Lecture 7: K-Nearest Neighbors (KNN) classifier

Dr. Huiping Cao

K-Nearest Neighbors (KNN) classifier

- It is a lazy classifier.
- It is also called **instance-based learning approach**.
- It does not have a training stage. Instead, in the prediction stage, it directly utilizes the training data to make predictions.

KNN algorithm

- Input
  - *K*
  - A distance metric *metric*
  - Training data
  - Testing instances

Distance

Manhattan distance: $d\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \sum_{k=1}^{m} \left|\mathbf{x}_k^{(i)} - \mathbf{x}_k^{(j)}\right|$

Euclidean distance: $d\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \sqrt{\sum_{k=1}^{m} \left|\mathbf{x}_k^{(i)} - \mathbf{x}_k^{(j)}\right|^2}$

Minkowski distance: $d\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \left(\sum_{k=1}^{m} |\mathbf{x}_k^{(i)} - \mathbf{x}_k^{(j)}|^p\right)^{\frac{1}{p}}$

Example: $\mathbf{x}^{(i)}$ =(1,2,3), $\mathbf{x}^{(j)}$=(1,1,1)

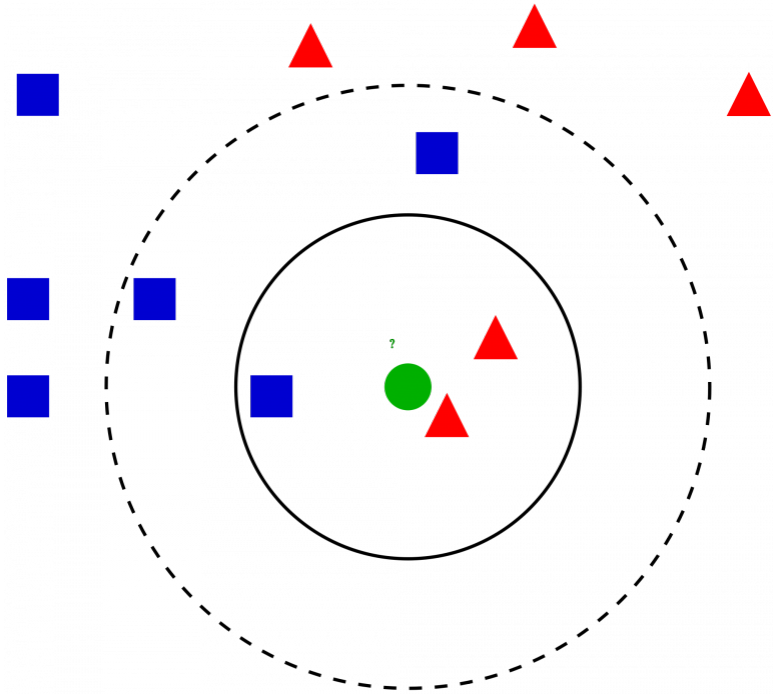
Manhattan distance: |1-1|+|2-1|+|3-1| = 3

Euclidean distance: $\sqrt{(1-1)^2 + (2-1)^2 + (3-1)^2} = \sqrt{5} = 2.24$

Minkowski distance: $\sqrt[3]{(1-1)^3 + (2-1)^3 + (3-1)^3} = \sqrt[3]{10} = 2.08$

KNN algorithm

- Algorithm
  - Find the *K*-nearest neighbors of the sample
  - Assign the class label by majority voting.
- scikit-learn algorithm breaks ties by (1) choosing the neighbors with closer distance and (2) choosing the class label that comes first in the training dataset (when the distances are similar).



Discussions

- **Advantages**
  - Easy-to-interpret output
  - Low calculation time
  - Reasonable prediction power
- **Curse of dimensionality**: when the dimensionality increases, the volume of the space increases dramatically. The available data in the high dimensional space become sparse. Then, neighbors are too far away in a high-dimensional space to make a given good estimate.
  - High-dimensional space: hundreds or thousands of dimensions
- To help avoid curse of dimensionality issue, feature selection or dimensionality reduction techniques can be applied.

KNeighborsClassifier

scikit-learn 0.22.1
class sklearn.neighbors.KNeighborsClassifier(***n_neighbors****=5*, *weights='uniform'*, *algorithm='auto'*, *leaf_size=30*, ***p****=2*, ***metric****='minkowski'*, *metric_params=None*, *n_jobs=None*, ***kwargs*)

- **n_neighbors**: Number of neighbors to use by default for kneighbors queries.
- **p***:* Power parameter for the Minkowski metric. When p = 1, this is equivalent to using manhattan_distance (l1), and euclidean_distance (l2) for p = 2. For arbitrary p, minkowski_distance (l_p) is used.
- **Metric***:* the distance metric to use. The default metric is minkowski, and with p=2 is equivalent to the standard Euclidean metric.