UNSW
THE UNIVERSITY OF NEW SOUTH WALES

COMP9417 Project Report

Topic 2.2: Nearest Neighbour

z5191513 JIQING WANG

z5124871 WEI ZHOU

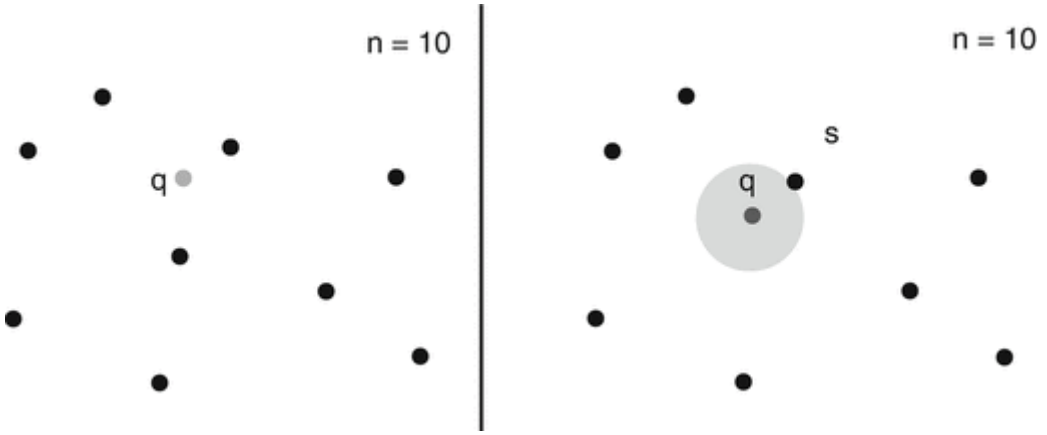
Introduction

'Nearest Neighbour' is a famous optimization problem in machine learning by finding the nearest points, which is widely used in many fields such as statistical classification and similarity scores. Formally, this problem is defined as, given a set S which contains points in a space P and a query point p in space P, find the closest point s in S to p.

This project implements both k-NN and w-NN to do classification and numeric prediction with different data sets (classification—ionosphere, numeric prediction—autos). Manhattan distance and Euclidean distance are used for both methods. Also, a new dataset for classification is constructed to test the performance of classifiers (k-NN & w-NN) based on the Bayes error rate.

This report takes on the following structures:

- Explanation of the data set used in the project.
- Explanation of related methods for implementing the classification and numeric prediction
- Description and evaluation of the implementation
- Conclusion and further related applications.



Retrieved from https://link.springer.com/referenceworkentry/10.1007%2F978-3-319-17885-1_869

Methods

1. Datasets Explanation

1.1 ionosphere.arff (Classification)

This file is used for generating a classification model in the first part of this experimentation, which consists of 351 instances, valued between 0 and 1, 34 continues attributes plus the class label, and the class is either “g”(good) or “b”(bad), they are the true outputs for each instance. The data set does not contain any missing values. Furthermore, the performance of the classification model is evaluated by comparing the difference between predicted results generated by the learning model and the true class attributes indicated in this file.

1.2 autos.arff (Numeric Prediction)

It contains 26 attributes composed of 15 continuous, 1 integer and 10 nominal ones. Our task is predicting the continuous attribute ‘price’ (ranging from 5118 to 45400) by applying k-NN and w-NN algorithms on other attributes.

2. Leave-one-out Cross-validation

Assume there are n instances in the dataset, take 1 instance as test set and other (n - 1) instances into the training set. Repeat this process for n times until all the instances have been selected as the test set. For each time, a model is built based on the training output and tested on the test set.

Retrieved from: https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85

A stable and reliable model could be obtained by cross-validation which avoids overfitting problem on the training set. Additionally, leave-one-out cross-validation retains the real distribution of the datasets optimally.

3. Distance Measurements

It is widely used in physics and mathematics, which is a function or metric to describe "how close" or "how far apart" the elements are. In this project, the following two distance measures are used for implementing kNN and wNN:

3.1 Manhattan Distance

$$\text{Dist}(x, y) = |x_1 - x_2| + |y_1 - y_2|$$

Manhattan distance, also known as L1-norm, which is a method to measure the distance by summing all the horizontal and vertical difference between the coordinates of two data points.

3.2 Euclidean Distance

$$\text{Dist}(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

It is the most commonly used distance measurement in Machine Learning, and the generalized term is L2-norm, which calculates the square root of the sum of the squared differences between the corresponding coordinates of two points(ref.4).

4. k-NN (K-Nearest Neighbour)

Assume there are n instances in the training dataset, find the k nearest points to the query point which has to be predicted by calculating the distance between each position of the instances and the query point.

4.1 Classification

Counting the number of each class among the k instances, assign the class of query point with the class which has the highest frequency of appearance.

4.2 Numeric Prediction

Taking the mean of the prediction attribute value among the k instances, assign the attribute value of the query point with this mean.

5. w-NN (Weighted Nearest Neighbour)

This method is similar to the k-nearest neighbour above. The difference between these 2 methods is weights are added while determining the prediction value for the query point. The weight formula used in this project is shown below.

$$w_i \equiv \frac{1}{d(x_q, x_i)^2}$$

6. Target function

$$\mathrm{P}(y_1|x_1) \quad x_1 \in X, y_1 \in Y \qquad X - set\ of\ instances, Y - class\ set$$

The target function is a function that calculates the probability of hitting a target. If an arbitrary set of random variables X is given, choose a random subset from X as input, then this function will produce a probabilistic prediction as its output(ref.4). For instance, given a set of products X and a class Y with labels, “cheap” and “expensive”, a learner needs to predict which products are cheap or expensive based on the price (price is a continuous value), by using the above approach, a target function can be simply constructed, $\mathrm{P}(\mathrm{Y} = \mathrm{cheap}|\mathrm{An\ apple} > 5\$) = 0.1$ and $\mathrm{P}(\mathrm{Y} = \mathrm{expensive}|An\ apple \leq 5\$)$

, which means the probability of this learner will output “expensive” is 10% if the price of “An apple” is higher than 5$ where 90% that its outputs “cheap” when the price is lower than 5$.

7. Bayes error rate

$$\mathrm{error} = \min \mathrm{P}(y|x)\mathrm{P}(y), x\ is\ an\ instance, y \neq C(x), C\ is\ the\ true\ class$$

The lowest possible test error rate the classifier outputs is the Bayes error rate, and sometimes, it can be interpreted as an irreducible error (ref.2). For multi-class classification, the set of class labels can be divided into 2 or more independent subsets, each instance belongs to more than one label. The Bayes error rate of the data distribution measures the probability a classifier misclassified a random instance when the true class label is known. Back to the previous example, if P(expensive) and P(cheap)

are both 50%, then the Bayes error rate is 10% * 50% = 5%, that is the probability that the classifier will misclassify “An apple (price > 5$)” to “cheap”. The Bayes error rate is an important concept in the study of statistical classification.

Experimentation Process

1.k-NN & w-NN

1.1. Read Datasets

Read the datasets ionosphere.arff and autos.arff and transform them into the data frame form for later use. For autos.arff which is used for numeric prediction, we delete the categorical attributes at this step.

Split up these datasets into attributes sets (X) and prediction set (Y) for use in cross-validation.

1.2. Selection of k

Classification, k-NN: k is from 1 to 50

Classification, w-NN: k is all odds from 1 to 50

Numeric Prediction, k-NN: k is from 1 to 20

Numeric Prediction, w-NN: k is from 1 to 20

1.3. Leave-one-out Cross-validation

Suppose we have n instances in the whole dataset. Split up both X and Y into (n-1) training sets (X_train, Y_train) and 1 test set (X_test, Y_test). Then, calculate the distance between the points in X_train and the point in X_test and choose the nearest k points to do further classification or numeric prediction. Repeat this process for n times with different test sets to raise n tests.

1.4. Classification

For the nearest k points, count the number of each class label (g & b) among them (The information could be found in Y_train) and compare the 2 numbers to select a class label with more votes in these k points. Regard this test as a ‘correct’ prediction if the chosen class label is the same as the one in Y_test, or ‘wrong’ prediction if not.

For k-NN, only odd k values are chosen to avoid the situation that the number of ‘correct’ equals to the number of ‘wrong’.

For w-NN, calculate the weights by the formula given in the ‘METHODS’ part, add them to evaluate the prediction class label. (By the following formula, if the result > 0.5, ‘correct’. Otherwise, ‘wrong’) If the distance is 0, the weight is set to 1/0.0000000000000001.

$$\frac{\sum_{i=1}^{k} weight_i * class\_label_i}{\sum_{i=1}^{k} weight_i}$$,for class_label {'correct' : 1, 'wrong': 0}

1.5. Numeric Prediction

For the nearest k points, calculate the mean of their price (The information could be found in Y_train) and regard this mean as the prediction result of this test.

For w-NN, calculate the weights by the formula given in the 'METHODS' part, add them to evaluate the prediction class label. (By the following formula) If the distance is 0, the weight is

$$\frac{\sum_{i=1}^{k} weight_i * price_i}{\sum_{i=1}^{k} weight_i}$$

set to 1/0.0000000000000001.

1.6. Accuracy Calculation

Different methods are used in classification and numeric prediction respectively.

a) Classification

Count the number of 'correct' for each test (denoted as 'correct_num), then the average error

$$1-\frac{correct\_num}{total\_test\_num}$$

rate for each test is given by

Finally, find the best case (i.e. the error is minimum) among all tests to evaluate the performance for different k values

b) Numeric Prediction

Calculate the Percent Average Deviation Error for each test, for which the formula is

$$\frac{abs(Predicted\_price - real\_price)}{real\_price}$$

Finally, calculate the mean of the error rate for all tests to evaluate the performance for different k values

2.Testing Performances of k-NN & w-NN by Bayes Error Rate

2.1. Construct a probabilistic two-class classification target function

First, initialize a class for target function that includes the mean, covariance, number of features and probability. The mean is set up randomly by giving it an arbitrary value ranged between 0 and 1, the covariance matrix is constructed in the same way. The way to implement the target function is mentioned in Method, point 6.

2.2. Generate a new dataset

For each point in the dataset, it is uniformly distributed around the mean. The function takes the mean, covariance, class label (either "g" or "b") and the number of features (same as ionosphere.arff) as its inputs and it produces a new dataset that combines a set of instances and a set of class labels, for later use.

2.3. Calculate the Bayes error rate

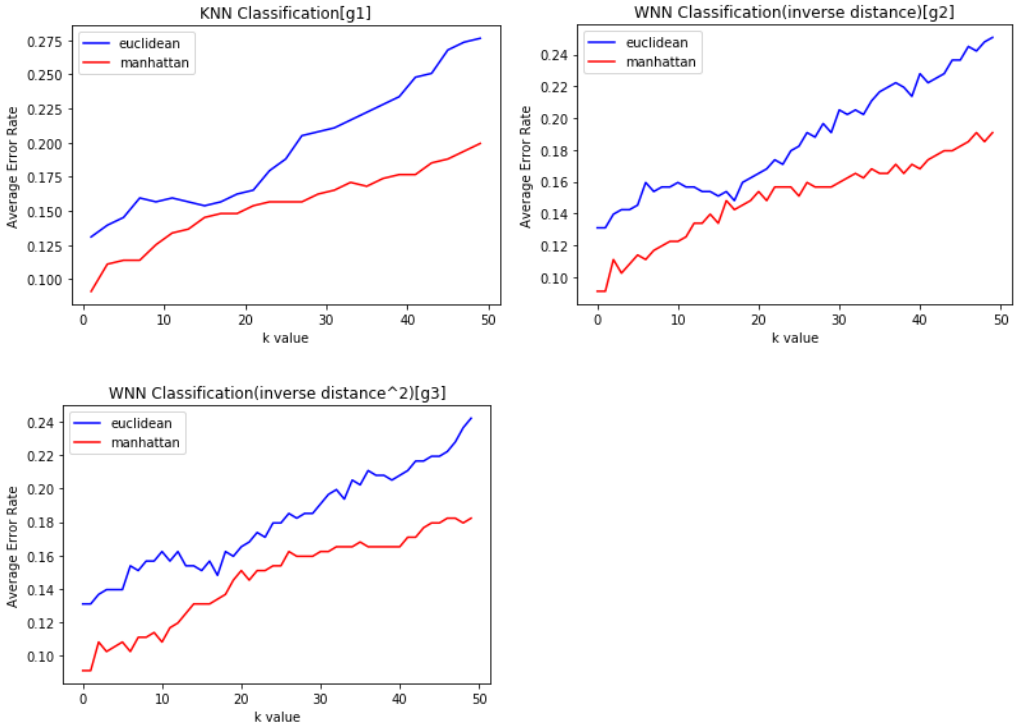
Based on the dataset generated before, the first step is to split the entire dataset into two parts: a subset of instances X and a subset of class Y, the python function is going to iteratively calculate the Bayes error rate for each target function of a single instances x_i and sums all the errors together as output.

2.4. Test the implementation of KNN and WNN on the new dataset

Taking the new dataset to the cross-validation function and test the performance on different distance measures and varies the value of K by comparing the Bayes error rate with the outputs produced by the classifier.

Results

1.Classification



a) For all the situations in 3 graphs, the lowest error rates appear when k =1.

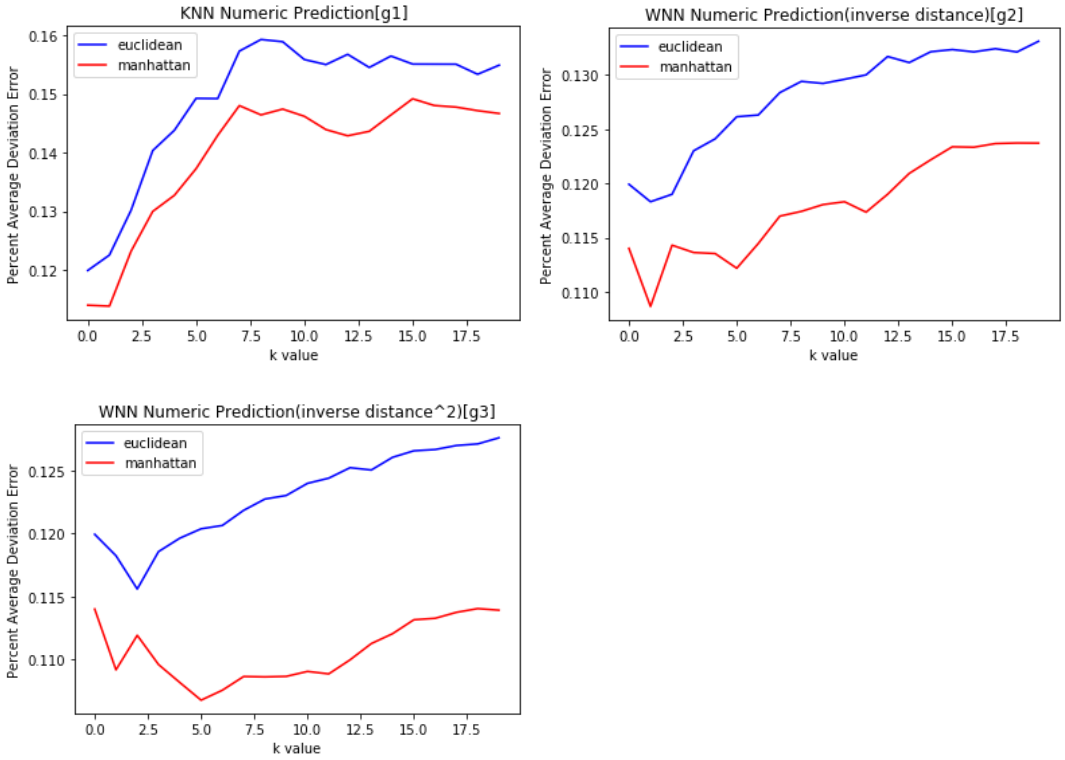
b) Comparing g2 and g3, the error rate of sets using Manhattan distance grows slower if we use the formula given in ’METHODS 5’ (inverse distance squared) to calculate weights. However, this difference is not clear for the sets using Euclidean distance.

c)Comparing the 3 graphs, the error rates of sets using w-NN is always lower than the ones using k-NN.

d) In the 3 graphs, The error rate of sets using Euclidean distance is always larger than the ones using Manhattan distance

e)In the 3 graphs, the polylines with different colors are becoming closer as k value increases until around 20, the closest point. Then, the distance between them increases

2.Numeric Prediction



a) For the 4 situations in g1 and g2, the lowest error rates appear when k = 1. For g3, the lowest error rate for sets using Euclidean distance appears when k = 2 while it for sets using Manhattan distance appears when k = 5

b) In the 3 graphs, the error rate of sets using Euclidean distance is always larger than the ones using Manhattan distance

c) Comparing g2 and g3, the error rates of sets grow slower if we use the formula given in 'METHODS 5' (inverse distance squared) to calculate weights.

d)Comparing the error rates at all k values in 3 graphs, the error rates are always lowest in g3, highest in g1. That is:

Error (w-NN(inverse distance^2))<Error(w-NN(inverse distance))<Error(k-NN)

3.Test K-NN and W-NN on the new dataset

Testing the dataset by using k-NN

```
In [101]: cross_validation_prediction(X, y, 1, 'classify', 'knn', 'manhattan')
Out[101]: 0.19658119658119655

In [102]: cross_validation_prediction(X, y, 1, 'classify', 'knn', 'euclidean')
Out[102]: 0.18518518518518523

In [103]: cross_validation_prediction(X, y, 9, 'classify', 'knn', 'manhattan')
Out[103]: 0.21082621082621078

In [104]: cross_validation_prediction(X, y, 9, 'classify', 'knn', 'euclidean')
Out[104]: 0.17948717948717952

In [105]: cross_validation_prediction(X, y, 27, 'classify', 'knn', 'manhattan')
Out[105]: 0.33618233618233617

In [106]: cross_validation_prediction(X, y, 27, 'classify', 'knn', 'euclidean')
Out[106]: 0.27350427350427353

In [107]: cross_validation_prediction(X, y, 81, 'classify', 'knn', 'manhattan')
Out[107]: 0.44729344729344733

In [108]: cross_validation_prediction(X, y, 81, 'classify', 'knn', 'euclidean')
Out[108]: 0.4330484330484331
```

Testing the dataset by using w-NN

```
In [109]: cross_validation_prediction(X, y, 1, 'classify', 'wnn', 'manhattan')
Out[109]: 0.19658119658119655

In [110]: cross_validation_prediction(X, y, 1, 'classify', 'wnn', 'euclidean')
Out[110]: 0.18518518518518523

In [111]: cross_validation_prediction(X, y, 9, 'classify', 'wnn', 'manhattan')
Out[111]: 0.19373219373219375

In [112]: cross_validation_prediction(X, y, 9, 'classify', 'wnn', 'euclidean')
Out[112]: 0.1737891737891738

In [113]: cross_validation_prediction(X, y, 27, 'classify', 'wnn', 'manhattan')
Out[113]: 0.27350427350427353

In [114]: cross_validation_prediction(X, y, 27, 'classify', 'wnn', 'euclidean')
Out[114]: 0.2136752136752137

In [115]: cross_validation_prediction(X, y, 81, 'classify', 'wnn', 'manhattan')
Out[115]: 0.33618233618233617

In [116]: cross_validation_prediction(X, y, 81, 'classify', 'wnn', 'euclidean')
Out[116]: 0.301994301994302
```

The above two charts indicate that both classifiers produce the same error rate, no matter what the distance measurements are when the k is set to be 1. However, as the k increases, the classifier by using “Euclidean” to compute the distance between objects performs slightly better than “Manhattan”. By comparing K-NN and W-NN, W-NN classifies instances more accurate than K-NN, since it weights nearer neighbors more heavily, and relatively reduces the error of high-dimensional feature space.

Conclusion

This study was completed to determine how k-NN and w-NN perform differently by comparing the accuracy on a variety of cross-validation tests. This study further confirms that these two algorithms are easy to implement but are powerful in doing statistical prediction. In most situations, w-NN performs better than k-NN, since, to some extent, it slightly decreases the influence of noise points. When the number of attributes becomes larger, it is more clear since the w-NN could avoid the curse of dimentionality but k-NN cannot.

The main issue faced by both w-NN and k-NN is the selection of k value. A bad k value may cause an extremely high error rate. Additionally, the way to calculate weights in w-NN is also considerable, it influences the importances of k nearest points.

Further related works

K-NN and W-NN algorithm is often used to find similarity between objects, a wide variety of applications can be found in numeric prediction and classification.

1.Medical application

By learning the previous case reports, the learner can predict whether a patient will have a heart attack. The training process can be based on learning the patients' diet, living habits, blood pressure, living environment, clinical variables, and other factors. (ref.1)

2.Finance

Same as other computational algorithms for searching similar patterns and statistical prediction techniques, KNN and WNN are widely applied in the field of Finance, examples are(ref.1):

- predicting the future price of a stock
- loan management
- managing investment strategies
- identifying credit risk or investment risk.

Reference

Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background -research article

Fukunaga, Keinosuke (1990) *Introduction to Statistical Pattern Recognition* by ISBN 0122698517 pages 3 and 97

Imandoust, S.B. & Bolandraftar, Mohammad. (2013). Application of K-nearest neighbor (KNN) approach for predicting economic events theoretical background. Int J Eng Res Appl. 3. 605-610.

K. Tumer, K. (1996) "Estimating the Bayes error rate through classifier combining" in *Proceedings of the 13th International Conference on Pattern Recognition*, Volume 2, 695–699