MITx 6.86x

Machine Learning with Python-From Linear Models to Deep Learning

6. Automative review analyzer

Bookmark this page

Now that you have verified the correctness of your implementations, you are ready to t
this project: building a classifier that labels reviews as positive or negative using text-b
linear classifiers that you implemented in the previous section!

The Data

The data consists of several reviews, each of which has been labeled with -1 or $+1$, c
negative or positive review, respectively. The original data has been split into four files:

- `reviews_train.tsv` (4000 examples)
- `reviews_validation.tsv` (500 examples)
- `reviews_test.tsv` (500 examples)

To get a feel for how the data looks, we suggest first opening the files with a text edito
or other scientific software package (like pandas).

Translating reviews to feature vectors

We will convert review texts into feature vectors using a **bag of words** approach. We st
words that appear in a training set of reviews into a **dictionary** , thereby producing a lis

We can then transform each of the reviews into a feature vector of length d by setting
feature vector to 1 if the i^{th} word in the dictionary appears in the review, or 0 otherwis
two sample documents “Mary loves apples" and “Red apples". In this case, the dictionar
$\{\text{Mary}; \text{loves}; \text{apples}; \text{red}\}$, and the documents are represented as $(1;1;1;0)$ and (

A bag of words model can be easily expanded to include phrases of length m. A **unigra**
which $m = 1$. In the example, the unigram dictionary would be $(\text{Mary}; \text{loves}; \text{apples}$
case, $m = 2$, the dictionary is $(\text{Mary loves}; \text{loves apples}; \text{Red apples})$, and repres
sample are $(1;1;0)$, $(0;0;1)$. In this section, you will only use the unigram word featu
already implemented for you in the bag of words function.

In utils.py, we have supplied you with the load data function, which can be used to
returns the labels and texts. We have also supplied you with the bag_of_words functio
which takes the raw data and returns dictionary of unigram words. The resulting diction
`extract_bow_feature_vectors` which computes a feature matrix of ones and zeros
input for the classification algorithms. Using the feature matrix and your implementatio
from before, you will be able to compute θ and θ_0 .

Previous Next