Machine Learning
Assignment 1.1

Nikolaj Dybdahl Rathcke (rfq695)

November 23, 2015

1 Classification

1.1 Nearest Neighbour

The source code can be found in the source folder, name `k-NN.py`. It is quite a general implementation as it can take a distance metric function as argument (which in my case has been euclidian distance in a plane). It loads data in depending om some `loadData` function. It also calls an auxillary function to get the classification which also has to be implemented by yourself. The function returns the classification of all element in a test set - so it is not called with a single element.

The result from running with $k = 1$, $k = 3$ and $k = 5$ can be seen below. The i'th element in the list is the outputted classification of the i'th element from the file (using 0-indexing). The float below is the classification error (wrongly_classified / total_number_of_tests).

```
k = 1
[0.0, 2.0, 1.0, 2.0, 2.0, 1.0, 2.0, 2.0, 0.0, 1.0, 0.0, 0.0, 2.0, 1.0, 0.0, 2.0, 0.0, 1.0,
 0.0, 0.0, 1.0, 0.0, 0.0, 2.0, 1.0, 2.0, 0.0, 2.0, 1.0, 0.0, 0.0, 2.0, 0.0, 2.0, 1.0, 0.0,
 2.0, 2.0]
0.184210526316

k = 3
[0.0, 2.0, 2.0, 2.0, 2.0, 1.0, 2.0, 2.0, 1.0, 1.0, 0.0, 0.0, 2.0, 1.0, 0.0, 2.0, 0.0, 1.0,
 0.0, 0.0, 1.0, 0.0, 0.0, 2.0, 1.0, 2.0, 0.0, 2.0, 1.0, 0.0, 0.0, 2.0, 0.0, 2.0, 1.0, 0.0,
 2.0, 2.0]
0.184210526316

k = 5
[0.0, 2.0, 1.0, 2.0, 2.0, 1.0, 2.0, 2.0, 1.0, 2.0, 1.0, 0.0, 2.0, 1.0, 0.0, 2.0, 0.0, 1.0,
 0.0, 0.0, 1.0, 1.0, 0.0, 2.0, 1.0, 2.0, 0.0, 2.0, 1.0, 0.0, 0.0, 1.0, 0.0, 2.0, 1.0, 0.0,
 2.0, 2.0]
0.315789473684
```

So it classifies wrong 18% of the time when using $k = 1$ or $k = 3$. When increasing k, it gets even worse, so it shows that it is not nessecarily better to increase k. Another thing to note is that the x and y coordinates (length and width) have a difference of more than a factor 10. Which means length (the larger value) actually weighs more when we use euclidian distance.

1.2 Hyperparameter selection using cross-validation

The function `knn_best` has been implemented in `knn.py`. It takes a training set and a parameter k, the k-fold cross validation, it then applies the `knn` algorithm with parameter $k = \{1, 2, .., 25\}$ and returns the one that produced a best average classification error and what the average was.
Running it on the test set produced $k = 3$ with an average of 0.2 in error (The same result was obtained with $k = 4$). Since we already did $k = 3$, we run `knn` with $k = 4$ and get:

```
[0.0, 2.0, 1.0, 2.0, 2.0, 1.0, 2.0, 2.0, 1.0, 1.0, 2.0, 0.0, 2.0, 1.0, 0.0, 2.0, 0.0, 1.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 2.0, 1.0, 2.0, 0.0, 2.0, 1.0, 0.0, 0.0, 1.0, 0.0, 2.0, 1.0, 0.0,
```

```
 2.0, 2.0]
0.236842105263
```

Which is actually worse than what we got with $k = 3$.

1.3 Data Normalization

A function to find mean and variance has been implemented and `fNorm` uses this to normalize the data set. The output from calling the function that finds mean and variance on the training set is:

```
[[5.756000000000003, 0.3017000000000001], [0.6888639999999999, 0.0017421099999999998]]
```

So the first feature has a mean of 5.756 and a variance of 0.6889. The second feature has a mean of 0.3017 and a variance of 0.0017.
If we normalize the data set, we find the mean and variance to be:

```
[[-3.468336728928989e-15, -1.8174350913113814e-15], [1.0000000000000007, 0.9999999999999997]]
```

Where both means are very close to 0 and the both variances are very close to 1 as it should be.
We now find the number of neighbours by the function `knn_best`:

```
(1, 0.13999999999999999)
```

which suggests $k = 1$ with error 0.14. The same error is obtained when $k = 11$ but it only outputs one.
Now we use the nearest neighbour algorithm on the normalized data and get:

```
k = 1
[0.0, 2.0, 2.0, 1.0, 2.0, 1.0, 2.0, 2.0, 0.0, 1.0, 0.0, 0.0, 2.0, 2.0, 0.0, 2.0, 0.0, 1.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 2.0, 1.0, 1.0, 0.0, 2.0, 1.0, 0.0, 0.0, 1.0, 0.0, 2.0, 2.0, 0.0,
 2.0, 2.0]
0.210526315789

k = 3
[0.0, 2.0, 2.0, 2.0, 2.0, 1.0, 2.0, 2.0, 0.0, 1.0, 0.0, 0.0, 2.0, 2.0, 0.0, 2.0, 0.0, 1.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 2.0, 1.0, 1.0, 0.0, 2.0, 1.0, 0.0, 0.0, 2.0, 0.0, 2.0, 2.0, 0.0,
 1.0, 2.0]
0.184210526316

k = 5
[0.0, 2.0, 1.0, 2.0, 2.0, 1.0, 2.0, 2.0, 0.0, 2.0, 0.0, 0.0, 2.0, 2.0, 0.0, 2.0, 0.0, 1.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 2.0, 1.0, 2.0, 0.0, 2.0, 1.0, 0.0, 0.0, 1.0, 0.0, 2.0, 2.0, 0.0,
 2.0, 2.0]
0.263157894737
```

So the error is generally higher, which means that normalized data actually performs worse than raw data in this case.

Note: The tests performed are in the bottom of the `k-NN` file.

2 Probability Refreshment

2.1

The sample space is the set of all outcomes, thus when drawing two balls we can get the following sample space S:

$$\begin{aligned} S = \{&(Red, Red), (Red, Orange), (Red, Blue),\\ &(Orange, Red), (Orange, Orange), (Orange, Blue),\\ &(Blue, Red), (Blue, Orange)\} \end{aligned}$$

This is assuming the balls that have the same color are indistinguishable but we can distinguish between the drawn balls. Since it is without replacement and only one blue ball exists, the event $(Blue, Blue)$ is not possible.

2.2

The following table shows the probability of each event and how it was calculated (there are 9 balls total).

Event	Calculation	Probability
(Red, Red)	$5/9 \cdot 4/8$	$20/72$
$(Red, Orange)$	$5/9 \cdot 3/8$	$15/72$
$(Red, Blue)$	$5/9 \cdot 1/8$	$5/72$
$(Orange, Red)$	$3/9 \cdot 5/8$	$15/72$
$(Orange, Orange)$	$3/9 \cdot 2/8$	$6/72$
$(Orange, Blue)$	$3/9 \cdot 1/8$	$3/72$
$(Blue, Red)$	$1/9 \cdot 5/8$	$5/72$
$(Blue, Orange)$	$1/9 \cdot 3/8$	$3/72$

Since we used a common denominator, we can easily verify that the sum of all probabilities is indeed 1 as it should be.

2.3

Since the number of orange balls are 3, but we only draw 2, the random variable X can take values in $\{0, 1, 2\}$.

2.4

The value $\mathbb{P}\{X = 0\}$ is equal to the sum of all probabilities in the events where there are no orange balls, that is

$$\mathbb{P}\{X = 0\} = 20/72 + 5/72 + 5/72 = 30/72$$

Which can be reduced to $5/12$.

2.5

The value $\mathbb{E}[X]$ is equal to the probability it takes each value times the value summed together. Then we divide by 3 as there are 3 options to find the expected value.
We find

$$\mathbb{P}\{X = 1\} = 15/72 + 15/72 + 3/72 + 3/72 = 36/72$$
$$\mathbb{P}\{X = 2\} = 6/72$$

And we can find the expected value of X.

$$\begin{aligned}\mathbb{E}[X] &= 30/72 \cdot 0 + 36/72 \cdot 1 + 6/72 \cdot 2 \\ &= 48/72 = 2/3\end{aligned}$$

Which means we can expect that there is $2/3$ of an orange ball when we draw 2 balls at random.

3 Probability Refreshment 2

3.1

We want to prove that

$$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y] \tag{1}$$

We know that from definition that

$$\mathbb{E}[X] = \sum_{x \in \mathcal{X}} x p_X(x)$$

This means we can write

$$\begin{aligned}\mathbb{E}[X+Y] &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}(x+y)p_{XY}(x,y)\\ &= \sum_{x\in\mathcal{X}} x\sum_{y\in\mathcal{Y}} p_{XY}(x,y) + \sum_{y\in\mathcal{Y}} y\sum_{\in\mathcal{X}} p_{XY}(x,y)\\ &= \sum_{x\in\mathcal{X}} xp(x) + \sum_{y\in\mathcal{Y}} yp(y)\\ &= \mathbb{E}[X]+\mathbb{E}[Y]\end{aligned}$$

Which means we have proven equation 1.

3.2

We want to show, when X and Y are independent, then

$$\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y] \tag{2}$$

We have the following definition

$$E[X] = \sum_{x\in\mathcal{X}} xp_X(x)$$

This means we can write the following

$$\begin{aligned}\mathbb{E}[XY] &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} xyp_{XY}(x,y)\\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} xyp_X(x)p_Y(y) && \text{(Using indepedence assumption)}\\ &= \sum_{x\in\mathcal{X}} xp_X(x)\sum_{y\in\mathcal{Y}} yp_Y(y)\\ &= \mathbb{E}[X]\mathbb{E}[Y]\end{aligned}$$

which proves equation 2.

3.3

As an example, let X and Y be two cards drawn from the same standard deck of cards (without replacement).
In this example, for both X and Y then $\mathbb{E}\{\text{card is red}\} = \mathbb{E}\{\text{card is black}\} = \frac{1}{2}$.
The joint distribution of X and Y is

X\Y	R	B
R	25/102	26/102
B	26/102	25/102

We get that the expected values as

$$\begin{aligned}&\mathbb{E}[X=R, Y=R] = 25/102\\ &\mathbb{E}[X=R]\mathbb{E}[Y=R] = 1/2\cdot 1/2 = 1/4\end{aligned}$$

This means that

$$\mathbb{E}[XY] \neq \mathbb{E}[X]\mathbb{E}[Y]$$

because the random variables are not independent of eachother.

3.4

We want to prove

$$\mathbb{E}[\mathbb{E}[X]] = \mathbb{E}[X] \tag{3}$$

We call the expected value a random variable for the constant c as it will always evaluate to a real number - so $\mathbb{E}[X] = c$. An expected value of a constant is obviously the constant itself, so $\mathbb{E}[c] = c$, thus we get

$$\begin{aligned}\mathbb{E}[\mathbb{E}[X]] &= \mathbb{E}[c] \\ &= c \\ &= \mathbb{E}[X]\end{aligned}$$

which proves equation 3.

3.5

We want to prove

$$\mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 \tag{4}$$

We can now derive the following:

$$\begin{aligned}\mathbb{E}\left[(X - \mathbb{E}[X])^2\right] &= \mathbb{E}\left[X^2 - 2X\mathbb{E}[X] + (\mathbb{E}[X])^2\right] \\ &= \mathbb{E}[X^2] - \mathbb{E}[2X\mathbb{E}[X]] + (\mathbb{E}[\mathbb{E}[X]])^2 && \text{(Using equation 1)} \\ &= \mathbb{E}[X^2] - \mathbb{E}[2X]\mathbb{E}[\mathbb{E}[X]] + (\mathbb{E}[\mathbb{E}[X]])^2 && \text{(Using equation 2)} \\ &= \mathbb{E}[X^2] - 2\mathbb{E}[X]\mathbb{E}[X] + (\mathbb{E}[X])^2 && \text{(Using equation 3)} \\ &= \mathbb{E}[X^2] - (\mathbb{E}[X])^2\end{aligned}$$

and we have proved equation 4.

4 Markov's inequality, Hoeffding's inequality and binomial bound

4.1

The Markov inequality:

$$\mathbb{P}\{X \geq \varepsilon\} \leq \frac{\mathbb{E}[X]}{\varepsilon}$$

We define a new random variable $S = \sum_{i=1}^{10} X_i$. We can now apply Markov's inequality to S with $\varepsilon = 9$:

$$\begin{aligned}\mathbb{P}\{S \geq 9\} &\leq \frac{\mathbb{E}[S]}{9} \\ &= \frac{5}{9} \approx 0.5555\end{aligned}$$

Since $\mathbb{E}[S] = 10 \cdot (\frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 1)$. Thus an upper bound on the probability that 9 out of 10 of the Bernouilli random variables is 1 is 5/9. This is obviously a bit larger than the actual probability on $11 \cdot \frac{1}{2}^{10} = 11/1024$.

4.2

The Hoeffding inequality:

$$\mathbb{P}\{\sum_{i=1}^{n} X_i - \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] \geq \varepsilon\} \leq e^{-2\varepsilon^2 / \sum_{i=1}^{n}(b_i - a_i)^2}$$

In our case $b_i = 1$ and $a_i = 0$ for all i and $n = 10$. We have already calculated the expected sum in the previous question, it was 5. Since we want the probability that the first sum is greater than or equal to 9, that means $\varepsilon = 3$, so we get:

$$\begin{aligned}\mathbb{P}\{\sum_{i=1}^{n} X_i - 5 > 3\} &\leq e^{-2\cdot 3^2/\sum_{i=1}^{n}(1-0)^2} \\ &= e^{-2\cdot 3^2/10} \\ &= e^{-9/5} \approx 0.1653\end{aligned}$$

Which is significantly less than the bound the Markov inequality gave.

4.3

To elaborate on the calculation done in (4.1), since each X_i takes a value with probability 1/2, we also know that each of the 2^10 combinations happens with the same probability, or 1/1024. Since the number of combinations that yields 10 is equal to the combination where all $X_i = 1$, that is just one. The number of combination that yields 9 is equal to the case where exactly one is 0, that means there are 10 of these. Thus we get $11 \cdot \frac{1}{2} \approx 0.0107$.

4.4

The Markov inequality is the weakest of the three and is actually a very sloppy bound. The Hoeffding inequality provides a much better result and is a lot closer to the "real" bound we found in (4.3).

5 Hoeffding Inequality

5.1

This does actually not require us to use the Hoeffding inequality, as an exact bound can be calculated very easily. Since the only case the flight exceeds 99 people is when there are all 100. So we can calculate the probability as:

$$\left(\frac{19}{20}\right)^{100} \approx 0.006$$

Since each person has a probability of 19/20 of not missing the flight.

The bound obtained by using Hoeffding inequality yields:

$$\begin{aligned}\mathbb{P}\{\sum_{i=1}^{n} X_i - 95 > 4\} &\leq e^{-2\cdot 4^2/\sum_{i=1}^{n}(1-0)^2} \\ &= e^{-2\cdot 4^2/100} \\ &= e^{-8/25} \approx 0.7261\end{aligned}$$

Which is a lot worse.