K-Nearest-Neighbors

Jiaqi Zeng 15220152202409

April 12, 2019

Abstract

This paper will talk about K-Nearest-Neighbors and k-nearest neighbors algorithm. We will introduce different distance to measure similarity, how to choose k value and KNN using in high dimension.

1 Introduction

In pattern recognition, the k-nearest neighbors algorithm (KNN) 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 KNN is used for classification or regression:

- In KNN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class 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 KNN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors.

For example, if k=1 and object nearest neighbor's value is 10, then we predict object's value is 10, too. If k=3 and nearest three values are 6, 8 and 7, then we predict object's value is 7 (mean of the nearest three neighbors).

2 How to choose a k value?

How to choose a k value is an important question with facing a bias-variance tradeoff.

- When K = 1, the KNN training error rate is 0. Bias is low but the variance is high.
- In KNN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors.

The best choice of k depends upon the data; generally, larger values of k reduces effect of the noise on the classification, but make boundaries between classes less distinct.

A good k can be selected by various heuristic techniques. The special case where the class is predicted to be the class of the closest training sample (i.e. when k = 1) is called the nearest neighbor algorithm.

The accuracy of the KNN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance. Much research effort has been put into selecting or scaling features to improve classification.

A particularly popular approach is the use of evolutionary algorithms to optimize feature scaling. Another popular approach is to scale features by the mutual information of the training data with the training classes.

In binary (two class) classification problems, it is helpful to choose k to be an odd number as this avoids tied votes. One popular way of choosing the empirically optimal k in this setting is via bootstrap method.

3 Measure of Distance

Both for classification and regression, a useful technique can be used to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.

For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor. But 1/d is so simple that can't measure the distance accurately, so people find more ways to measure the distance.

Euclid Distance:

$$d = \sqrt{\sum_{i=1}^{N} (x_{1i} - x_{2z})^2}$$

Mahalanobis distance:

$$\sqrt{(x - u_x)^T \Sigma^{-1} (y - u_y)}$$

, where

$$u_x = E(X)$$
$$X = \{x_1, x_2, \ldots, x_N\}$$
$$u_y = E(Y)$$

$$X = \{y_1, y_2, \ldots, y_N\}$$

Manhattan distance

$$\sum_p \|I_I^p - I_2^p\|$$

, where p is dimension of I.

4 Feature Extraction

When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters) then the input data will be transformed into a reduced representation set of features (also named features vector).

Transforming the input data into the set of features is called feature extraction.

If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input.

Feature extraction is performed on raw data prior to applying KNN algorithm on the transformed data in feature space.

5 KNN in high dimension

KNN suffers from the curse of dimensionality: given N, when p is large, data become relatively sparse. In high dimensions, the neighborhood represented by the K nearest points may not be local.

For high-dimensional data, dimension reduction is usually performed prior to applying the KNN algorithm in order to avoid the effects of the curse of dimensionality.

Feature extraction and dimension reduction can be combined in one step using principal component analysis (PCA), linear discriminant analysis (LDA), or canonical correlation analysis (CCA) techniques as a pre-processing step, followed by clustering by k-NN on feature vectors in reduced-dimension space. In machine learning this process is also called low-dimensional embedding.

6 Conclusion

In general, KNN can work well for prediction in a wide variety of situations, since they dont make any real assumptions. But KNN needs lots of data to give

an accurate estimation.In contrast, parametric methods summarize the data with a fixed set of parameters, which are sufficient for prediction. Whether to choose KNN is a tradeoff.

References

[1] Altman, N. S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". The American Statistician. 46 (3): 175185.doi:10.1080/00031305.1992.10475879. hdl:1813/31637

[2] Cover TM, Hart PE (1967). "Nearest neighbor pattern classification" (PDF). IEEE Transactions on Information Theory. 13 (1): 2127.

[3] Toussaint GT (April 2005). "Geometric proximity graphs for improving nearest neighbor methods in instance-based learning and data mining". International Journal of Computational Geometry and Applications. 15 (2): 101150.