Data Mining Homework 3

Jennifer Probst

11 November 2020

Exercise 1: k-NN

Value-of-k	Accuracy	Precision	Recall
1	0.81	0.81	0.93
2	0.81	0.81	0.93
3	0.71	0.79	0.79
4	0.71	0.79	0.79
5	0.76	0.85	0.79
6	0.76	0.85	0.79
7	0.76	0.8	0.86
8	0.76	0.8	0.86
9	0.71	0.79	0.79
10	0.71	0.79	0.79

b) As we learned in the lecture, simply testing different values for k leads to overfitting of this parameter. A better approach would be cross-validation, where the dataset is split into n subsets. In n different training rounds, n-1 subsets are then used for training and one for testing. With this strategy we can find an optimal k, namely the one with the n best results from the training according to a chosen metric.

c) Assume we have data already split in train and test set, we use a x-fold crossvalidation to find the best k out of i possible ks. Then we predict x times values for the test dataset for all possible ks, which is done in O(x*i*prediction-runtime). From the lecture we know that the runtime of computing the distance to all n neighbours is O(n+n*log n). So the overall complexity of the algorithm is O(x*i*(n+n*log n). The space complexity is O(x*i+n*m), where for x folds i different scores for possible k values and n data points with m features are stored.

d) I don't think that there is a difference for the prediction complexity, because in linear time we can count how many nearest neighbors belong to each class and then also in linear time select the most common one.

e) Yes, other metrics work for the k-NN algorithm as well as for semimetrics as DTW. The selection of such a semimetric might be very useful e.g. for time series data as used in homework 2.

f) Yes, this is also possible. We can compute the k-NN of this x' with some chosen distance metric. If we then use the weighted average of the y of the k-NN, we get a prediction for y'. For the weight of y_i in the weighted average, the inverse of the distance $xi - x'$ could be used.

Exercise 2: Naive Bayes

a) Here the output probabilities of class 2 (benign) can be seen:

Value	clump	uniformity	marginal	mitoses
1	0.315	0.836	0.821	0.970
2	0.103	0.081	0.080	0.019
3	0.209	0.053	0.067	0.005
4	0.145	0.019	0.011	0.000
5	0.182	0.000	0.007	0.002
6	0.034	0.005	0.009	0.000
7	0.002	0.002	0.000	0.002
8	0.009	0.002	0.000	0.002
9	0.000	0.002	0.002	0.000
10	0.000	0.000	0.002	0.000

Here the output probabilities of class 4 (malignant) can be seen:

Value	clump	uniformity	marginal	mitoses
1	0.013	0.018	0.137	0.574
2	0.013	0.037	0.091	0.112
3	0.054	0.115	0.105	0.126
4	0.049	0.138	0.119	0.049
5	0.188	0.124	0.087	0.018
6	0.067	0.106	0.078	0.013
7	0.090	0.069	0.050	0.027
8	0.166	0.115	0.105	0.031
9	0.063	0.018	0.014	0.000
10	0.296	0.258	0.215	0.049

To predict the class of [clump = 6, uniformity = 2, marginal = 2, mitoses = 1] with the Naive Bayes Classifier, we calculate $argmax_{y_i} P(Y = y|X = (6,2,2,1) = max(P(Y = 2|X = (6,2,2,1), P(Y = 4|X = (6,2,2,1))$. From Bayes Theorem we know that

$$P(Y = y|(6,2,2,1)) = \frac{P((6,2,2,1)|Y = y) * P(Y = y)}{P((6,2,2,1))}$$

We assume that P(X=x) is the same for all classes, so this can be neglected. P(Y=2)=0.663 and P(Y=4)=0.337. Furthermore, P((6,2,2,1)|Y=y) can be disassembled to $\prod_{n=1}^{4} P(X_i = x_i|Y = y$. So $P(Y = 2|X = (6,2,2,1) = P((6,2,2,1)|Y = 2) * P(Y = 2) = (0.034 * 0.081 * 0.080 * 0.970) * 0.663 = 1.42 * 10^{-4}$. In a similar manner we obtain $(P(Y = 4|X = (6,2,2,1) = 4.36 * 10^{-5}$. Therefore we predict the classlabel to be 2 (benign), as max(P(Y=2|X=(6,2,2,1),P(Y=4|X=(6,2,2,1)) = P(Y=2|X=(6,2,2,1).

b) The probabilities of figure 4 will change depending on how missing values are handled. In our case there were some missing values for features. Calculating the probabilities of feature values, missing values per feature were simply ignored in the counting process and the counts were then divided by the number of non missing points for this feature. Alternatively one could ignore entire datapoints which contain at least one missing value or try to estimate the missing value. Both alternative strategies will alter the probabilities computed.

c) In order to not end up with the whole probability being 0 (if there is one 'zero-frequency' in the multiplication process of the individual probabilites), we can start our counting at 1 instead from 0. Doing this the product will not be 0 and still depend on the probabilities of other features. We therefore don't loose the information of the other feature probabilities.

Exercise 3: Bayes' Theorem

a) We have to calculate the probability to have selected bowl 1 given we picked a vanilla brownie. This we denote as $P(B1|vanilla)$. From the Bayes Theorem we know that

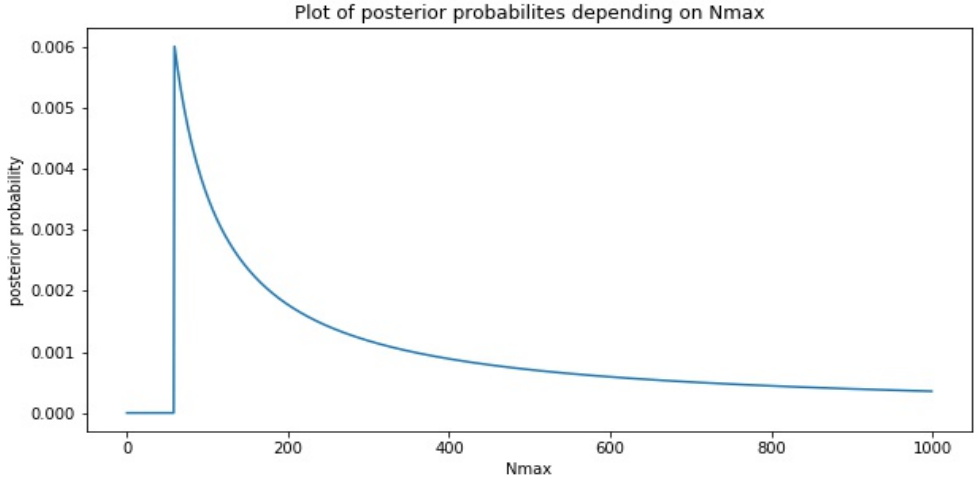
$$P(B1|vanilla) = \frac{P(vanilla|B1) * P(B1)}{P(vanilla)}$$

Because we choose one bowl at random, we get $P(B1) = 1/2$. $P(vanilla|B1)$ can easily be computed as in bowl one there are 30 vanilla brownies out of 40 total brownies, so $P(vanilla|B1) = 3/4$. To calculate $P(vanilla)$ we sum up the probabilities of getting a vanilla brownie in both bowl, hence $P(vanilla) = P(vanilla|B1)*P(B1)+P(vanilla|B2)*P(B2) = 3/4*1/2+1/2*1/2 = 5/8$. We used $P(B2) = P(B1)$ and $P(vanilla|B2) = 2/4 = 1/2$, with the same reasoning as for $P(vanilla|B1)$. Now we can compute

$$P(B1|vanilla) = \frac{P(vanilla|B1) * P(B1)}{P(vanilla)} = \frac{3/4 * 1/2}{5/8} = 3/5$$

.

b)



The python code and the graph of plotted posterior probabilities show highest probability for N=60 (P(N=60|D=60)=0.005843), which is the lowest N possible. This makes sense as the likelihood further decreases for higher N, making the whole posterior probability smaller.