Übungen 1: Informationstheorie und Klassifikation

Abgabedatum Sonntag, 18. März 2016, 13:00h MEZ

Setup

Installiere Anaconda für Python 3.6 wie auf der Web-Site[1] beschrieben. Um die benötigten Packages zu installieren, führe im *Terminal* oder in der *Eingabeaufforderung* folgenden Befehl aus:

```
conda install nb_conda
conda env create -f ki2_fs2018_env.yml
```

Um die Daten nutzen zu können, müssen wir diese über das NLTK-Toolkit herunterladen. Führe dazu folgende Befehle aus:

```
$ python
>>> import nltk
>>> nltk.download("names")
>>> nltk.download("europarl_raw")
```

Um das Notebook zu starten gehe im Terminal zum Pfad in dem das *Language Classification.ipynb* und das *Sklearn Intro.ipynb* Notebook gespeichert sind und führe folgenden Befehl aus:

```
jupyter notebook
```

1 Maximum-Likelihood-Modelle und Mutual Information

In dieser Aufgabe testest du mögliche Merkmale auf ihren erwarteten Informationsgewinn bezüglich einer binären Klassifikation. Als Beispiel dient uns die Geschlechtsklassifikation für englische Vornamen, welche du in der Vorlesung kennen gelernt hast. Die **Zufallsvariable C** mit den beiden möglichen Ergebnissen $\Omega = \{male, female\}$ repräsentiert die beiden zulässigen **Klassen** der Vornamen. Die boolesche **Zufallsvariable F** ($\Omega = \{True, False\}$) ist wahr, wenn ein Vorname auf -a endet und sonst falsch. Dieses Merkmal F ist vordefiniert mit Hilfe der Merkmalsextraktionsfunktion `feature(Vorname)`.

1.1 Verstehen der Kodevorlage

1. In den Vorlesungsunterlagen findest du die Datei `Sklearn Intro.ipynb`. Studiere den Quellcode und die Kommentare dazu. Falls du die Klasse `collection.Counter` für Häufigkeitsverteilungen nicht kennst, studiere deren offizielle Dokumentation[2]. Falls du Listenkomprehension (*list comprehension*) und Generatorausdrücke (*generator expressions*) nicht kennst, studiere die entsprechenden Abschnitte im NLTK-Buch[3].

2. Im Abschnitt 4 des Notebooks findest du den *Data Analysis* Teil. Berechne aus den Häufigkeitsverteilungen (*frequency distributions*) folgende Maximum-Likelihood-Wahrscheinlichkeiten sowohl für die Gesamtdaten als auch für ein zufällig gezogenes Sample von 500 Vornamen:

[1] https://www.anaconda.com/download/
[2] `https://docs.python.org/2/library/collections.html#collections.Counter`
[3] Z.B. `http://www.nltk.org/book/ch04.html#generator_expression_index_term`

(a) $p(C = female)$
(b) $p(F = True)$
(c) $p(C = female|F = True)$
(d) $p(F = True|C = Female)$
(e) $p(C = female, F = True)$

Gib die entsprechenden Resultate mit einem Titel wie in der Vorlage mittel `print` aus.

1.2 Entropie und Informationsgewinn implementieren

1. Implementiere die Formel, welche die Entropie berechnet, und evaluiere sie für C:

$$H(X) = -\sum_{x \in X} p(x) \log(p(x)) \tag{1}$$

2. Selbsttest: Die Formel sollte 0.9510 ergeben.

3. Implementiere die Mutual Information für $I(C; F)$ und evaluiere sie auf den Gesamtdaten und auf einem zufälligen Sample von 500 Vornamen:

$$I(X; Y) = \sum_{x \in X, y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{2}$$

Hinweis: $0 \log(0)$ soll im Kontext der Informationstheorie als 0 definiert sein.
Hinweis: `math.log` in Python berechnet standardmässig den natürlichen Logarithmus. Der Logarithmus mit Basis 2 wird mit dem optionalen Basis-Argument aufgerufen: `math.log(8, 2)` evaluiert zu `3.0`.

4. Selbsttest: Die Formel sollte 0.1524 ergeben auf den Gesamtdaten.

2 Klassifikation und Experimentation

Im Notebook *Sklearn Intro.ipynb* wird der generelle Aufbau des *Scikit-Learn*[4] gezeigt. In dieser Aufgabe geht es darum, selber mit dem Toolkit zu experimentieren. Im Notebook *Language Classification.ipynb* findest du den rudimentären Aufbau eines Experiments, um die Sprache einzelner Wörter zu klassifizieren.

1. Studiere den Code und beschreibe kurz, was er macht.

2. *Feature Extraction*: Im Code wird der *CountVectorizer* benutzt für die Merkmalsextraktion. Lass den Code mit verschiedenen Parameter-Settings im *CountVectorizer* laufen. Nutze drei verschiedene Werte *ngram-range* und beschreibe die Resultate. Verwende auch nicht *lower-case character*s und beschreibe die Resultate.

3. *Model*: Verwende verschiedene Algorithmen aus dem Sklearn-Sortiment: Wähle basierend auf dem Chart im *Sklearn Intro.ipynb* Notebook zwei weitere Algorithmen aus und schreib die Resultate auf.

4. *Model Selection*: Nutze *GridSearchCV*, um für einen der drei Algorithmen die beste Kombination an Hyperparametern zu finden.

[4] http://scikit-learn.org

3 Feedback

Bitte melde kurz zurück, welche Aufgaben du hilfreich und welche du nicht hilfreich fandest, um dein Lernen zu unterstützen.

Viel Spass und Erfolg beim Programmieren!