Sprachverarbeitung: Übung

SoSe 24

Janis Pagel
Department for Digital Humanities, University of Cologne

2024-05-28

For this exercise, you need to both submit manual calculations as well as Python code. Please submit two files in Ilias, one a PDF with your calculations and one a file containing your Python code (either Jupyter Notebook or Python script). You can also combine both files into a zip-archive and submit only the archive. You can either solve the calculations by hand on a sheet of paper, scan it and submit as a PDF file or use the capabilities to write mathematical equations of tools like MS Word / LibreOffice / LaTeX, etc. to write down your calculations digitally.

Exercise 1.

For this part of the exercise, you need to do manual calculations and submit your solution in a PDF file.

Given are the following six SMS, three spam and three not-spam (ham):

SMS text	Class
You are a winner U have been specially selected 2 receive £1000 or a 4* holiday (flights inc) speak to a live operator 2 claim 0871277810910p/min (18+)	spam
As a valued customer, I am pleased to advise you that following recent review of your Mob No. you are awarded with a å£1500 Bonus Prize, call 09066364589	spam
Message Important information for O2 user. Today is your lucky day! 2 find out why log onto http://www.urawinner.com there is a fantastic surprise awaiting you	spam
I cant pick the phone right now. Pls send a message	ham
Pls pls find out from aunt nike.	ham
Love that holiday Monday feeling even if I have to go to the dentists in an hour	ham

Table 1: SMS spam dataset

Create a table similar to the one in the lecture slides, on slide no. 17, by annotating if a certain token occurs in a specific SMS text. Only use the following three features to create the table: “holiday”, “I” and “pls”. You can ignore spelling differences, e.g. “Pls” and “pls” count as the same token. Using this table, use the Decision Tree pseudo code on lecture slide no. 14 to calculate the information gain for each feature and each subsequent data partitioning. Once the entropy for a dataset partition reaches 0, you can stop traversing that branch. Do this until you have an entropy of 0 for each leaf in the tree. You can draw the resulting tree if you’d like, but you do not need to submit it, just the calculations for obtaining the tree.

Use your resulting decision tree to classify the following three texts:

SMS text
Hmm...my uncle just informed me that he’s paying the school directly. So pls buy food.
Dear Matthew please call 09063440451 from a landline, your complimentary 4*Lux Tenerife holiday or £1000 CASH await collection. ppm150 SAE T&Cs Box334 SK38XH.
Camera quite good, 10.1mega pixels, 3optical and 5digital dooms. Have a lovely holiday, be safe and i hope you hav a good journey! Happy new year to you both! See you in a couple of weeks!

Are they spam or ham according to your tree?

Now, calculate the information gain for the feature “you” on the original dataset in table 1. What do you observe? How do you explain the result?

Solution 1.

IG for all features for full dataset:

holiday	I	pls	class
1	0	0	spam
0	1	0	spam
0	0	0	spam
0	1	1	ham
0	0	1	ham
1	1	0	ham

$$H(D)\widehat{=}H(spam\ spam\ spam\ ham\ ham\ ham) = -(\frac{3}{6} * log_2(\frac{3}{6}) + \frac{3}{6} * log_2(\frac{3}{6})) = 1$$

$$H(holiday = 1)\widehat{=}H(spam\ ham) = -(\frac{1}{2} * log_2(\frac{1}{2}) + \frac{1}{2} * log_2(\frac{1}{2})) = 1$$

$$H(holiday = 0)\widehat{=}H(spam\ spam\ ham\ ham) = -(\frac{2}{4} * log_2(\frac{2}{4}) + \frac{2}{4} * log_2(\frac{2}{4})) = 1$$

$$H(holiday) = \frac{2 * 1 + 4 * 1}{6} = 1$$

$$IG(holiday) = H(D) - H(holiday) = 1 - 1 = 0$$

$$H(I = 1)\widehat{=}H(spam\ ham\ ham) = -(\frac{1}{3}log_2(\frac{1}{3}) + \frac{2}{3}log_2(\frac{2}{3})) = 0.92$$

$$H(I = 0)\widehat{=}H(spam\ spam\ ham) = -(\frac{2}{3}log_2(\frac{2}{3}) + \frac{1}{3}log_2(\frac{1}{3})) = 0.92$$

$$H(I) = \frac{3 * 0.92 + 3 * 0.92}{6} = 0.92$$

$$IG(I) = 1 - 0.92 = 0.08$$

$$H(pls = 1)\widehat{=}H(ham\ ham) = -(\frac{2}{2}log_2(\frac{2}{2})) = 0$$

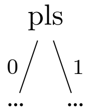
$$H(pls = 0)\widehat{=}H(spam\ spam\ spam\ ham) = -(\frac{3}{4}log_2(\frac{3}{4}) + \frac{1}{4}log_2(\frac{1}{4})) = 0.81$$

$$H(pls) = \frac{2 * 0 + 4 * 0.81}{6} = 0.54$$

$$IG(pls) = 1 - 0.54 = 0.46$$

$$IG(pls) > IG(I) > IG(holiday)$$

After calculating the Information Gain for all features for the complete dataset, the feature “pls” has the highest IG, so choose this feature as top feature for the tree:

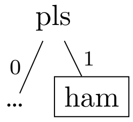


IG for features, if $pls = 1$:
Dataset partition for $pls = 1$:

holiday	I	class
0	1	ham
0	0	ham

$$H(D_{pls=1})\widehat{=}H(ham\ ham)= -(\frac{2}{2} * log_2(\frac{2}{2})) = 0$$

We are done for this path, since the entropy for splitting the dataset with $pls = 1$ is 0.
New tree:



IG for features, if $pls = 0$:
Dataset partition for $pls = 0$:

holiday	I	class
1	0	spam
0	1	spam
0	0	spam
1	1	ham

$$H(D_{pls=0}) \widehat{=} H(spam\ spam\ spam\ ham) = -(\frac{3}{4} * log_2(\frac{3}{4}) + \frac{1}{4} * log_2(\frac{1}{4})) = 0.81$$

$$H(holiday = 1) \widehat{=} H(spam\ ham) = -(\frac{1}{2} * log_2(\frac{1}{2}) + \frac{1}{2} * log_2(\frac{1}{2})) = 1$$

$$H(holiday = 0) \widehat{=} H(spam\ spam) = -(\frac{2}{2} * log_2(\frac{2}{2})) = 0$$

$$H(holiday) = \frac{2 * 1 + 2 * 0}{4} = 0.5$$

$$IG(holiday) = H(D) - H(holiday) = 0.81 - 0.5 = 0.31$$

$$H(I = 1) \widehat{=} H(spam\ ham) = -(\frac{1}{2} log_2(\frac{1}{2}) + \frac{1}{2} log_2(\frac{1}{2})) = 1$$

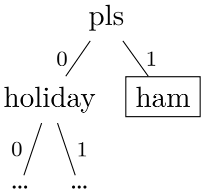
$$H(I = 0) \widehat{=} H(spam\ spam) = -(\frac{2}{2} log_2(\frac{2}{2})) = 0$$

$$H(I) = \frac{2 * 1 + 2 * 0}{4} = 0.5$$

$$IG(I) = 0.81 - 0.5 = 0.31$$

$$IG(holiday) = IG(I)$$

The IG for the features “holiday” and “I” are the same, so randomly pick one, e.g. “holiday”:

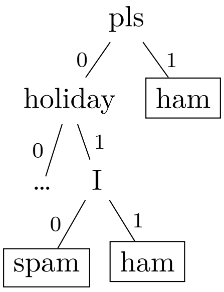


IG for features, if $holiday = 1$ **and** $pls = 0$**:**
Dataset partition for $holiday = 1$ and $pls = 0$:

I	class
0	spam
1	ham

$$
\begin{aligned}
H(D_{pls=0,holiday=1}) \widehat{=} H(spam\ ham) &= -(\frac{1}{2} * log_2(\frac{1}{2}) + \frac{1}{2} * log_2(\frac{1}{2})) = 1 \\
H(I=1) \widehat{=} H(ham) &= -(\frac{1}{1} log_2(\frac{1}{1})) = 0 \\
H(I=0) \widehat{=} H(spam) &= -(\frac{1}{1} log_2(\frac{1}{1})) = 0 \\
H(I) &= \frac{1*0+1*0}{2} = 0 \\
IG(I) &= 1 - 0 = 1
\end{aligned}
$$

New tree:

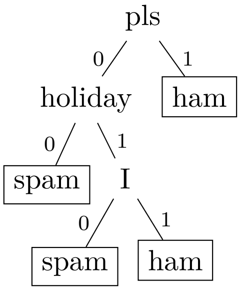


IG for features, if $holiday = 0$ and $pls = 0$:
Dataset partition for $holiday = 0$ and $pls = 0$:

I	class
1	spam
0	spam

$$
H(D_{pls=0,holiday=0}) \widehat{=} H(spam\ spam) = -(\frac{2}{2} * log_2(\frac{2}{2})) = 0
$$

We can stop since the entropy for the dataset partition is 0.
New (and final) tree:



Tranforming the three texts that we want to classify into the feature representation, we get:

holiday	I	pls
0	0	1
1	0	0
1	1	0

Using the tree on this feature matrix, we get:

SMS text	Predicted class
Hmm...my uncle just informed me that he's paying the school directly. So pls buy food.	ham
Dear Matthew please call 09063440451 from a landline, your complimentary 4*Lux Tenerife holiday or £1000 CASH await collection. ppm150 SAE T&Cs Box334 SK38XH.	spam
Camera quite good, 10.1mega pixels, 3optical and 5digital dooms. Have a lovely holiday, be safe and i hope you hav a good journey! Happy new year to you both! See you in a couple of weeks!	ham

When adding the feature "you" to our original dataset, we get:

holiday	I	pls	you	class
1	0	0	1	spam
0	1	0	1	spam
0	0	0	1	spam
0	1	1	0	ham
0	0	1	0	ham
1	1	0	0	ham

IG for feature “you”:

$$H(D)\widehat{=}H(spam\ spam\ spam\ ham\ ham\ ham) = -(\frac{3}{6} * log_2(\frac{3}{6}) + \frac{3}{6} * log_2(\frac{3}{6})) = 1$$

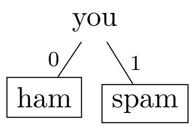
$$H(you = 1)\widehat{=}H(spam\ spam\ spam) = -(\frac{3}{3} * log_2(\frac{3}{3})) = 0$$

$$H(you = 0)\widehat{=}H(ham\ ham\ ham) = -(\frac{3}{3} * log_2(\frac{3}{3})) = 0$$

$$H(you) = \frac{3 * 0 + 3 * 0}{6} = 0$$

$$IG(you) = H(D) - H(you) = 1 - 0 = 1$$

The IG of the single feature “you” is 1, that means this feature provides all the information we need to correctly classify the data in the dataset. Hence, we do not need to calculate the IG for the other features, since the single feature “you” is enough to cover the whole dataset. This makes sense, since there is a 1-to-1 correspondence between the values of “you” and the classes. The final decision tree we get then looks like:



Exercise 2.

For this part of the exercise, you need to write Python code.

The dataset in table 1 is part of a larger dataset from `https://archive.ics.uci.edu/dataset/228/sms+spam+collection`. From the course website, download the files `https://lehre.idh.uni-koeln.de/site/assets/files/5151/smsspamcollection_train.tsv` and `https://lehre.idh.uni-koeln.de/site/assets/files/5151/smsspamcollection_test.tsv`, which contain a train and test split for this dataset, with each column representing a token feature. The column containing the classes “spam” and “ham” is called “___class___”[1]. Train a sklearn Decision Tree on the train set and let it predict on the test set. Use sklearn’s `classification_report` function to obtain several evaluation metrics for your prediction. Also visualize the tree using sklearn’s `plot_tree` function.

[1] This is because there is already a feature called “class”

Solution 2.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import plot_tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import pandas as pd

# Read data and split into features and classes
train_data = pd.read_csv("SMSSpamCollection_train.tsv", sep="\t")
test_data = pd.read_csv("SMSSpamCollection_test.tsv", sep="\t")
X_train = train_data.drop("__class__", axis=1)
y_train = train_data["__class__"]
X_test = test_data.drop("__class__", axis=1)
y_test = test_data["__class__"]

# Intialize tree algorithm
clf = DecisionTreeClassifier(criterion="entropy", random_state=42) #
                                       random_state makes sure that we always
                                        get the same tree in case that there
                                       are random decisions for picking
                                       features

# Train decision tree on train data
clf = clf.fit(X_train,y_train)

# Test trained tree on test data
y_pred = clf.predict(X_test)

# Evaluate predictions against gold test data
print(classification_report(y_test, y_pred))

# Plot tree
plot_tree(clf,
          feature_names = X_train.columns,
          class_names=["ham", "spam"]
         )
```