ch5: K-Nearest Neighbor

K-Nearest Neighbors: overview(P8)

Instance-based learning, also called lazy learning.

- simply stores the training instances without learning a model.
- whenever we have new data to classify, we find its K-nearest neighbors from the training data.

K-Nearest Neighbor (KNN) is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure.

> An object is classified to the most common class amongst its k nearest neighbors measured by a “distance” function

Distance Measures(P9-10)

- **Euclidean Distance**

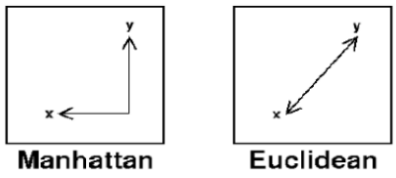
$$D(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ represent the n attribute values of two records.

- doesn’t work well in high dimensions and for categorical variables because it ignores the similarity between attributes.(在高维和分类变量中工作得不好，因为它忽略了属性之间的相似性。)

- **Manhattan Distance** (a.k.a. city block distance)

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



$|x_1 - x_2| + |y_1 - y_2|$

- **Minkowski Distance**

$$D(x, y) = \left(\sum_{u=1}^{n} |x_u - y_u|^p \right)^{\frac{1}{p}}$$

Normalization(P12)

- Standardize the range of independent variables (features of data)

Z-score normalization: rescale the data so that the mean is zero and the standard deviation from the mean (standard scores) is one.

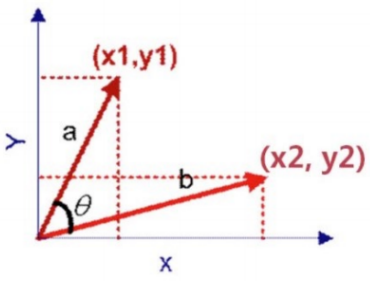
$$x_{norm} = \frac{x - \mu}{\sigma}$$

min-max normalization: scale the data to a fixed range between 0 and 1.

$$x_{morm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Similarity(P13)

- **Similarity:** numerical measure of how alike two data objects are.
  - Is higher when objects are more alike.
  - Often falls in the range [0, 1].
- **Cosine Similarity**



$$\cos(\theta) = \frac{a \bullet b}{||a|| \times ||b||}$$

$$= \frac{(x_1, y_1) \bullet (x_2, y_2)}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}}$$

$$= \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}}$$

$$\cos(d_1, d_2) = \begin{cases} 1: \text{exactly the same} \\ 0: \text{orthogonal} \\ -1: \text{exactly opposite} \end{cases}$$

The Algorithm(P15)

Algorithm

1. Determine parameter K
2. Choose a sample from the test data that needs to be classified and compute its distance to all the training examples.
3. Sort the distances obtained and take the k-nearest data samples.
4. Assign the test class to the class based on the majority vote of its k neighbors.

Decision Boundary(P21)

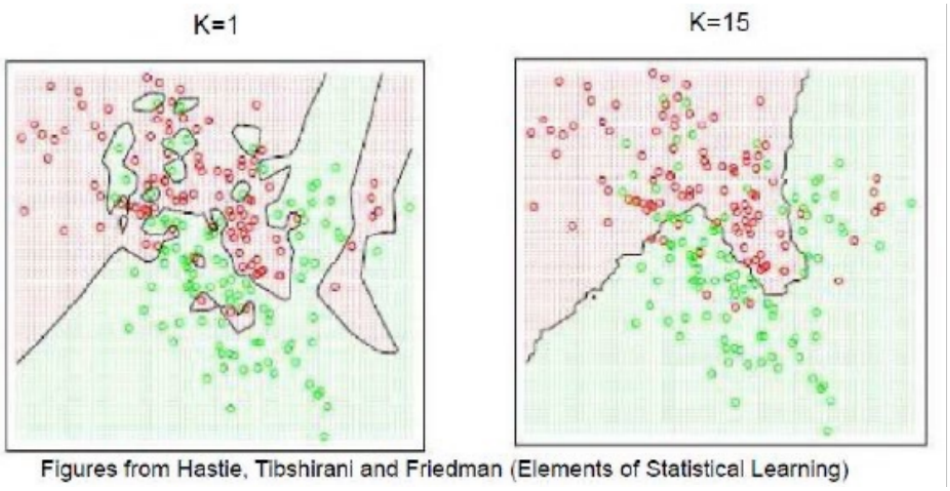
Voronoi Tessellation (沃罗诺伊分割)

- Partition the space into areas that are nearest to any given point
- Boundary: points at the same distance from two different training examples.

Decision Boundary: boundaries that separates two different classes.

Effect of K(P22-23)

- Larger K produces smoother boundary effect
- When K==N, always predict the majority class



Figures from Hastie, Tibshirani and Friedman (Elements of Statistical Learning)

How to choose K?

- If K is too small, efficiency is increased but becomes susceptible to noise.
- Larger K works well. But too large K may include majority points from other classes, but risk of over-smoothing classification results

Pros and Cons(P25-26)

Advantages:

- Simple to understand, explain, and implement,
- No effort for training,
- New data can be added seamlessly without hampering the model accuracy
  新的数据可以无缝地添加，而不妨碍模型的准确性

Disadvantages:

- Does not scale with large data sets (calculating distance is computationally expensive)
- Highly susceptible to the curse of dimensionality 维度增大会很受影响
- Large storage requirements
- Data normalization is required