kNN algorithm

Thanh

October 17, 2019

Detailed presentation of all the contents presented in Math6350 class during the lectures on kNN automatic classification

Goal: Automatically assign a class of cases to each cases

Number of classes is $N \rightarrow$ Classes : $C_1, C_2, ..., C_N$, C_j = class of group N cases

Overall, KNN automatic classification is a SUPERVISED machine learning model that try to predict output with input value base on how the output model's neighbor is classified.

Example : x describe the current state of engine - Blackbox

IBM $\rightarrow$ Peugeot or VW $\rightarrow$ analyze the data $\rightarrow$ Software in England to diagnosis and feedback

Preliminary treatment of the data set

Divide the data set to Training set(N cases) and Test set(M cases). Each case $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,p})$, p features

Training set

Case 1	Case 2	...	Case N
x_1	x_2	...	x_N

y_i is name of class

Example 3 classes Good|OK|Bad ~ $C_1|C_2|C_3$

y_1 = 3="Good" y_N = 1="Bad"

Machine learning

Build a program = software = algorithms scan the training set $\rightarrow$ Generate a smart decision making software

Algorithm for KNN method

First we fix the k value we assume k = 9 and set the new data to be $z = (z_1, z_2, ..., z_p)$

Then to find the nearest k neighbors we calculate the Euclidean distance. According to Euclidean distance formula, a distance between two points in a p dimensions which is the number of the features is given by

$dist_i(z) = \sqrt{(z_1 - x_{i,1})^2 + (z_2 - x_{i,2})^2 + ... + (z_p - x_{i,p})^2}$ where i = 1,2,...,N

After we have calculated all the Euclidean distance of the unknown data point from all the points in Training set, we can sort all $dist_i(z)$ where i = 1,2,...,N from smallest to the largest e.g. $dist_3(z) < dist_7(z) < dist_{121}(z) < dist_{67}(z) < dist_{54}(z) < dist_{18}(z) < dist_{21}(z) < dist_{220}(z) < dist_{196}(z) < ....$ After all, we can find the 9 smallest $dist_i(z)$ to be the 9 nearest neighbors for z.

Hence we can see how many points of the 9 nearest neighbors fall into each class which is CL1, CL2 and CL3. Then z belongs to the class that get the largest amount of points.

Evaluate model at the end

we can evaluate the model performance of our KNN automatic classification by applying the confusion matrix

		Predict Values		
		CL1	CL2	CL3
		—	—	—
	CL1	a_{11}	a_{12}	a_{13}
Actual values	CL2	a_{21}	a_{22}	a_{23}
	CL3	a_{31}	a_{32}	a_{33}

The value of a_{11} is the amount of number that we predict the data is in CL1 and the actual value is in CL1 so as a_{22} and a_{33}. The acurracy of the acurracy rate will be $\frac{a_{11}+a_{22}+a_{33}}{a11+...+a33}$ which the dinominator is the size of the test set.

How to choose K?

The performance of the model could be affected by the value of k or the size of the test set.Therefore, if we fix the test set, we can decide the best value of k by computing the accuracy rate.

Weighted distance for KNN Algorithm

Additionally, we can do weighted distance to the Eucludean space we applied to find the k nearest neighbors as following

$Dist_i(z) = \sqrt{w_1(z_1 - x_{i,1})^2 + w_2(z_2 - x_{i,2})^2 + ... + w_p(z_p - x_{i,p})^2}$ where weight $w_1 + w_2 + ... + w_p = 1$

So how do we discover the weight? We select one of the features says the i^{th} feature where $i = 1, ..., p$ then we discard all other features. Then we do KNN algorithm with the only i^{th} feature and compute the performance under this condition. Eventually, we get p performances then we can denote that

$w_i = \frac{performance_i}{performance_1+...performance_p}$

Pretreatment for KNN

The main reason to do retreatment for KNN algorithm is that we want to transform qualitative data to quantitative data. Moreover, by the retreatment for KNN algorithm, we can also transform discrete features to continuous features and discrete characteristics to continuous characteristics.

Let's take a socialogical (medical) data for example. Assume that one of the features in the original data set is the profession $pro = (pro_1, ..., pro_{23})$. Then we make this feature transform to 23 numerical features, each of them are unit binary vector of length 23

i.e. $pro_2 = [0, 1, 0, 0, 0, ..., 0], pro_4 = [0, 0, 0, 1, 0, ..., 0]$, $||pro_2 - pro_4||^2 = 2 \rightarrow ||pro_2 - pro_4|| = \sqrt{2}$

Other features in the original data set:

$country = (cty_1, ..., cty_{50})$

$race = (race_1, ..., race_5)$

$TypeOfDegree = (tod_1, ..., tod_{15})$

$age = (5, 11, ..., 74)$

$SAT = (404, 536, ..., 1193)$

By using the same idea to all other nonnumrical features in the original data set which is the top three features above, we can get $23 + 50 + 5 + 15 = 93$ new features. For feature age and SAT they are already numerical features so we don't tranform they to the new fetures. Therefore, the final new numerical feature will be $93+2=95$ features. Eventually, we can do KNN algorithm to this data by it's new features $x=(x_1,...,x_{95})$.