tek5020 project 1

Anders Bråte
Institue for Physics
University of Oslo
Norway
November 26, 2021

In this project we have been following the workflow of a made up classification problem, which involved looking at three different datasets (1,2 and 3) and applying classification methods on these. 1 and 2 were synthetic datasets, meaning they were drawn from a known probability distribution, whilst the third was made using silhouettes from different car models. dataset number 1 and 4 were binary with four properties, whilst the second was binary with three properties.
As with most classification we split our dataset into train and test, where as the names imply the classification algorithms were tweaked using the train dataset, and tested by applying the same models on data the model had not seen yet. This is to gauge to what degree the algorithms work on other, similar data. In our case the training data consisted of the datapoints with odd numbered indexes, and the test data with even.

Initially we applied the well known nearest neighbours method, to find the best combination of properties. This means we must loop over all combinations, for all three datasets, then compare the best combination to the other classifiers. Once we have our best combination, we apply the minimum error rate method and the least squares method and compare these.
To give an idea of the distribution of our data we can plot the second dataset fairly easily in 3d, as shown below.

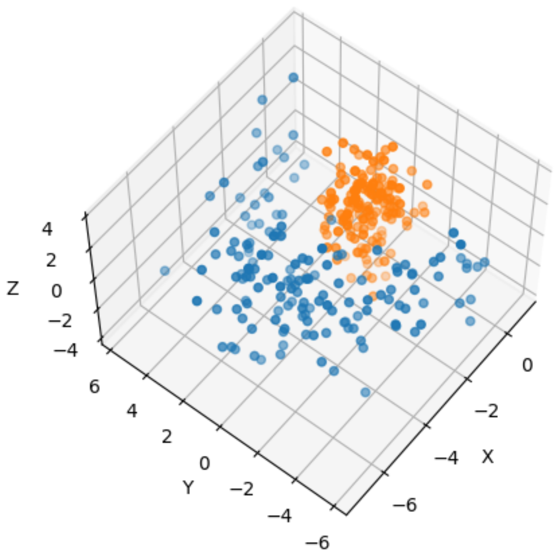

Figure 1. The second dataset visualized in 3d. This dataset only has three properties, labeled as x,y and z in the figure.

Table I. The rate of error for dimension one, with single features, not combinations.

dimention 1	1	2	3	4
dataset 1	0.24	0.36	0.4333	0.3867
dataset 2	0.18	0.28	0.4933	
dataset 3	0.33	0.31	0.345	0.395

Table II. The rate of error for different combination of features, in the second dimension.

dimention 2	12	13	14	23	24	34
dataset 1	0.18	0.1933	0.1667	0.32	0.2267	0.3
dataset 2	0.0133	0.1933		0.287		
dataset 3	0.215	0.17	0.285	0.095	0.24	0.19

Table III. The rate of error for different combinations of features in the third dimension. It now becomes very clear that dataset 2 only has three properties

dimension 3	123	124	134	234
dataset 1	0.1467	0.1	0.1267	0.2133
dataset 2	0.02			
dataset 3	0.1	0.2	0.15	0.075

Table IV. The rate of error for the combination of all four features. the second dataset is left out as it has only three sets of features

dimension 4	1234
dataset 1	0.0933
dataset 3	0.095

results

Tasks

1, Hvorfor er det fornuftig å benytte nærmeste-nabo klassifikatoren til å finne gunstige egenskapskombinasjoner?

The nearest neighbour method has some handy properties, which we also take advantage of in our implementation. We can find which properties separates the classes the best, like we do when we find the best combination of data. This is especially useful if a dataset has a large amount of properties.
The nearest neighbours method is also one which doesn't need any explicit training, meaning this phase is rather quick. It does however have a drawback in that the testing phase requires a run-through of the whole dataset.
The algorithm is also a so called non-parametric, meaning it makes no assumptions in advance regarding the datset. This means it can perform better on real world data, which often doesn't have a clear underlying distribution.
unlike methods such as random forests or ensemble methods, this method doesn't do well with skewed data. If one class appears much more frequent that another, the algorithm could have a hard time learning the correct patterns.

Table V. The best combinations of features per dimension and dataset.

	dataset 1	dataset 2	dataset 3
dimension 1	[0]	[0]	[1]
dimension 2	[0,3]	[0,1]	[1,2]
dimension 3	[0,1,3]	[0,1,2]	[1,2,3]
dimension 4	[0,1,2,3]		[0,1,2,3]

Table VI. Error rate for the two different methods, for the given combinations, for the 1st dataset

features	nearest neighbour	minimum error	least squares
0	0.24	0.1867	0.1867
0,3	0.1667	0.1133	0.1133
0,1,3	0.1	0.1	0.0933
0,1,2,3	0.0933	0.08	0.0733

Table VII. Error rate for the two different methods, for the given combinations, for the 2nd dataset

features	nearest neighbour	minimum error	least squares
0	0.18	0.1067	0.1067
0,1	0.0133	0.02	0.12
0,1,2	0.02	0.02	0.12

2, Hvorfor kan det i en praktisk anvendelse være fornuftig å finne en lineær eller kvadratisk klassifikator til erstatning for nærmeste-nabo klassifikatoren?

The drawbacks of using a nearest neighbour method was to some degree explored in the previous section, but we can reiterate some of the more important ones. The scaling of the nearest neighbour method can be tricky, if you are lucky enough to have a very large dataset, and especially if working with limited resources in terms of storage. The method has to iterate over each other point for a given point in the whole dataset. Meaning this method doesn't scale very well with big datasets.
The nearest neighbour method is however well versed with real life data, since it does not assume.

3, Hvorfor er det lite gunstig å bruke samme datasettet både til trening og evaluering av en klassifikator?

It is very common in all of classification to split the dataset into train and test (and sometimes validation). Using the same dataset on both would lead to something commonly known as overfitting. This in practice means the dataset will not perform well on other data, even if it is of the same type and from the same source. There is naturally some midpoint, as you want the algorithm to perform as well as possible.

Table VIII. Error rate for the two different methods, for the given combinations, for the 3rd dataset

features	nearest neighbour	minimum error	least squares
1	0.31	0.225	0.335
1,2	0.095	0.2	0.2
1,2,3	0.075	0.13	0.16
0,1,2,3	0.095	0.07	0.12

1. 4, Hvorfor gir en lineær klassifikator dårlige resultater for datasett 2?

As we saw in figure 1, the second dataset has a doughnout type shape. As one can imagine, this does not bode well for a linear classifier, and this is exactly what we see from our results. Using only the second feature in the first dimension yields a result of 0.18. Adding a second feature however means this drops to 0.013, nearly one tenth.