SNLP
Exercise 7

Submission date: 16.06.2017, 23:59

1 Naive Bayes Classifier

1.1 (3 points)

Create a **Naive-Bayes** Classifier using word frequencies as features, from the materials provided in *author*1 and *author*2. Use unigram distribution with Lidstone smoothing. Assume the document counts to be representative of the class probabilites. Now classify the texts in the *test_author* folder by their author and report the results.

2 Features

Use the documents provided in the train folder to construct a vocabulary. Perform the following tasks:

- Tokenize
- Remove stop words using the given stopwords.txt file
- Lowercase the tokens
- Perform lemmatization and stemming

2.1 (2 points)

- Now compute Pointwise Mutual Information(PMI) between each term and each topic. Compute PMI in the case we want each term to discriminate well for a single category. (1 point)
- Use PMI to do feature selection such that it results in 10 features and report them. By this how much have you reduced the dimension of your problem. (1 point)

3 Classification with Reuters dataset

For this exercise you need to install scikit-learn library on your computer. Import the Reuters dataset from NLTK and perform the following: (you can use NLTK functionalities for this)

- Tokenize
- Perform Stemming
- Remove Stopwords(use the English stopwords provided in NLTK)
- Consider words which are of length at least 3

After you have pre-processed the corpus following the above mentioned guidelines, perform the following tasks:

3.1 (1 point)

Create a subset of the Reuters dataset and split it into training and test sets respectively by considering only those categories that have at least one document in the training set and the test set.

3.2 (1 point)

Represent the data such that it reflects the multi-label nature of the classification, i.e., represent the category of each document as a list of bits which has the value 1 for the specific category to which document belongs and 0 for every other category. (Hint: Use sklearn.preprocessing.MultiLabelBinarizer)

3.3 (2 points)

Train a one-vs-rest SVM classifier on the represented training data.(Hint: Use sklearn.svm.LinearSVC and sklearn.multiclass.OneVsRestClassifier)

3.4 (1 point)

Predict the labels for each one of the represented testing documents.

4 Bonus Problem: Term Frequency-Inverse Document Frequency(tf-idf)

Use the documents provided in the train folder to construct a vocabulary. Perform the following tasks:

- Tokenize
- Remove stop words using the given stopwords.txt file
- Lowercase the tokens
- Perform lemmatization and stemming

4.1 (4 points)

Information on **tf-idf** can be obtained from here: `https://en.wikipedia.org/wiki/Tf-idf`

- Compute **tf-idf** for each word in the vocabulary and perform feature selection using 500 greatest **tf-idf** words. Briefly specify the advantages of using tf-idf over document frequency. (1 + 1 points)
- Use the features obtained from above to classify the text file "test_2.txt" by **Naive Bayes** Classifier and report the result. (1 point)
- Now try KNN as your classifier with $K = 1$. Use the frequency of the words as values of each feature in the feature vector of each document. Also consider the tf-idf features from above for performing classification with KNN. For computing distance between two vectors use the *Euclidean* norm. Compare your findings. (1 point)

5 Submission Instructions: Read carefully

- You can form groups of maximum 2 people.
- Submit only 1 archive file in the ZIP format with name containing the MN of all the team members, e.g.:

  Exercise_01_MatriculationNumber1_MatriculationNumber2.zip

- Provide in the archive:
  - your code, accompanied with sufficient comments,
  - a PDF report with answers, solutions, plots and brief instructions on executing your code
  - a README file with the group member names, matriculation numbers and emails
  - Data necessary to reproduce your results [1]
- The subject of your submission mail must contain the string "[SNLP]" (including the braces) and explicitly denoting that it is an exercise submission, e.g:

  [SNLP] Exercise Submission 01

- Depending on your tutorial group, send your assignment to the corresponding tutor:

[1] If you feel that these files are beyond reasonable size for an email submission and also reasonably convenient, please provide a means for us to access them online

- David Adelani: *s8daadel@stud.uni-saarland.de*
- Rajarshi Biswas: *rbisw17@gmail.com*
- Xiaoyu Shen: *s8xishen@stud.uni-saarland.de*

6 General Information

- In your mails to us regarding the tutorial please add the tag "[SNLP]" in the subject accompanied by an appropriate subject briefly describing the contents.
- Feel free to use any programming language of your liking. However we strongly advise in favor of Python, due to the abundance of available tools (also note that Python3 comes with an excellent native support of UTF8 strings).
- Avoid using libraries that solve what we ask you to do (unless otherwise noted).
- Avoid building complex systems. The exercises are simple enough.
- Do not include any executable files in your submission, as this may cause the e-mail server to reject it.
- **In case of copying, all the participants (including the original solution) will get 0 points for the whole assignment. Note: it is rather easy to identify a copied solution. Plagiarism is also not tolerated.**
- **Missing the deadline will result in 0 point for the whole assignment. So make sure to submit on time.**
- **Please submit in your solutions necessary to support your claims. Failure to do so, might results in reduction of points in the relevant questions.**
- Each assignment has 10 points and perhaps some bonus points (usually 2 or 3). In order to qualify for the exams, you need to have 2/3 of the total points. For example, in case there are 12 assignments, you need to collect at least 80 out of the 120 points to be eligible for the exams. A person that gets 10 plus 2 bonus points in every exercise, needs to deliver only 7 assignments in order to be eligible for the exams, since 7*12=84.
- Attending the tutorial gives 1 point increase for the corresponding assignment.
- Presenting a solution will give bonus equal to the point of the presented question (Up to 4 poins). The bonus is divided evenly in the group member. Each group can present up to 3 times.
- Exercise points (including any bonuses) guarantee only the admittance to the exam, however have no further effect on the final exam grade.