Sun Yat-Sen University, School of Data and Computer Science

k-nearest neighbors algorithm

Yaowei Wang, Xueying Zhan

September 19, 2016

1 Overview

In Machine Learning, the k-Nearest Neighbors algorithm (or k-NN for short) is a **non parametric** method used for **classification** and **regression**. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

- In k-NN classification, the output is a class **label** (for one instance). An object is classified by **majority voting** of its neighbors, with the object being assigned to the class which is most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.
- In k-NN regression, the output is the property **value** for the object. This value is the **average** of the values of its k nearest neighbors.

2 Dataset and Process

In this part, you must know two kinds of datasets, text/numeric dataset (classification and regression). **They are only different from their label (output)**. We assume only two emotions (happy and sadness). The example as follows:
Classification:
You must process this dataset using any method, such as one-hot, TF, TF-IDF. We show a simple example as follow:
Regression:

Document number	The sentence words	emotion
train 1	I buy an apple phone	happy
train 2	I eat the big apple	happy
train 3	The apple products are too expensive	sadnesss
test 1	My friend has an apple	?

Table 2.1: example of classification dataset

Document number	I	buy	an	apple	...	friend	has	emotion
train 1	1	1	1	1	...	0	0	happy
train 2	1	0	0	1	...	0	0	happy
train 3	0	0	0	1	...	0	0	sadness
test 1	0	0	1	1	...	1	1	?

Table 2.2: process of classification dataset

Document number	The sentence words	the probability of happy
train 1	I buy an apple phone	0.8
train 2	I eat the big apple	0.6
train 3	The apple products are too expensive	0.1
test 1	My friend has an apple	?

Table 2.3: example of regression dataset

You must process this dataset using anything method, such as one-hot, TF, TF-IDF. We show a simple example as following:

Document number	I	buy	an	apple	...	friend	has	probability
train 1	1	1	1	1	...	0	0	0.8
train 2	1	0	0	1	...	0	0	0.6
train 3	0	0	0	1	...	0	0	0.1
test 1	0	0	1	1	...	1	1	?

Table 2.4: process of regression dataset

3 K-NN CLASSIFICATION

In k-NN classification, the k-NN algorithm is used for estimating **discrete** variables. One such algorithm uses a mode label of the k nearest neighbors. This algorithm works as follows:

- Compute the **Euclidean** or Mahalanobis distance from the query example to the labeled examples.
- **Order** the labeled examples by increasing distance.

- Find a heuristically **optimal** number k of nearest neighbors.
- Find the **mode label** with the k-nearest multivariate neighbors.

An demo based on the previous classification dataset as following:
In previous dataset, the query example is test 1 and the labeled examples are train 1,2,3. We compute their Euclidean distances by the formulation.

$$d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

We will get three different distances:

$$d(train1, test1) = \sqrt{(1-0)^2 + (1-0)^2 + \cdots + (0-1)^2} = \sqrt{6};$$
$$d(train2, test1) = \sqrt{(1-0)^2 + (1-0)^2 + \cdots + (0-1)^2} = \sqrt{8};$$
$$d(train3, test1) = \sqrt{(0-0)^2 + (0-0)^2 + \cdots + (0-1)^2} = \sqrt{9};$$

We find the d(train3,test1)>d(train2,test1)>d(train1,test1), that is, the text train 1 is the nearest neighbour of the test 1. They are has the similar class label. Suppose we set the k = 3. Therefore, we need 3 neighbour of test 1. It's train 1,2,3. Finally, we find the mode label of their neighbour. In the example, we have 2 happy, 1 sadness. So we can forecast the emotion of test 1 is happy.

4 K-NN REGRESSION

In k-NN regression, the k-NN algorithm is used for estimating **continuous** variables. One such algorithm uses a weighted average of the k nearest neighbors, weighted by the inverse of their distance. This algorithm works as follows:

- Compute the **Euclidean** or Mahalanobis distance from the query example to the labeled examples.
- **Order** the labeled examples by increasing distance.
- Find a heuristically **optimal** number k of nearest neighbors.
- **Calculate an inverse distance weighted average** with the k-nearest multivariate neighbors.

An demo based on the previous classification dataset as following:
The step 1 to 3 is the same as the clarification. When we get d(train1,test1), d(train2,test1) and d(train3,test1) and we set k = 3, use their inverse as the output weight.

$$\begin{aligned} P(test1\ is\ happy) &= \frac{train1\ probability}{d(train1, test1)} + \frac{train2\ probability}{d(train2, test1)} + \frac{train3\ probability}{d(train3, test1)} \\ &= 0.47 \end{aligned}$$

5 SPECIAL CASE

When we set the **k=1**, the method is called the **1-nearest neighbour**. In this case, we only assign a point x to the label of its **closest neighbour** in the feature space.

6 MORE METHODS

In previous introduction, we use the Euclidean distance as our method to find their neighbour. If we use **different distance ways**, such as City block distance, Supremum distance and so on. we can get high accuracy. Do you have other ideas to find their neighbours? Besides, if we **change the k**, the result can be different. Is the result good or bad? Why?
In the k-NN regression, we use the inverse of the distances as a probability output. But **the value is too big**, the final probability is more than 1. How to deal with this problem? In this case, we will use the **normalization the weight**.
In statistics and applications of statistics, normalization can have a range of meanings. In the simplest cases, normalization of ratings means adjusting values measured on different scales to a notionally common scale, often prior to averaging.
There are two common normalization methods in statistics.

Name	Formula	Explain
Standard score	$X' = \frac{X-\mu}{\sigma}$	μ is the mean and σ is the standard deviation
Feature scaling	$X' = \frac{X-X_{min}}{X_{max}-X_{min}}$	X_{min} is the min value and X_{max} is the max value

Table 6.1: two normalization methods

7 LAB TASKS

Lab2 consists of two tasks: k-NN classification task and regression task.

- Classification Task: Use semeval dataset for classification. Because semeval have six emotions (1-anger, 2-disgust, 3-fear, 4-joy, 5-sad and 6-surprise). Every instance only belongs to one emotion. You must implement the 1-NN classification with Euclidean distance. If you have sufficient time, you try to change the k value, different distance and any improving method. Please describe your discovery and interesting phenomenon in your experimental report.

- Use semeval dataset for regression. If you achieve the classification task, I believed that you have loved this algorithm. To test your learning level, we will give you a regression data without the answer of the test dataset. But you can debug and improve your code via the validation dataset. Please submit your predication results of test dataset.

8 SUBMISSION

- Experiment Report, including:

  -The classification results on testing set, use **accuracy** to describe.

  -The regression results on validation set, use **coefficient** to describe.

  -Explain how you get the results.

  -The illustration that describes the codes in detail.

  -The innovation of this experiment, if you have. give experimental results and illustration.

- Regression Result

  Please submit the file using "IDnumber_name_regression.txt".

  The format of this file

  test1 0.1 0.1 0.1 0.1 0.1 0.1

- Code

- FTP server: ftp://my.ss.sysu.edu.cn/~ryh

9 ACKNOWLEDGMENTS

Thanks to my excellent roommate **YiLin Zheng** for dealing with the data from raw HTML version. We are grateful to the other TAs and their suggestion on this lab.