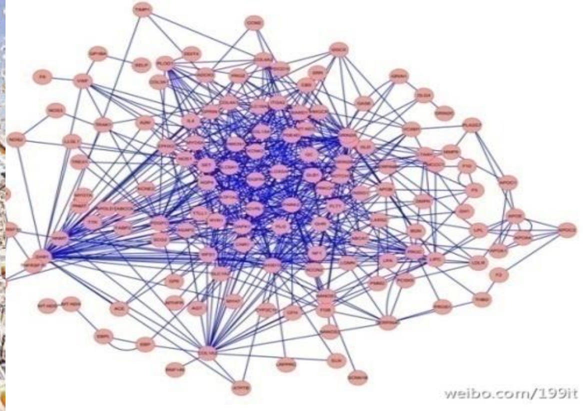


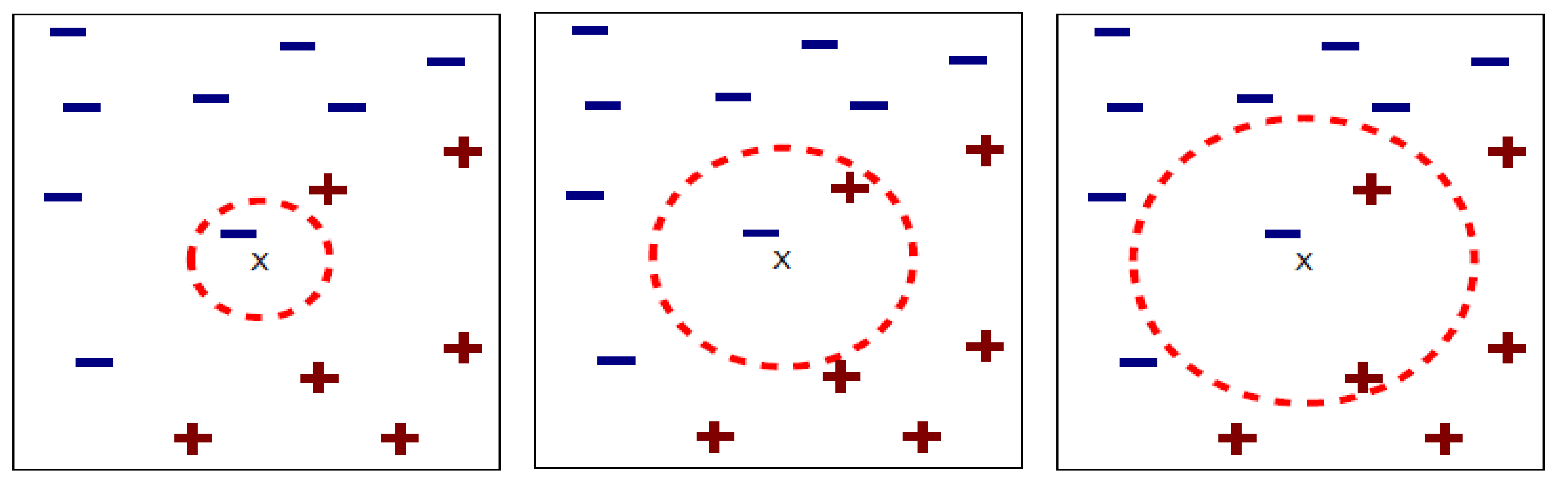
Chapter 5 K-Nearest Neighbors

2025 Autumn

Lei Sun

Basic Idea

- ✓ Definition of nearest neighbor



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Basic Idea

The K-Nearest Neighbors (KNN) algorithm operates on the principle of similarity, where it predicts the label or value of a new data point by considering the labels or values of its K nearest neighbors in the training dataset.

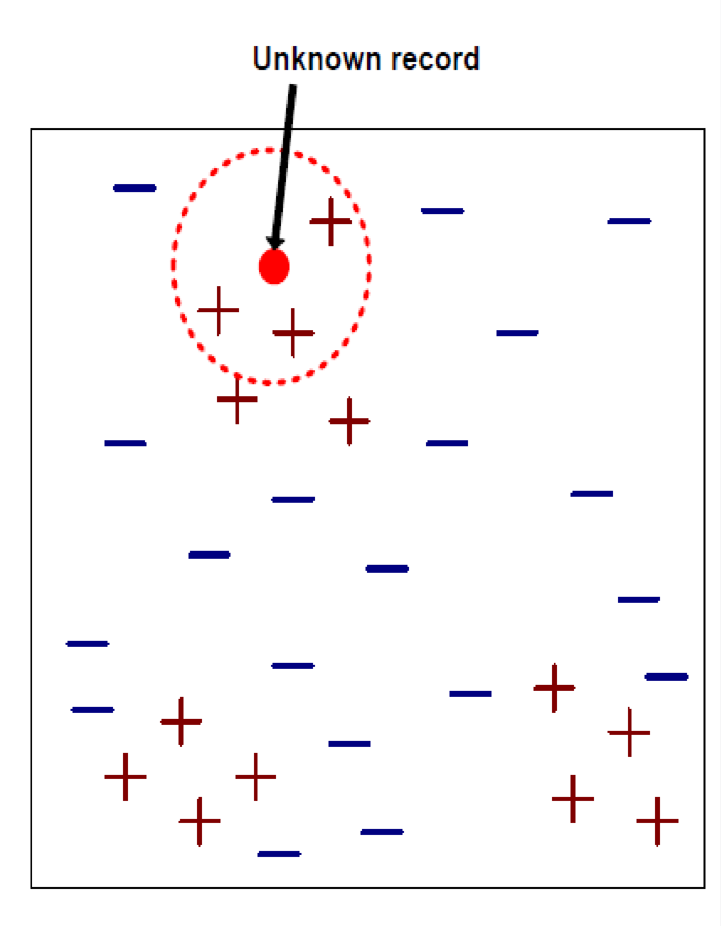
The class or value of the data point is then determined by the majority vote or average of the K neighbors.

similarity — **How to calculate similarity**-- Choose some "distance function" between records

K nearest neighbors — **How to identify K**

Basic Idea

- ✓ Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of k, the number of nearest neighbors to retrieve
- ✓ To classify an unknown record:
  - Compute distance to other training records
  - Identify k nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)



01

Distance

- ✓ Let $d(x,y)$ denote the distance from point x to point y.
- ✓ A legitimate distance should satisfy the following properties:

–Well-defined: $d(x,y) \geq 0$ for any two points x,y

–Identity: $d(x,x)=0$ for any point x

–Symmetry: $d(x,y)=d(y,x)$ for any two points x,y

–Triangle inequality: $d(x,z) \leq d(x,y)+d(y,z)$ for any three points x,y,z

Distance

Minkowski Distance

$$d(x, y) = \left(\sum_i |x_i - y_i|^p \right)^{1/p}$$

Euclidean Distance

$$\mathrm{p} = 2\text{: Euclidean } \mathrm{d(x, y)} = \left(\sum_i |x_i - y_i|^2\right)^{1/2}$$

The Euclidean, as well as the Manhattan distance, are special cases of the Minkowski Distance

Manhattan Distance

$$\mathrm{p} = 1\text{: Manhattan } \mathrm{d(x, y)} = \sum_i |x_i - y_i|$$

Normalization

$$x_{ij}' = \frac{\mathrm{x}_{ij} - \min(x_i)}{\max(x_i) - \min(x_i)}$$

01

Distance

Age	Loan	Default	Distance	
25	$40,000	N	102000	
35	$60,000	N	82000	
45	$80,000	N	62000	
20	$20,000	N	122000	
35	$120,000	N	22000	2
52	$18,000	N	124000	
23	$95,000	Y	47000	
40	$62,000	Y	80000	
60	$100,000	Y	42000	3
48	$220,000	Y	78000	
33	$150,000	Y	8000	1
48	$142,000	?		

$$D = \sqrt{[(48-33)^2 + (142000-150000)^2]} = 8000.01 \rightarrow \text{Default=Y}$$

With K=3, there are two Default = Y and one Default = N out of three closest neighbors. The prediction for the unknown case is again Default = Y.

01

Distance

Age	Loan	Default	Distance
0.125	0.11	N	0.7652
0.375	0.21	N	0.5200
0.625	0.31	N	0.3160
0	0.01	N	0.9245
0.375	0.50	N	0.3428
0.8	0.00	N	0.6220
0.075	0.38	Y	0.6669
0.5	0.22	Y	0.4437
1	0.41	Y	0.3650
0.7	1.00	Y	0.3861
0.325	0.65	Y	0.3771
0.7	0.61	?	

Using the standardized distance on the same training set, the unknown case returned a different neighbor which is not a good sign of robustness.

Distance

scaling

- ✓ Different features may be measure differently in a manner that may make the distance between observations meaningless (or less meaningful)
- ✓ This necessitates scaling or standardization to avoid issues with analysis
- ✓ Z-score
- ✓ Min-Max

$z = (x - \mu)/\sigma$

Where:

•x: is the test value

•μ: is the mean

•σ: is the standard value

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

02

KNN

break KNN down into steps:

Step #1 - Assign a value to K.

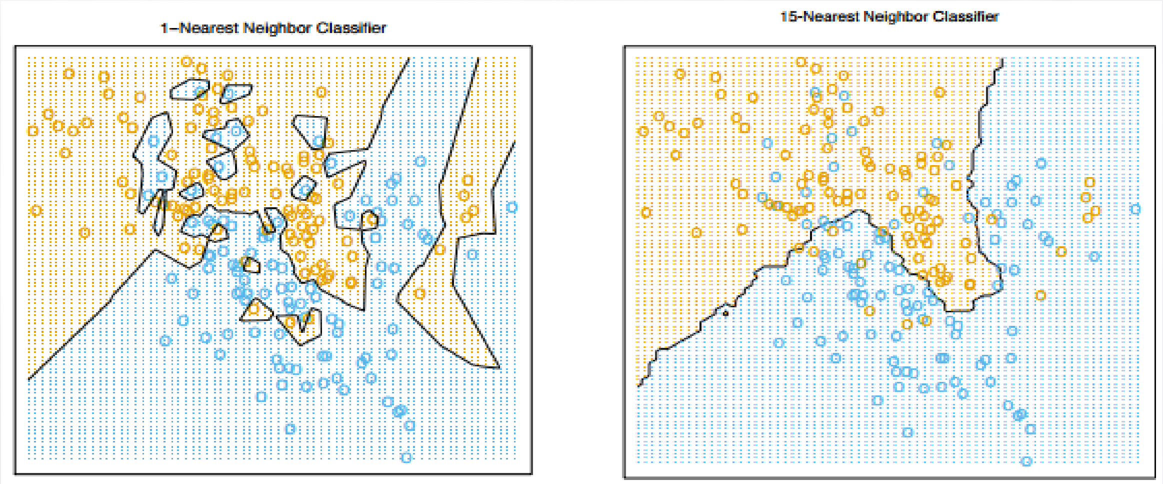
Step #2 - Calculate the distance between the new data and all other existing data. Arrange them in ascending order.

Step #3 - Find the K nearest neighbors to the new data based on the calculated distances.

Step #4 - Assign the new data to the majority class in the nearest neighbors.

02

KNN



Low k VS. High k

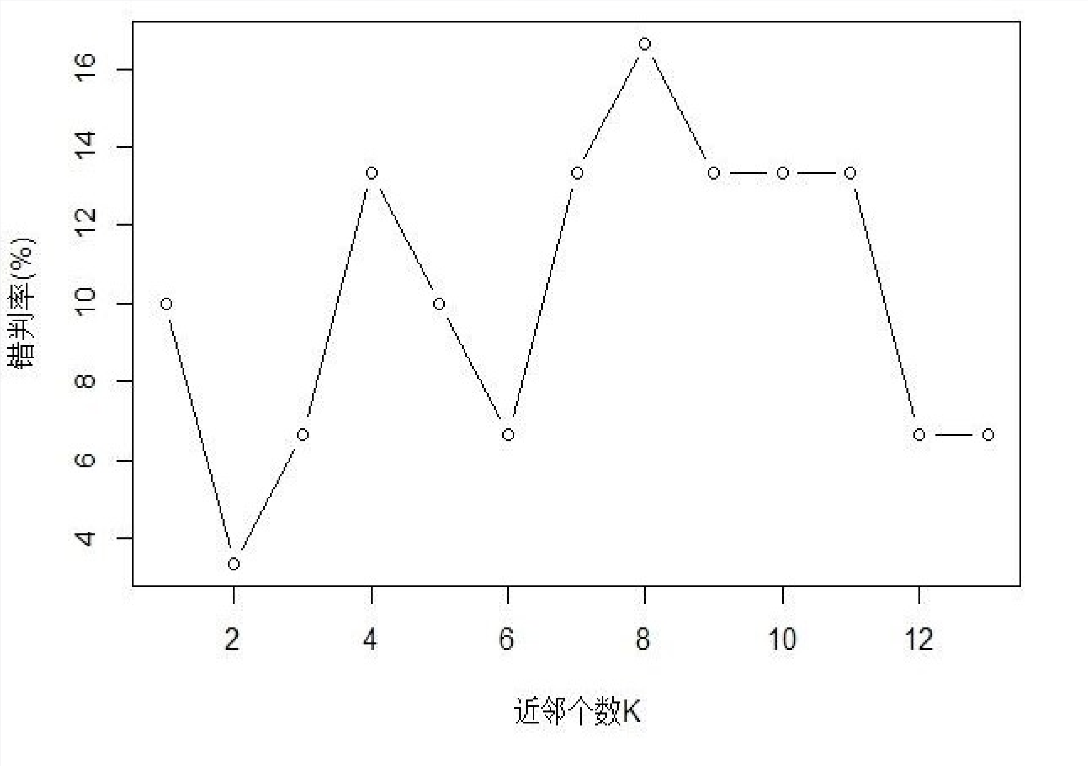
- ✓ Low values of k capture local structure in data (but also noise)， make the model more sensitive to noise but might overfit. high variance, but low bias
- ✓ High values of k provide more smoothing, less noise, but may miss local structure. high bias and lower variance
- ✓ it is recommended to have an odd number for k to avoid ties in classification, and cross-validation tactics can help you choose the optimal k for your dataset.

02

KNN

Choosing ‘K’ is crucial.

To select the value of K that fits data, run the KNN algorithm multiple times with different K values.



Choose K

The odd value of “K” is preferred over even values to avoid ties in voting.

Cross-validation can help to choose the optimal k for dataset.

02

KNN

Advantages	Disadvantages
Easy to implement: easy to understand and implement	**Computational cost:** especially with large datasets, as it need calculate distances for each data. Takes up more memory and time.
No training phase(lazy algorithm): doesn't require a separate training phase.	**Limited to Euclidean Distance:** This can be a disadvantage when working with non-Euclidean data, such as categorical or binary data.
Few parameters: only requires a K and a distance metric compared to other ML algorithm	**Requires Good Choice of K:** If K is too small, the algorithm may be too sensitive to noise in the data, while if K is too large, the algorithm may miss important patterns in the data.

Weighted KNN Based on Variable Importance

Break into following steps

① Identify K

② Exclude input variables one by one, compute error e_i

③ The importance of the *i*th variable

$$\mathrm{FI}_i = \mathrm{e}_i + \frac{1}{p}$$

p : the number of variables

④ Compute weighted distance： important variables with higher weight

$$EUCLID(x, y) = \sqrt{\sum_{i=1}^{p} \mathrm{w}_i (x_i - y_i)^2}$$

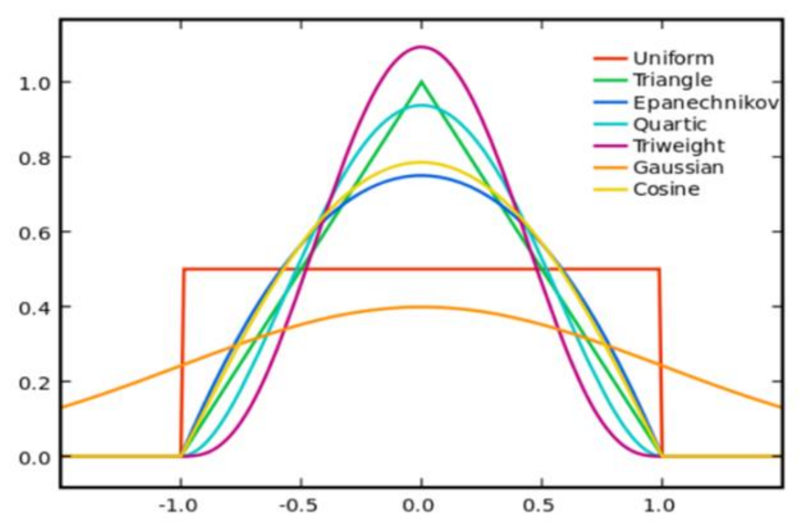
$$\mathrm{w}_i = \frac{FI_i}{\sum_{j=1}^{p} FI_j}$$

Weighted KNN Based on Variable Importance

Two issues

- ✓ KNN believes that K nearest neighbors have an equal impact on the prediction results. KNN认为K个近邻对预测结果有同等的影响
- ✓ When an input variable is categorical or ordinal, the calculation of Euclidean distance is no longer appropriate.
  当某个输入变量是分类型或顺序型时，欧几里得距离计算不再恰当

Weighted KNN Based on Similarity



Idea

- ✓ Weights： depends on similarity
- ✓ Define similarity as a nonlinear function of the distance between each observation and X_0.
- ✓ The higher is the weight, the more important is variable .
- ✓ The closer the distance, the stronger the similarity.
- ✓ Common kernel function
  - Triangle core
  - Gaussian core

$$\begin{cases} K(d) \geq 0 \\ d = 0, K(d) & \text{: the biggest} \\ K(d) & \text{: monotone reduction function} \end{cases}$$

$$K(d) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{d^2}{2}) \cdot I(|d| < 1)$$

$$\mathrm{I(d)} = \begin{cases} 1, |d| < 1 \\ 0, |d| \geq 1 \end{cases}$$

04

Weighted KNN Based on Similarity

Steps of computing similarity

① Preprocess input variable value

numerical data:

$$z_{ij} = \frac{x_{ij}}{\sigma_{ij}}$$

categorical variable

表 4.1 m=5 的分类型变量的虚拟变量

类别值	v_{i1}	v_{i2}	v_{i3}	v_{i4}	v_{i5}
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1

ordinal variable

表 4.2 m=5 的顺序型变量的虚拟变量

类别值	v_{i1}	v_{i2}	v_{i3}	v_{i4}
1	1	1	1	1
2	−1	1	1	1
3	−1	−1	1	1
4	−1	−1	−1	1
5	−1	−1	−1	−1

② Compute distance

$$d(Z_i, Z_0) = \sqrt{\sum_{j=1}^{p} \left| Z_{(i)j} - Z_{(0)j} \right|^2}$$

The Euclidean distance between the *i-th* observation point X_i and (new point)X_0

Sum of squares of subtraction of corresponding dummy variable

Eliminate the effect of the number of dummy variable on distance

Sum/m

m is the number of dummy variables

04

Weighted KNN Based on Variable Importance

Steps of computing similarity

③ Change distance to similarity by kernel function

- Find (K+1)th neighbor

$$d(Z_i, Z_0) = \sqrt{\sum_{j=1}^{p} \left| Z_{(i)j} - Z_{(0)j} \right|^2}$$

- Standardize distance

$$D(Z_i, Z_0) = \frac{\mathrm{d}(Z_i, Z_0)}{\mathrm{d}(Z_{k+1}, Z_0)}, i = 1,2, \ldots, k$$

The K+1th nearest neighbor is farthest from Xo

- Calculate similarity

$$\mathrm{w}_i = \mathrm{K}(D(Z_i, Z_0))$$

$$K(d) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{d^2}{2}) \cdot I(|d| < 1)$$

04

Weighted KNN Based on Variable Importance

summary

① Identify X_0

② Find (k+1)th nearest neighbor by $d(Z_i, Z_0) = \sqrt{\sum_{j=1}^{p} \left|Z_{(i)j} - Z_{(0)j}\right|^2}$

③ Calculate weights: $w_i = K(D(Z_i, Z_0))$ $D(Z_i, Z_0) = \frac{d(Z_i, Z_0)}{d(Z_{k+1}, Z_0)}, i = 1,2, \ldots, k$

④ Predict:

$$y_0 = max_r\left(\sum_{i=1}^{k} w_i\, I(y_i = r)\right)$$

I is a demonstrative function(示性函数)

The sum of weights for neighbors belonging to class r among the K neighbors of X_0 is maximum, so X_0 is classified to r category.

在KNN算法中，如果K值设置得太小，可能会出现以下哪种情况？

A. 过拟合

B. 欠拟合

C. 分类效果变好

D. 计算效率提高