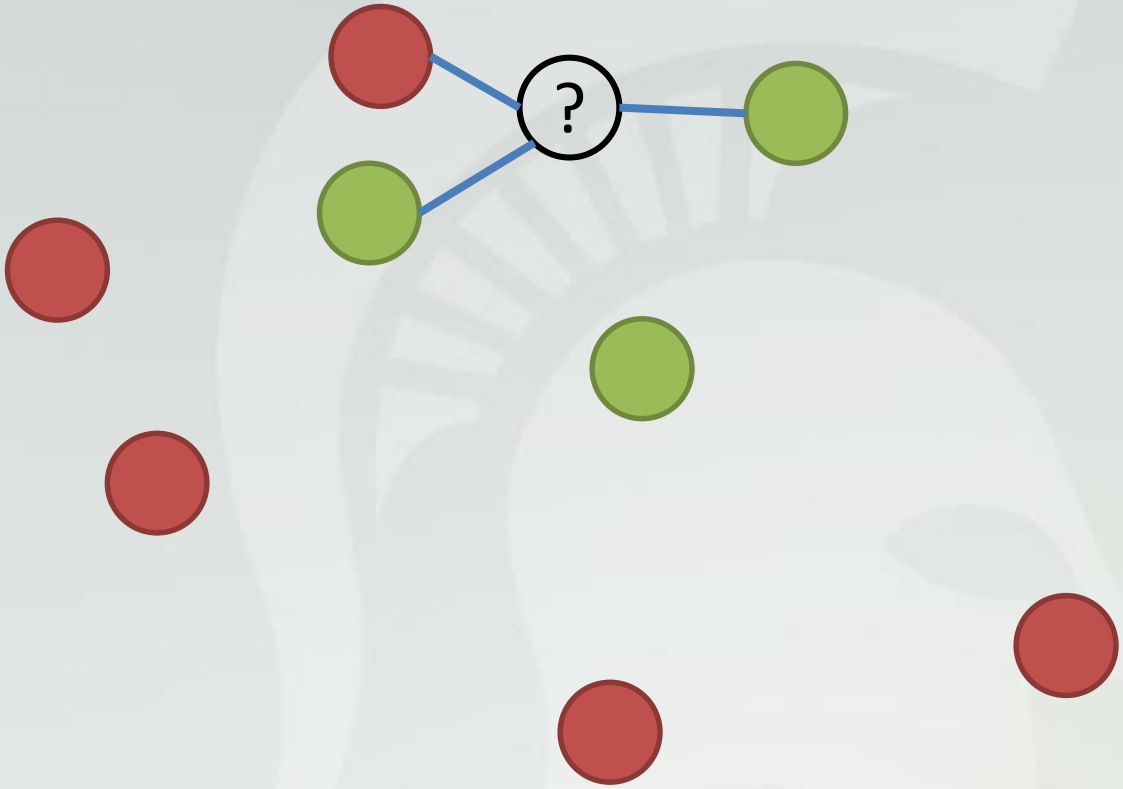
***k-NN* algorithm:**

- k is a given positive number
- $\mathbf{x}$ is the feature vector of new sample associated with unknown label y
- find k entries in our dataset that are closest to the new sample $\mathbf{x}$
- label of $\mathbf{x}$ decided by those k entries

K-NN predictions

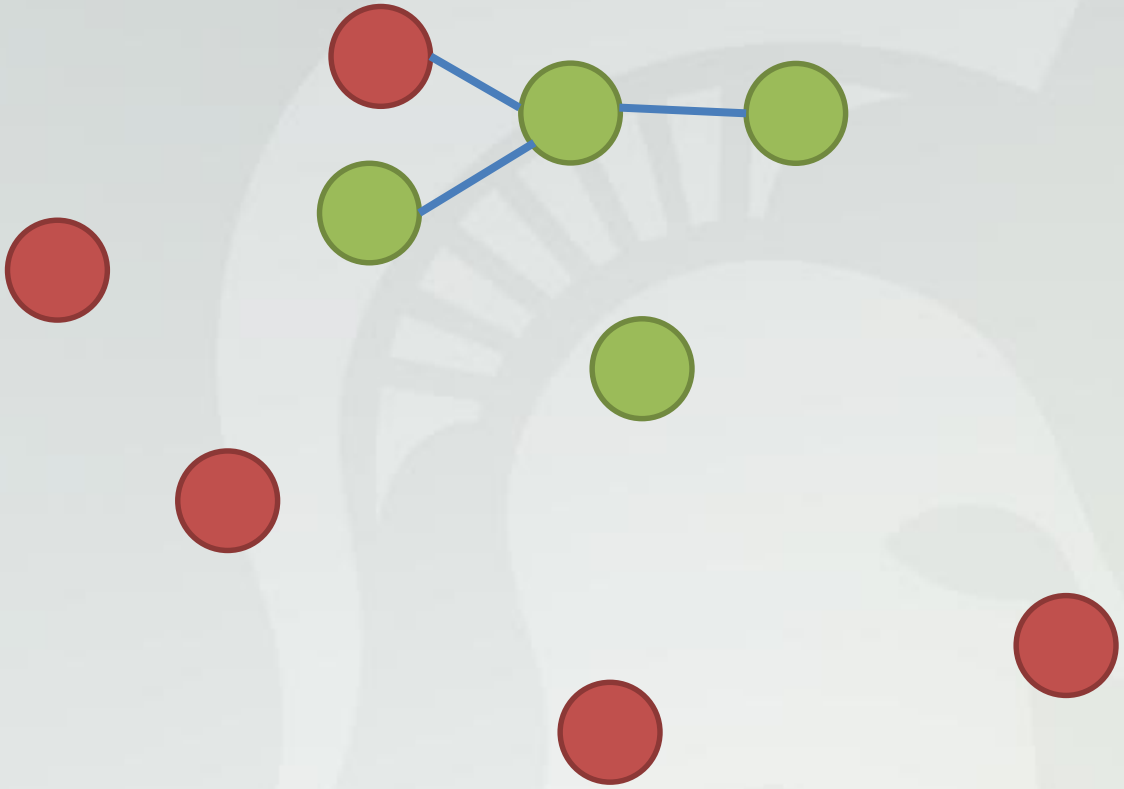
- In classification, the *k*-NN prediction is based on the majority rule of the *k* nearest neighbors
- In regression, the *k*-NN prediction is the average of the *k* nearest neighbor labels (values)

Intuitive Algorithm Illustration



3-NN

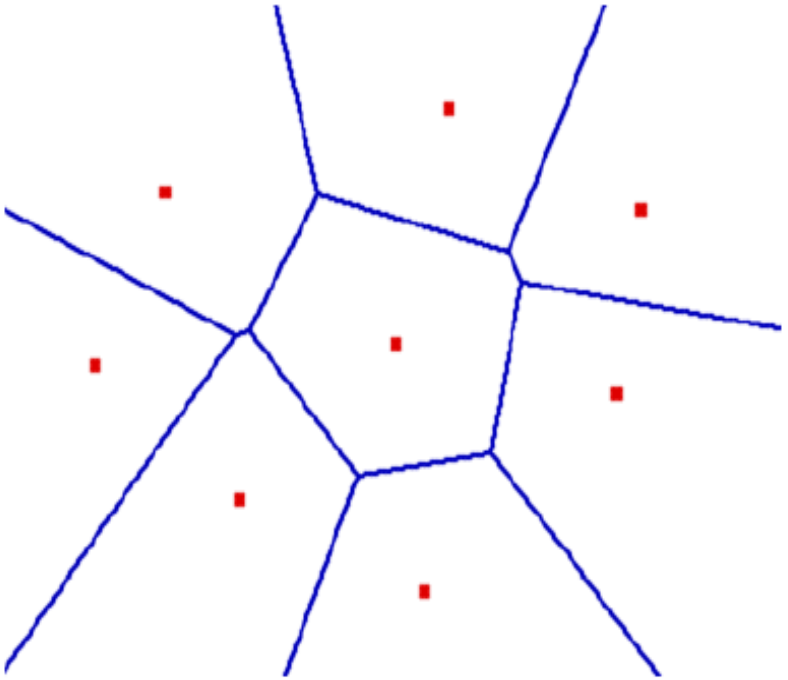
Intuitive Algorithm Illustration



3-NN

Decision boundary

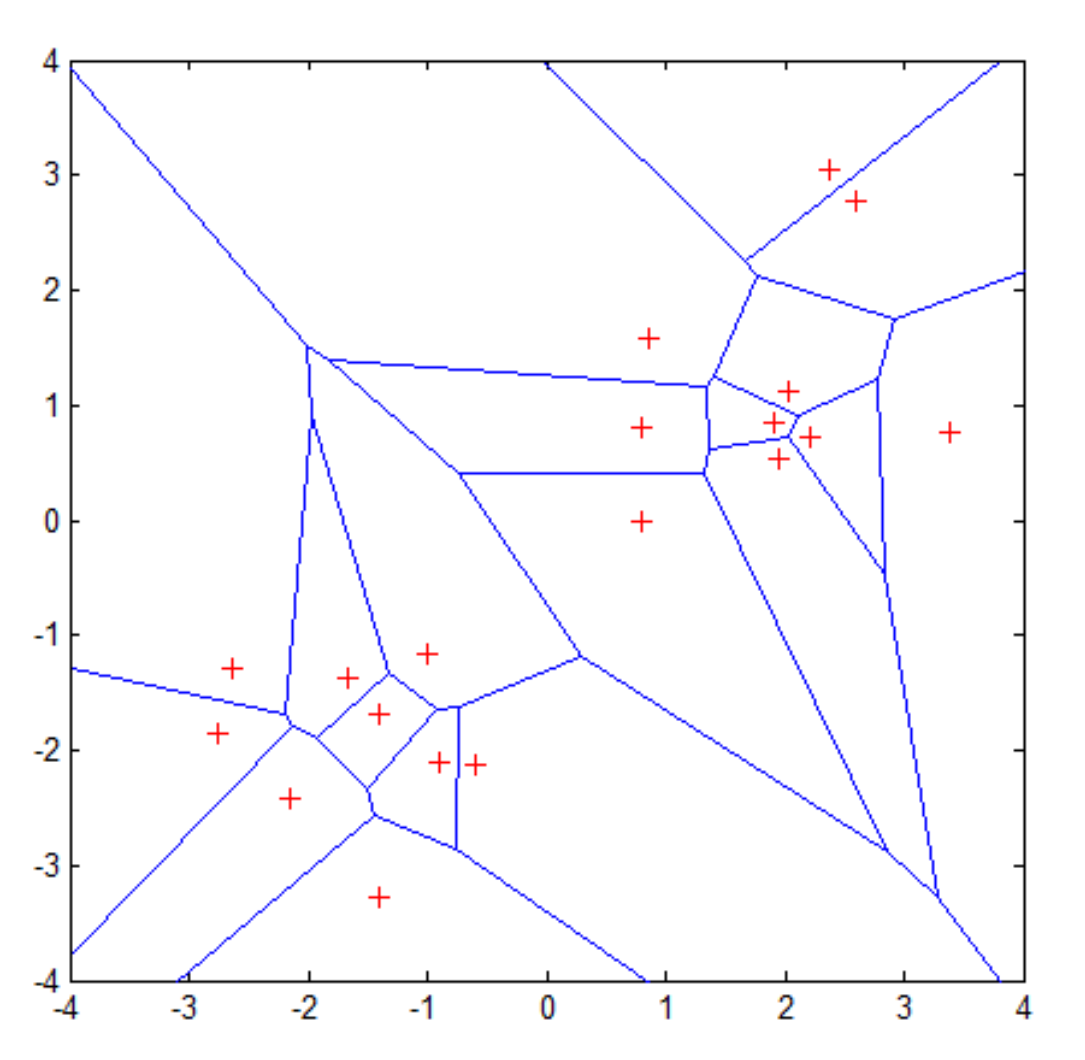
- Given a set of points, a **Voronoi diagram** describes the areas that are nearest to any given point.
- These areas can be viewed as zones of control.



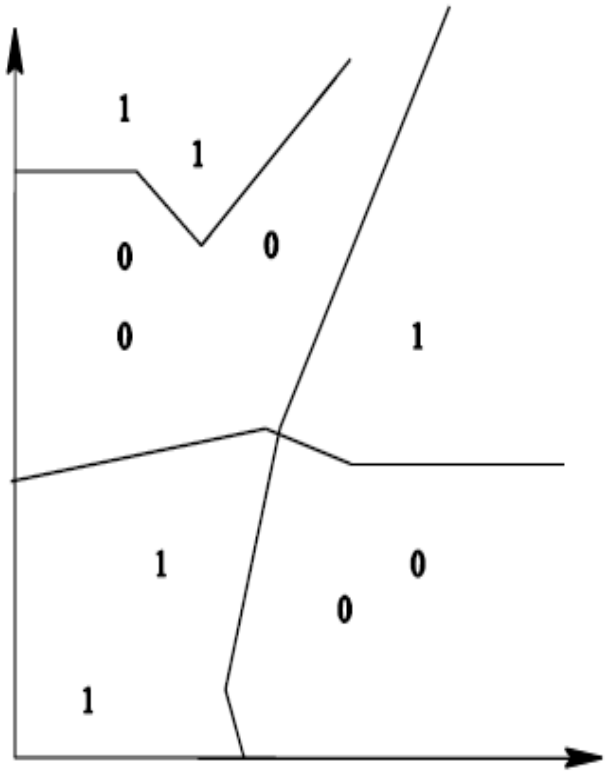
Graphic Depiction

Properties:

1) All possible points within a sample's Voronoi cell are the nearest neighboring points for that sample
2) For any sample, the nearest sample is determined by the closest Voronoi cell edge



- Decision boundaries are formed by a **subset** of the Voronoi diagram of the training data
- Each line segment is equidistant between two points **of opposite class**.
- The more examples that are stored, the more fragmented and complex the decision boundaries can become.



How To Define Closest Entries

- Distance metrics
  - Euclidean distance (L_2)

$$d(\mathbf{x}, \mathbf{z}) = \left(\sum_{i=1}^{N} |x_i - z_i|^2 \right)^{1/2}$$

  - Manhattan distance (L_1)

$$d(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{N} |x_i - z_i|$$

  - Minkowski distance (L_p)

$$d(\mathbf{x}, \mathbf{z}) = \left(\sum_{i=1}^{N} |x_i - z_i|^p \right)^{1/p}$$

How To Define Closest Entries

- Distance metric
  - Chebyshev distance

$$d(\mathbf{x}, \mathbf{z}) = \max_{\boldsymbol{i}} |x_i - z_i|$$

  - Natural log distance

$$d(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{n} \ln(1 + |x_i - z_i|)$$

  - Generalized exponential distance

$$d(\mathbf{x}, \mathbf{z}) = e^{-\left(\frac{\|\mathbf{x}-\mathbf{z}\|}{\eta}\right)^{\kappa}}$$

  - Generalized Lorentzian distance

$$d(\mathbf{x}, \mathbf{z}) = \frac{1}{1+\left(\frac{\|\mathbf{x}-\mathbf{z}\|}{\eta}\right)^{\kappa}} \qquad (\kappa = 1,2, \ldots)$$

- Camberra:

$$d(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{N} \frac{|x_i - z_i|}{|x_i + z_i|}$$

- Quadratic: (with a problem specific $\boldsymbol{Q}$ matrix)

$$d^2(\mathbf{x}, \mathbf{z}) = (\mathbf{x} - \mathbf{z})^T \boldsymbol{Q} (\mathbf{x} - \mathbf{z})$$

$$= \sum_{j=1}^{N} \left(\sum_{i=1}^{N} (x_i - z_i) q_{ji} \right) (x_j - z_j)$$

- Mahalanobis:

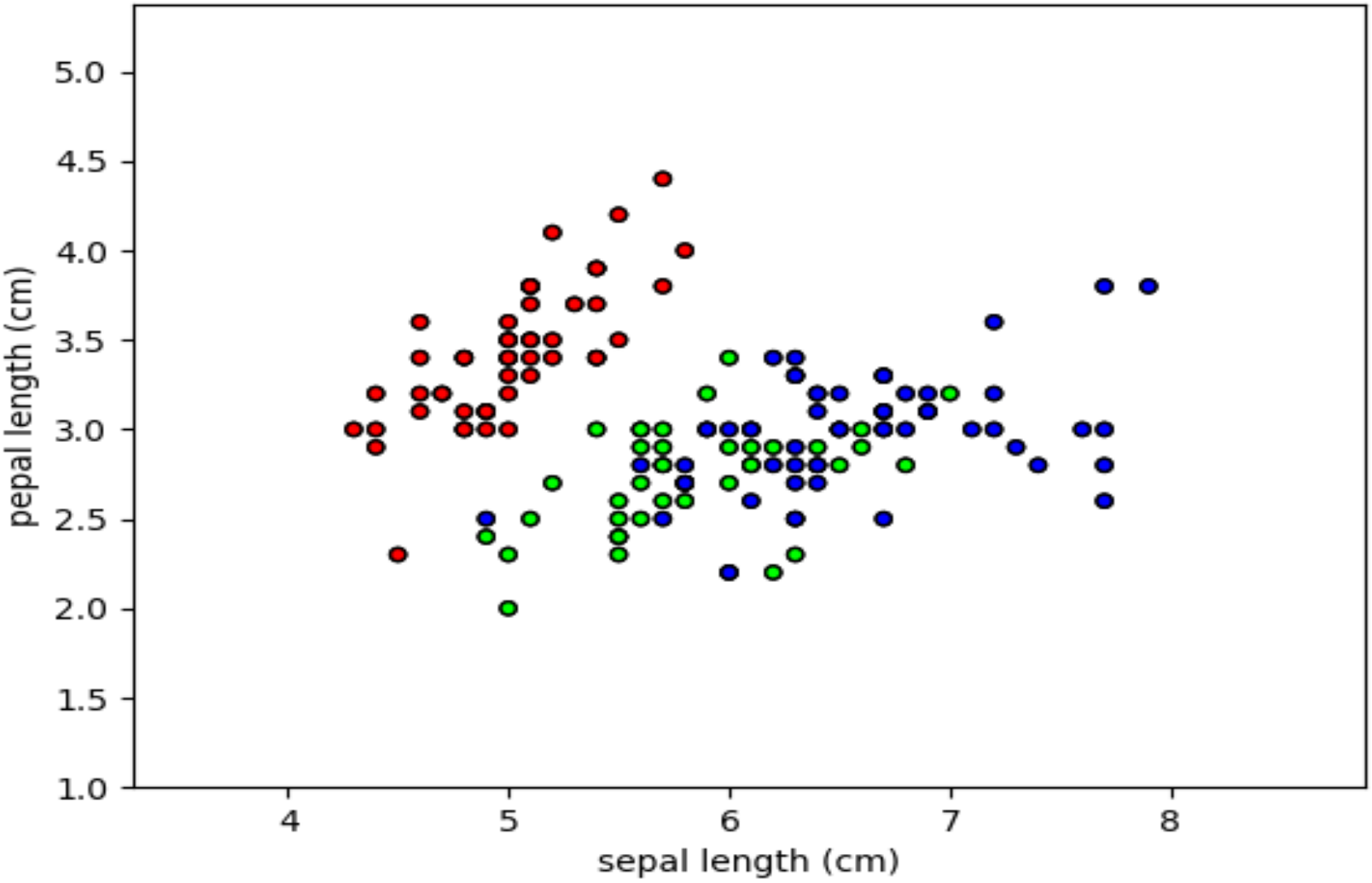
$$d^2(\mathbf{x}, \mathbf{z}) = [\det \boldsymbol{V}]^{1/N} (\mathbf{x} - \mathbf{z})^T \boldsymbol{V}^{-1} (\mathbf{x} - \mathbf{z})$$

$\boldsymbol{V}$ is the covariance matrix of $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_N$, and $\boldsymbol{A}_j$ is the vector of values for attribute j occurring in the training set instances $1, \ldots, m$

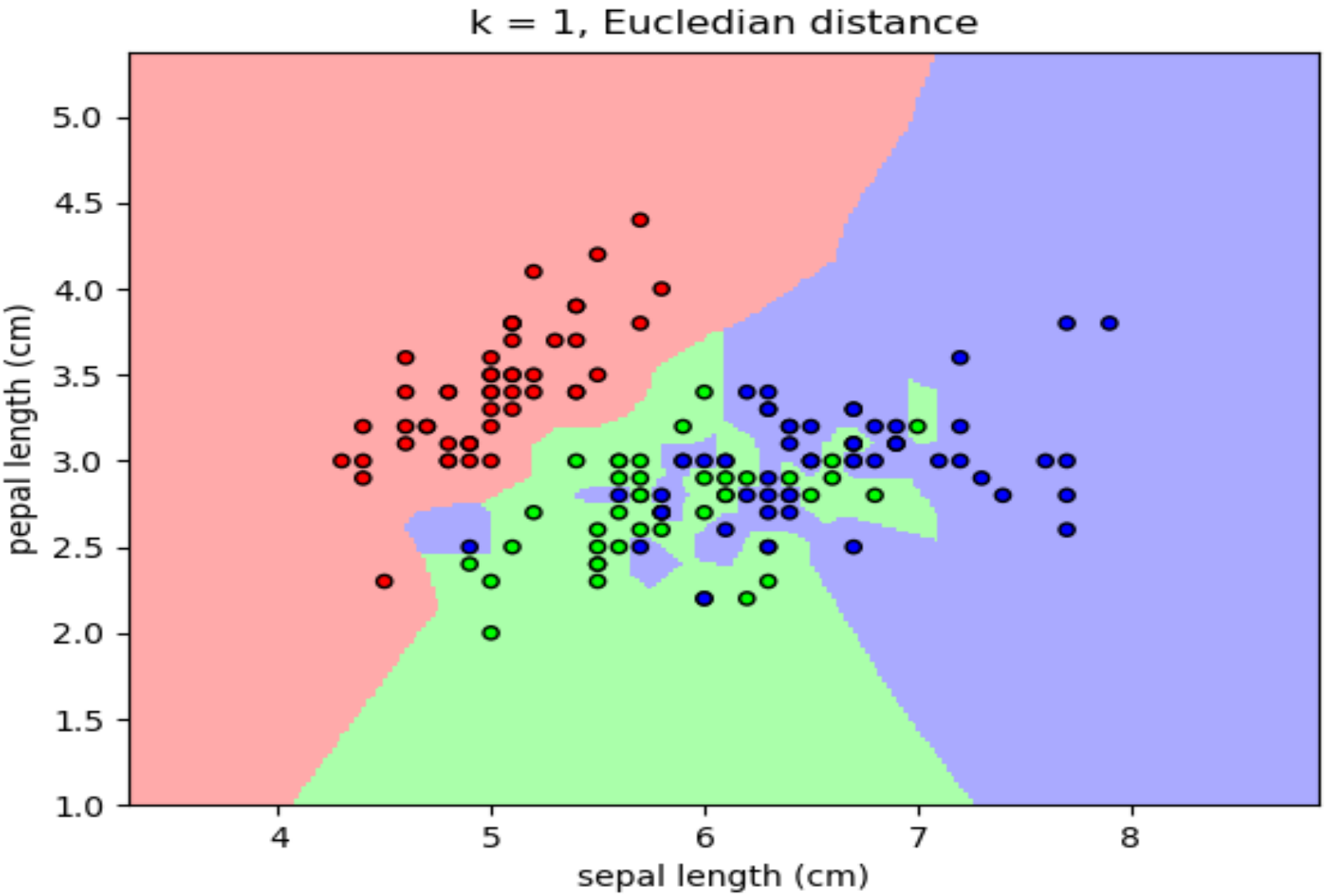
Issues with Distance Metrics

- Most distance measures were designed for linear/real-valued attributes
- Two important questions in the context of machine learning:
  - How to best handle nominal attributes
  - What to do when attribute types are mixed (which ones carry heavier weights)

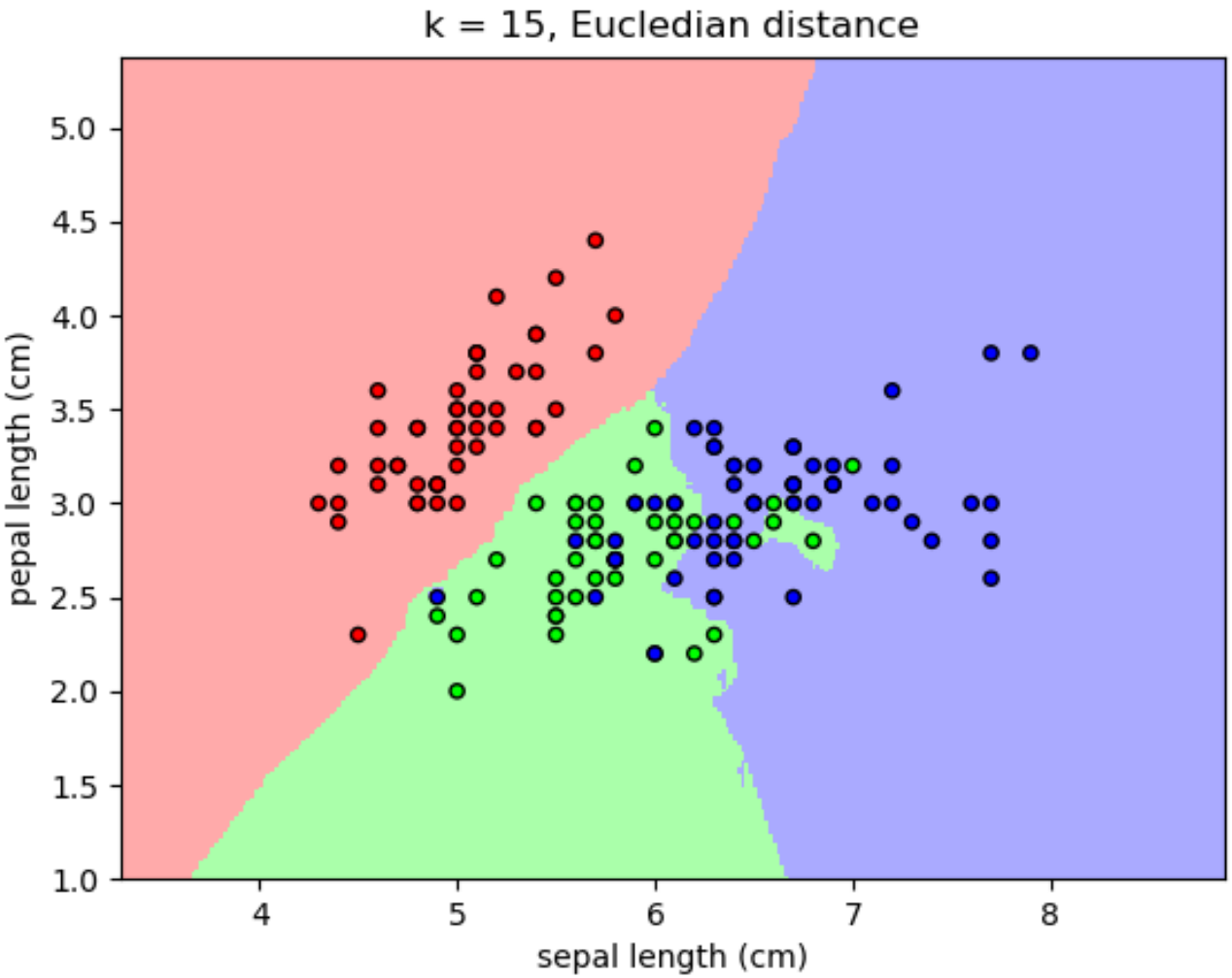
Use k –NN for Iris Dataset



Use *k*–NN for Iris Dataset



Use k −NN for Iris Dataset



Be Careful with Multi-magnitude Features

Consider the following dataset

Name	x_1	x_2	label
P1	1	100	Red
P2	1	120	Red
P3	4	200	Green
P4	4	250	Green

Name	x_1	x_2	label
P5	1	220	?

$$d(\mathrm{P5},\mathrm{P1}) = \sqrt{(1-1)^2 + (220-100)^2} = 120, d(\mathrm{P5},\mathrm{P2}) = 100$$

$$d(\mathrm{P5},\mathrm{P3}) \approx 20.22, \qquad d(\mathrm{P5},\mathrm{P4}) \approx 30.0$$

If we use 2-NN then label of P5 will be **Green**!? **(misclassified)**

Feature Normalization

- Purpose: all features will have relatively similar magnitude
- How?:
  1. Linearly scale features to range $[0,1]$

$$x_{\text{new}} = \frac{x_{\text{old}} - x_{\text{old}}^{\min}}{x_{\text{old}}^{\max} - x_{\text{old}}^{\min}}$$

  2. Linearly scale features to 0 mean and variance 1 (normal distribution)

$$x_{\text{new}} = \frac{x_{\text{old}} - \mu}{\sigma}$$

$$\mu: \text{mean}, \sigma^2: \text{variance}$$

Feature Normalization

Previous dataset

Name	x_1	x_2	Label
P1	1	100	Red
P2	1	120	Red
P3	4	200	Green
P4	4	250	Green

After normalizing features using normal distribution $(\mu(x_1) = 2.5, \sigma(x_1) = 1.5, \mu(x_2) = 167.5, \sigma(x_2) \approx 60.57)$

Name	x_1	x_2	label
P1	-1	-1.11	Red
P2	-1	-0.78	Red
P3	1	0.54	Green
P4	1	1.36	Green

Feature Normalization

Test set (before)

Name	x_1	x_2	label
P5	1	220	?

Test set (after normalization)

Name	x_1	x_2	label
P5	-1	0.87	?

Feature Normalization

Training set (normalized)

Name	x_1	x_2	label
P1	-1	-1.11	Red
P2	-1	-0.78	Red
P3	1	0.54	Green
P4	1	1.36	Green

Test set (normalized)

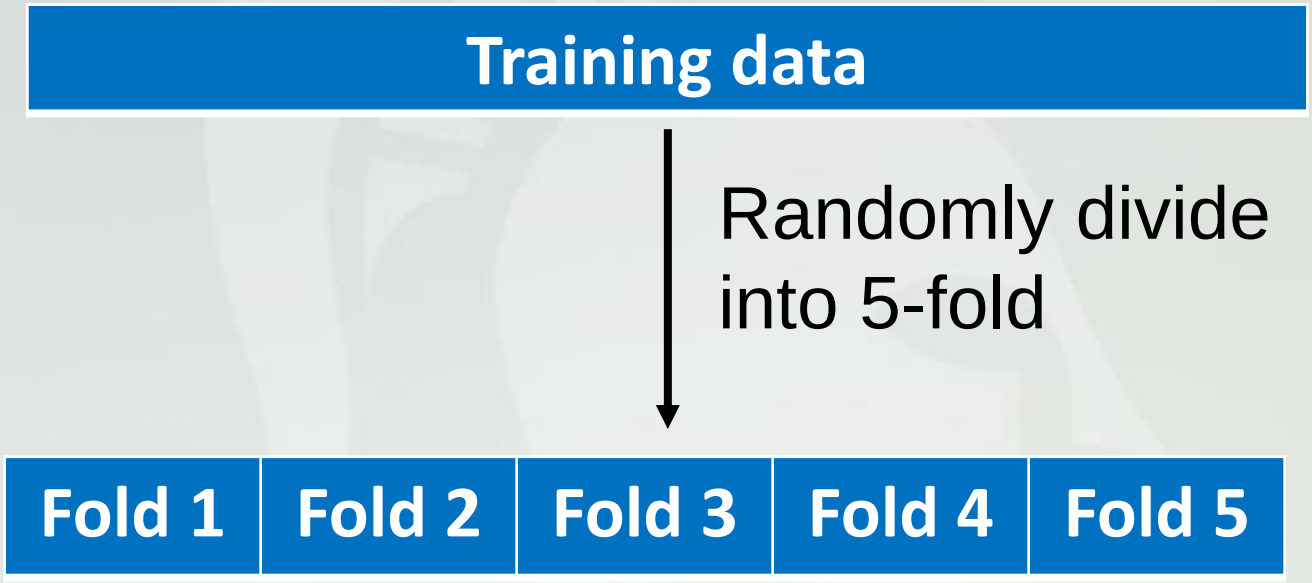
Name	x_1	x_2	label
P5	-1	0.87	?

$$d(\mathrm{P5}, \mathrm{P1}) = 1.98, \quad d(\mathrm{P5}, \mathrm{P2}) = 1.65,$$
$$d(\mathrm{P5}, \mathrm{P3}) = 2.03, \quad d(P5, P4) = 2.06$$

If we use 2-NN then label of P5 will be red.

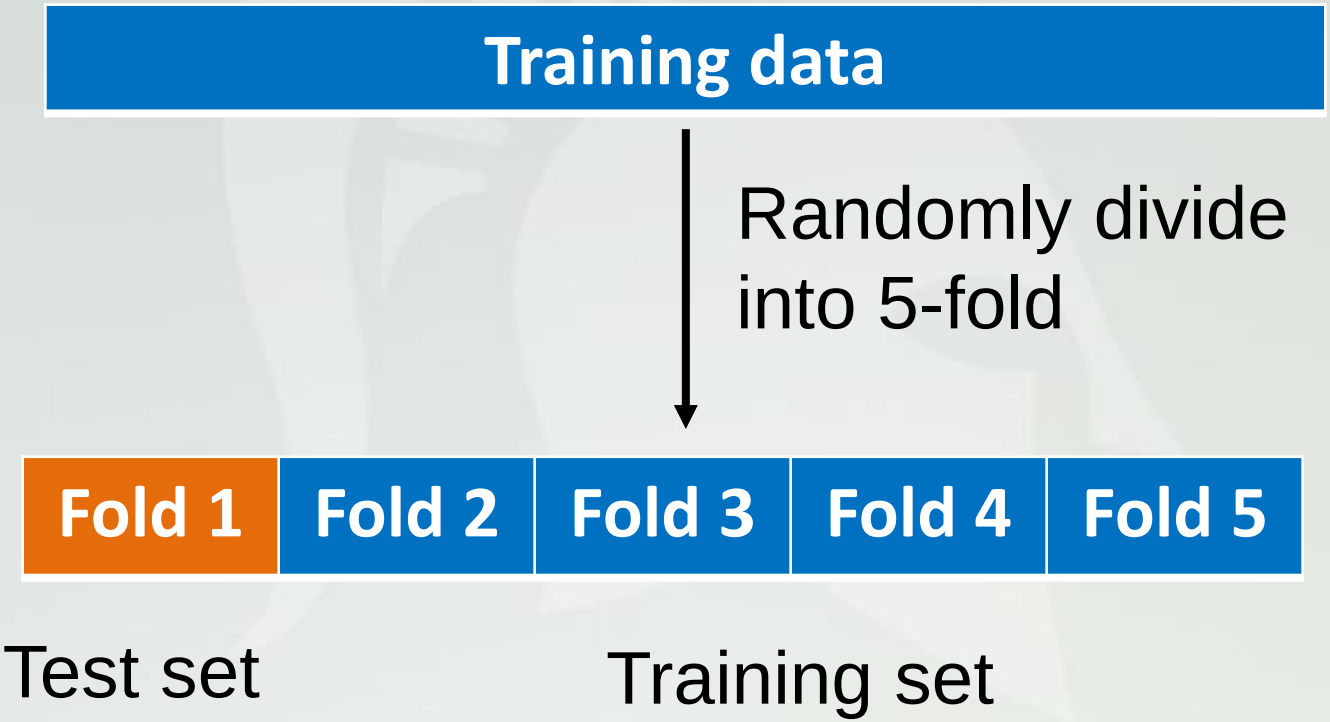
How to Choose k?

- Do cross-validation



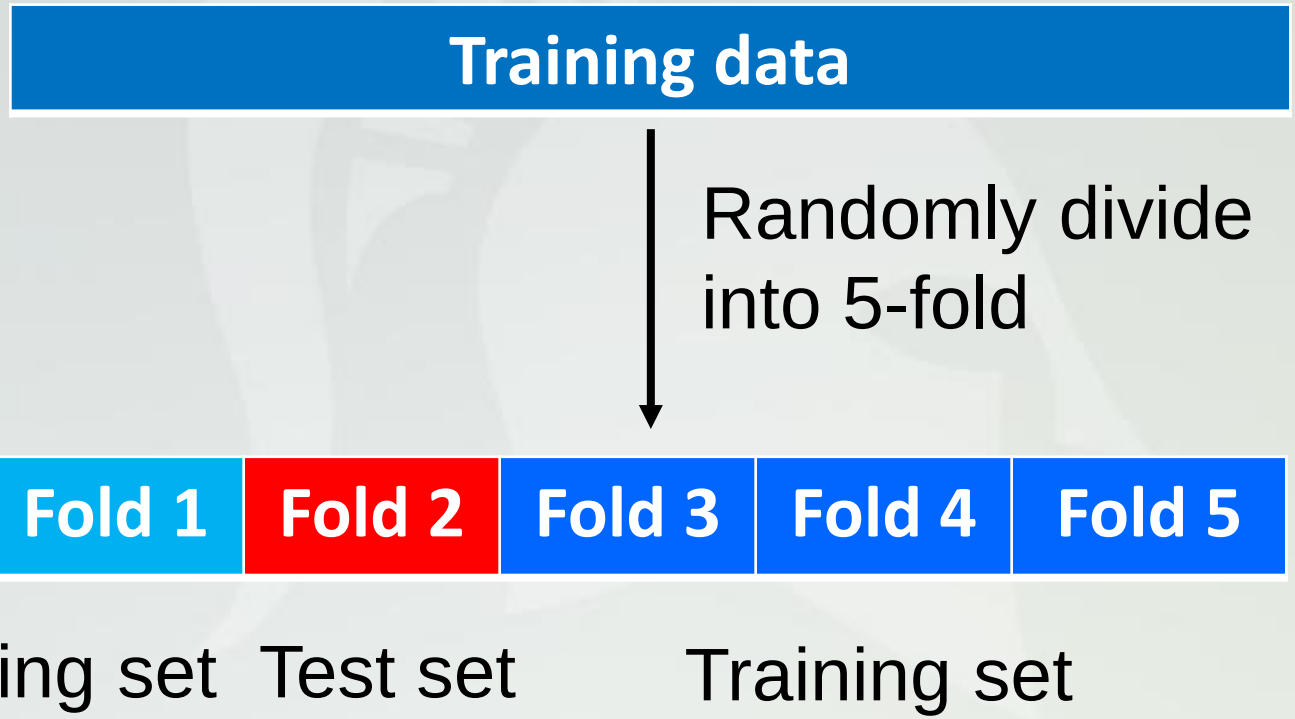
How to Choose k?

- Do cross-validation



How to Choose k?

- Do cross-validation

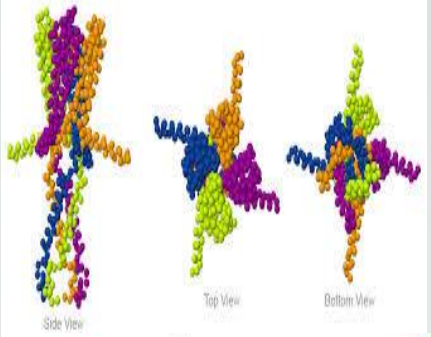


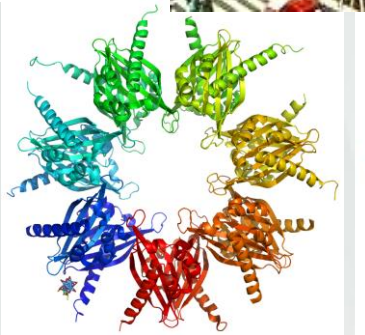
Pros and Cons

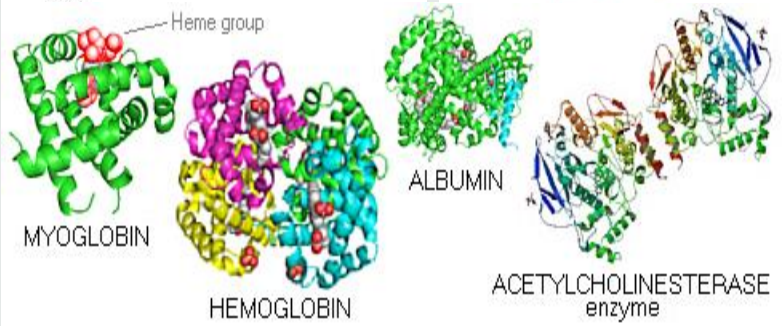
- Pros
  - Simple to understand and easy to implement
  - Zero to little training time (lazy method)
  - No parameters, no need to optimize loss function
  - Quite good accuracy (but other supervised methods are better)
- Cons
  - Computationally expensive
  - Not effective for high-dimension data (use PCA for dimension reduction first)
  - Prediction procedure might be slow
  - Sensitive to the noise (irrelevant data)
  - Memory requirement can be a problem too (Use data structure, like kd-tree)

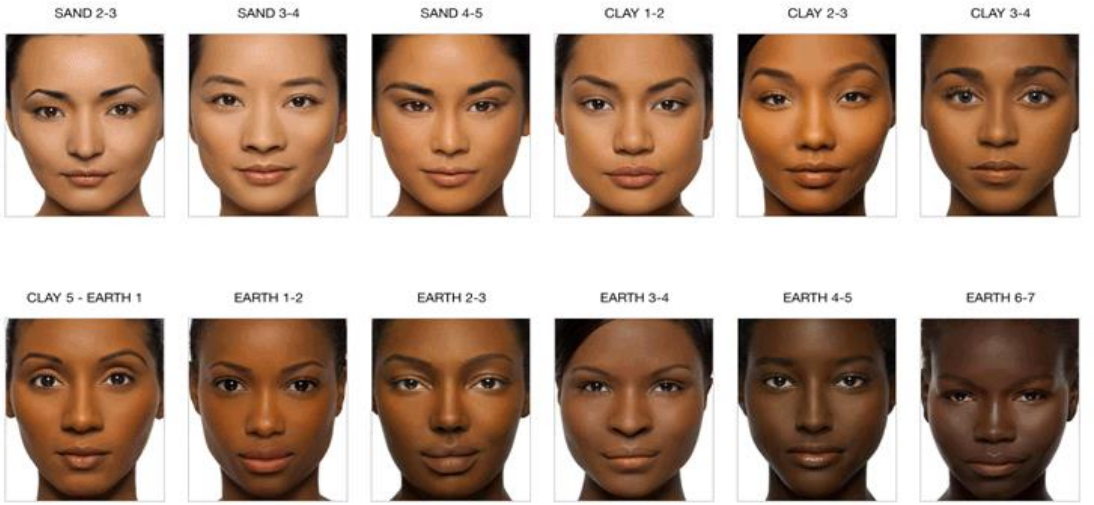
Challenges


TIMES SQUARE


Side View
Top View
Bottom View




Heme group
MYOGLOBIN
HEMOGLOBIN
ALBUMIN
ACETYLCHOLINESTERASE
enzyme

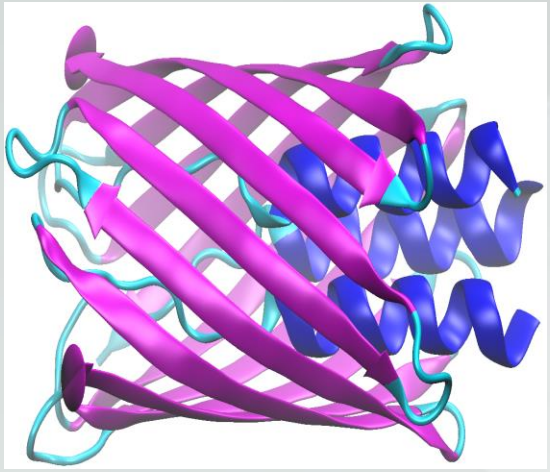

SAND 2-3
SAND 3-4
SAND 4-5
CLAY 1-2
CLAY 2-3
CLAY 3-4
CLAY 5 - EARTH 1
EARTH 1-2
EARTH 2-3
EARTH 3-4
EARTH 4-5
EARTH 6-7

Discussions

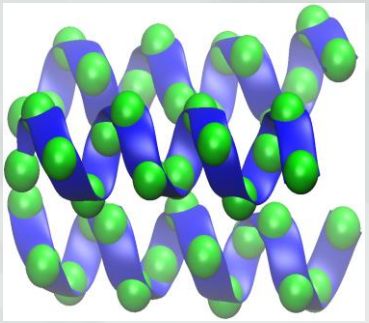
- Appropriate feature selections

Dimension reduction

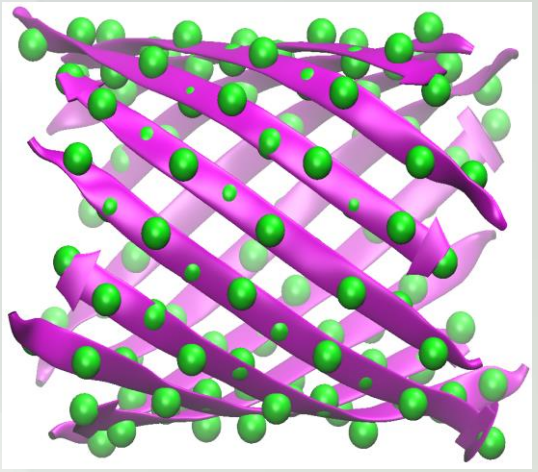
Topological fingerprints of beta barrel

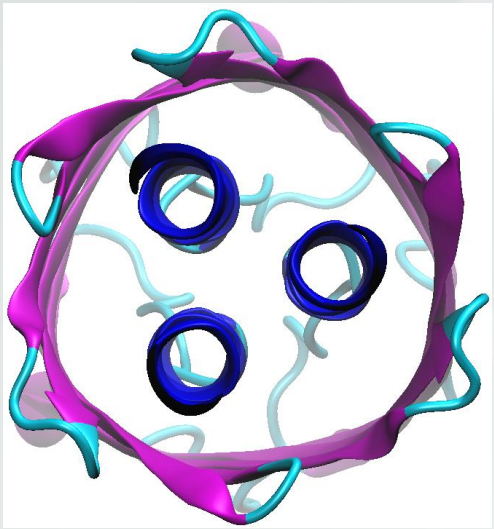


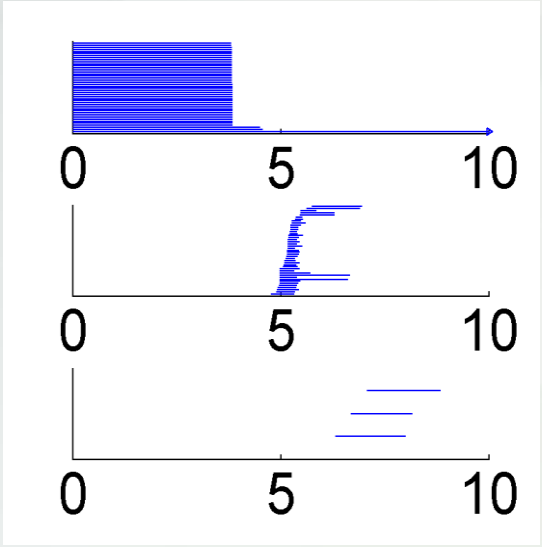
Protein:2GR8

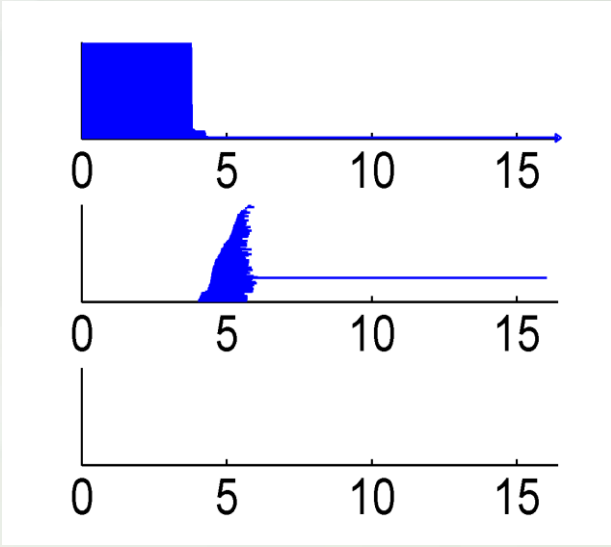


(Xia & Wei, IJNMBE, 2014)



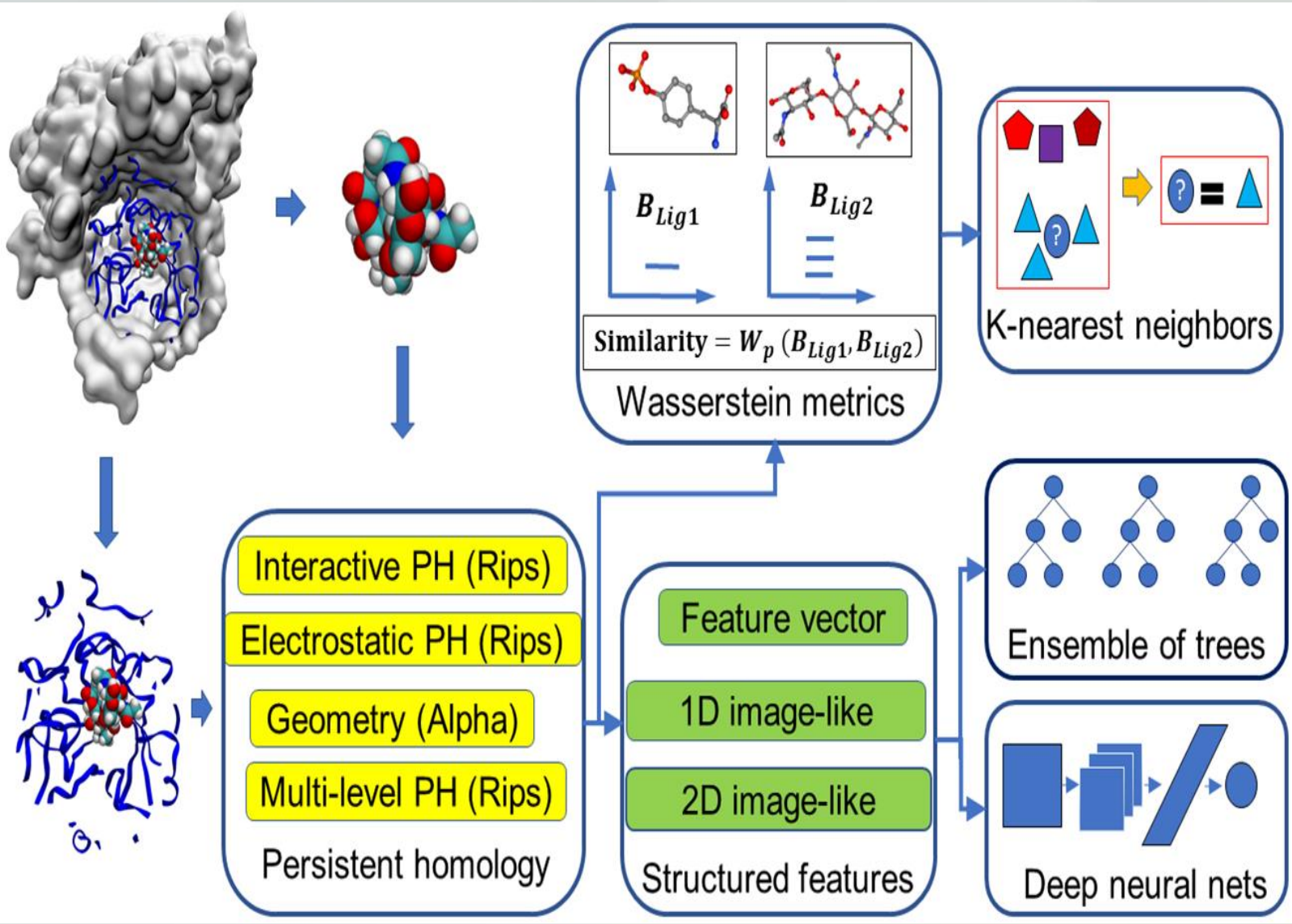






How to compare shapes and fingerprints?

Appropriate metrics



R (RMSE)

0.705 (1.49)

0.741 (1.40)

https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005929

Discussions

Mathematical foundation of data science

- Wasserstein metric
- Lévy metric
- Gromov–Wasserstein Distance
- WGAN (https://arxiv.org/pdf/1701.07875.pdf)
- Kolmogorov-Smirnov distance
- https://arxiv.org/pdf/math/0209021.pdf

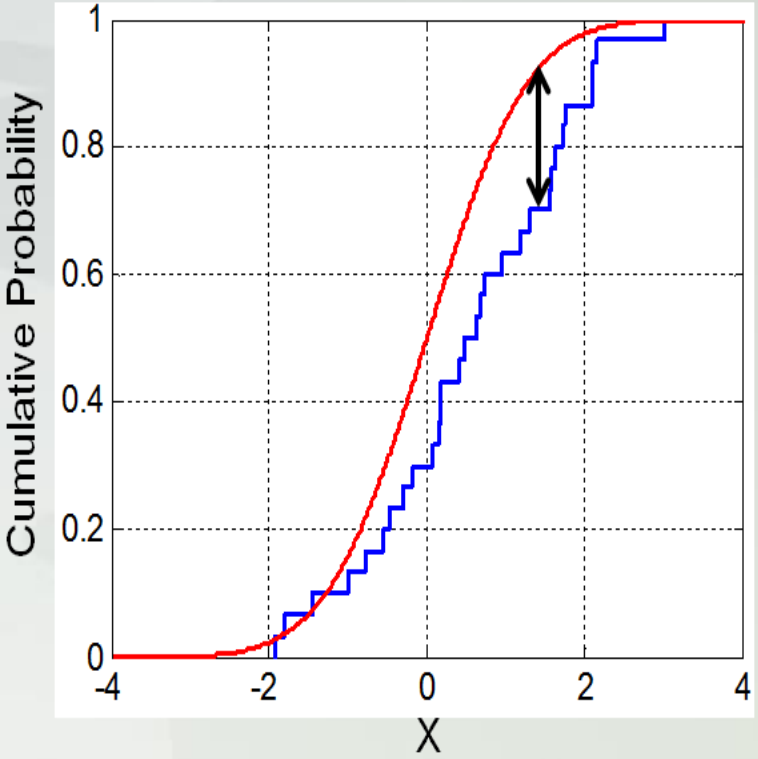
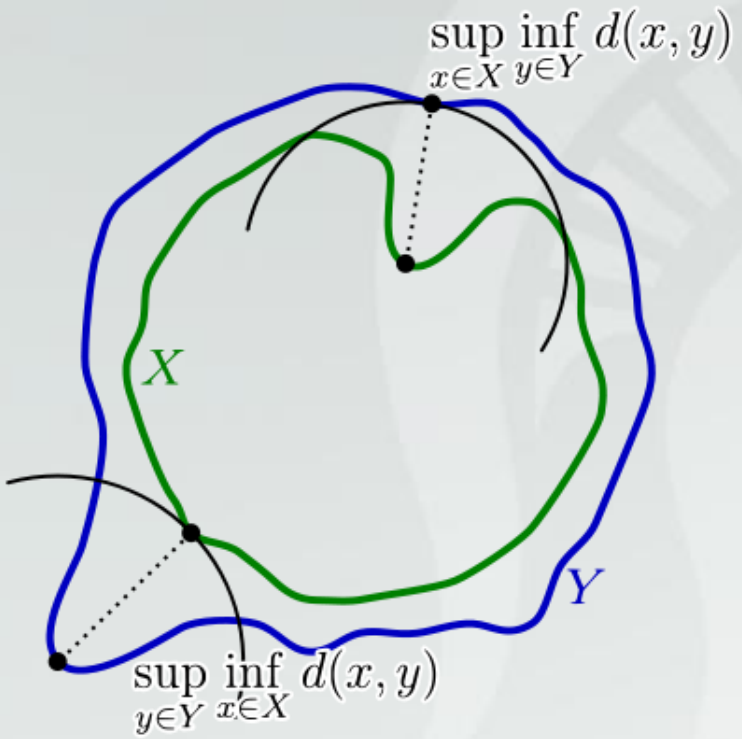


Illustration of the Kolmogorov–Smirnov statistic. Red line is CDF, blue line is an ECDF, and the black arrow is the K–S statistic.

Discussions

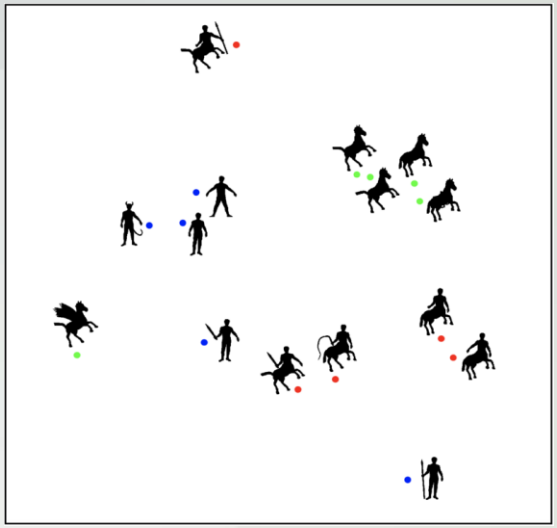
Mathematical foundation of data science

Hausdorff distance



(https://link.springer.com/article/10.1007/s10208-011-9093-5)

Gromov–Hausdorff distance



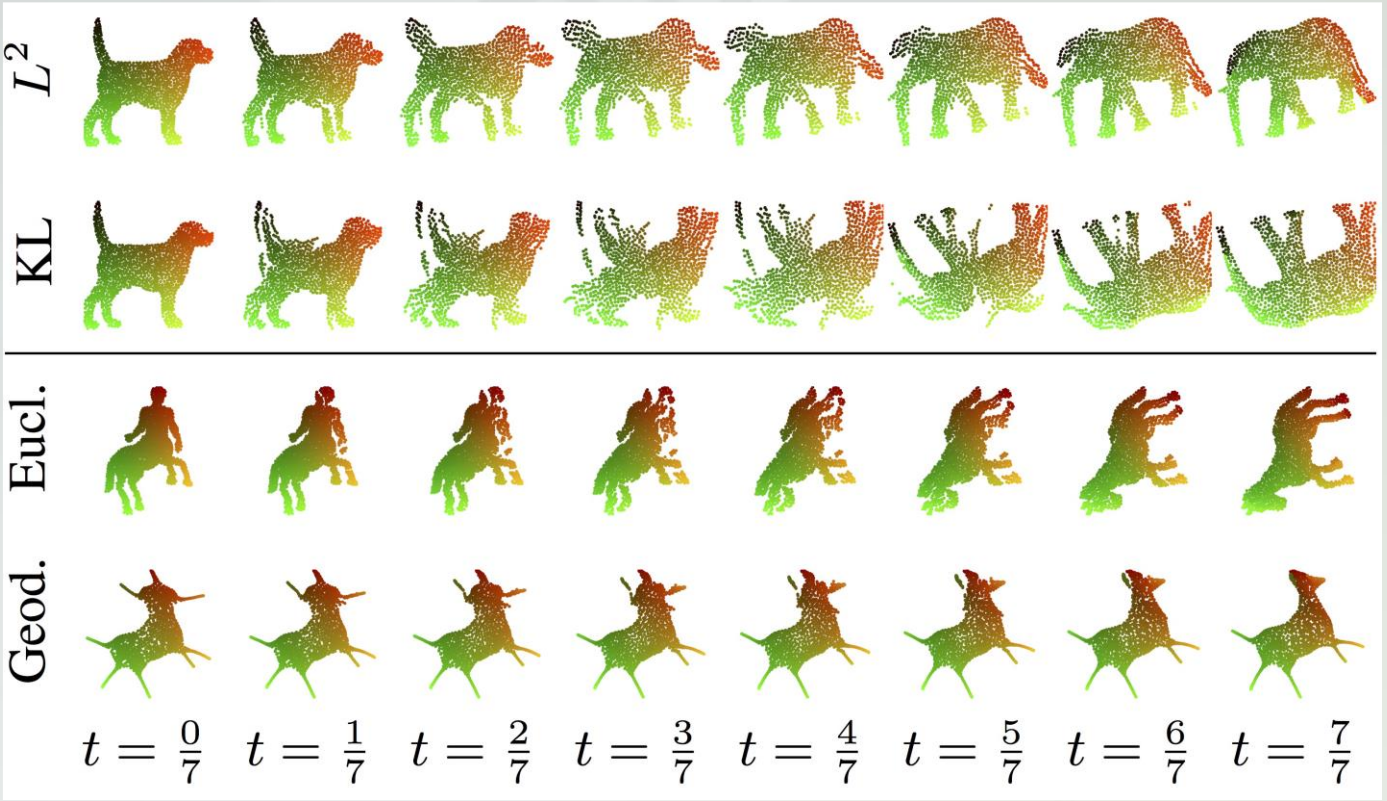
How far and how near are some figures under the Gromov-Hausdorff distance.

Discussions

Mathematical foundation of data science

Gromov-Wasserstein Averaging of Kernel and Distance Matrices

http://proceedings.mlr.press/v48/peyre16.pdf



Upper part: comparison between interpolation using L_2 loss and Kullback-Leibler loss. Lower part: comparison between interpolation using pairwise Euclidean and inner geodesic distances.

Discussions

Mathematical foundation of data science

Gromov-Wasserstein Averaging of Kernel and Distance Matrices

http://proceedings.mlr.press/v48/peyre16.pdf

Mean-Absolute and Root Mean Squared errors for the atomization energy prediction in the QM7 database of 7165 molecules.

Algorithm	MAE	RMSE
k-nearest neighbors	71.54	95.97
Linear regression	20.72	27.22
Gaussian kernel ridge regression	8.57	12.26
Laplacian kernel ridge regression (8)	3.07	4.84
Multilayer Neural Network (1000)	3.51	5.96
GW 3-nearest neighbors	10.83	29.27