Übung 1: Textklassifikation mit Naive Bayes

Tutorin: Jasmin Heierli, jasmin.heierli@uzh.ch

Abgabedatum: Mittwoch 16.03.2016, 20.00

Einführung

Ziel der Übung ist es, einen Classifier zu trainieren, der die Dokumente aus dem NLTK movie-review Corpus in die zwei Klassen *positiv* und *negativ* einteilt. Als Classifier soll der NLTK-Naive-Bayes-Classifier verwendet werden und zur Auswertung eine 10-fache Kreuzvaldierung durchgeführt werden. Dabei soll die Accuracy für jedes Trainings- und Testmengenpaar, sowie der Durchschnittswert berechnet werden.

Jede Funktion und jeder Parameter muss dokumentiert sein.

1 Daten einlesen, bereinigen und aufbereiten

Kapitel 6 im NLTK-Buch (`http://www.nltk.org/book/ch06.html`) beschreibt, wie man die movie reviews einlesen und vorbearbeiten kann. Lies Kapitel 6.1 bis und mit 6.1.3 sorgfältig und teste den Code aus. Falls das NLTK-Toolkit auf deinem Laptop fehlt, kannst du es wie auf der Web-Site[1] beschrieben installieren. Ansonsten sind alle Packages und Daten auf dem Studierendenserver r2d2.ifi.uzh.ch installiert.

2 Wähle 2 Merkmalstypen

Extrahiere die 200 häufigsten Unigramme und Bigramme, anhand derer der Classifier die Dokumente unterscheiden soll.

- Der NLTK-Classifier behandelt numerische Features alle gleich. Das heisst der Unterschied zwischen 99 und 100 ist gleich wie zwischen 1 und 100.
- Achte dich auf die System-Auslastung und optimiere gegebenfalls den Code oder schränke die Featurewahl zusätzlich ein.
-

[1] `http://www.nltk.org/install.html`

3 Training und Test

Um den Classifier zu trainieren, musst die Daten noch in Trainings- und Testset unterteilen. Es soll eine 10-fach-Kreuzvalidierung durchgeführt werden. Das heisst, das Korpus wird in 10 gleiche Teile unterteilt, wobei 1 Teil zum Testen und 9 Teile zum Trainieren des Classifiers sind. Dabei ist immer ein anderer Teil des Korpus Test-Set und der Rest Trainings-Set, so dass man am Ende 10 unterschiedliche Werte für die Genauigkeit berechnen kann.

- Die Daten müssen am Anfang mit `random.shuffle(list)` durchmischt werden.
- Jedes Set muss die für den Classifier richtige Datenstruktur haben. `[({feature:true,...},label`

3.1 Zu verwendender Classifier

```
...
classifier = NaiveBayesClassifier.train(trainfeats)


accurracy = nltk.classify.util.accuracy(classifier, testfeats)


classifier.show_most_informative_features (n)
```

4 Abgabe

Zip-Datei mit:

- Python-Skript
- kurze Zusammenfassung mit den Ergebnissen aller 10 Trainingsdurchläufe, und deinen Beobachtungen. Prüfe für zwei Modelle, ob ihre einflussreichsten Features identisch sind.
- Kurzes Feedback, wo du von der Übung profiteren konntest, wo eher nicht so.

Viel Spass und Erfolg!