Information Processing and Retrieval

Instituto Superior Técnico 2020

Lab 6: Document classification

Let us recover the *20 Newsgroup* dataset (http://qwone.com/~jason/20Newsgroups/).

As an alternative to retrieve all the document collection, you can select a standard split of the collection into training and test sets.

```
from sklearn.datasets import fetch_20newsgroups
train = fetch_20newsgroups(subset='train')
test = fetch_20newsgroups(subset='test')

from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer( use_idf=False )
trainvec = vectorizer.fit_transform(train.data)
testvec = vectorizer.transform(test.data)
```

You can see the first 10 documents in the dataset using train.data[:10] and the classes of those documents using train.target[:10]. You will notice that the classes are represented as numbers. To see the class names you can use: train.target$_n ames$.

Once you do this to all data, you can fit a classifier on the training data and test it on the testing data.

```
from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
classifier.fit(trainvec, train.target)
classes = classifier.predict(testvec)
```

The scikit-learn library also provides classes to evaluate classification results.

```
from sklearn import metrics
print metrics.accuracy_score(test.target, classes)
print metrics.classification_report(test.target, classes)
```

1 Classifying *20 newsgroups*

1.1

Implement a classifier for the 20 Newsgroups collection and measure its performance.
You can Use for instance a Multinomial Naïve Bayes classifier, available in scikit-learn.

1.2

Try to improve the classification by:

(a) Removing very rare words (e.g. words that occur less than 2 times) or very frequent words (e.g. words that occur in more than 90% of the documents) using the *Vectorizer* facilities provided by scikit-learn

(b) Compare the performance against alternative classification algorithms, such as:

- a nearest neighbour classifier (sklearn.neighbors.KNeighborsClassifier)
- the perceptron algorithm (sklearn.linear_model.Perceptron)
- support vector machines (sklearn.svm.LinearSVC)

2 Pen and Paper Exercises

2.1 BM25 calculus

2.2 Performing classification

Consider the following six textual documents, each associated to one of three possible classes.

ID	Document	Class
D1	the movie is nothing but great	Positive
D2	mixed feelings about the movie	Neutral
D3	not so great	Negative
D4	great fantastic movie	Positive
D5	good movie overall	Positive
D6	overall the movie is terrible	Negative

(a) Estimate the parameters of a binary naïve Bayes model required for classifying the document $<$ *great movie overall* $>$.

Which would be the most likely class for the given document?

Present involved calculations. Use maximum likelihood estimation without considering any smoothing technique.

(b) (*homework*) Estimate the parameters of a perceptron classifier based on the first 3 training instances, discriminating the positive instances from all other instances (i.e. the negative and the neutral). Start with an all-zero parameter vector, consider binary representations for the documents, and consider a single iteration over the training instances.

2.3 Evaluating a classifier

Consider a binary classification problem, where each instance can be assigned to either a positive or a negative class. Consider also that you have a dataset D with 10 instances, each assigned to the corresponding class by a domain expert E and by a classifier C.

$$E =< +, +, -, -, +, +, -, -, +, - >$$
$$C =< +, +, -, +, +, -, -, -, -, - >$$

(a) Draw the confusion matrix for the aforementioned classification results.

(b) Compute the accuracy, precision, recall and F1-measure.

(c) Using the kappa statistic introduced earlier in the course, discuss whether there is considerably agreement between E ad C.