CSE 258, Fall 2017: Homework 2

Instructions

Please submit your solution **by the beginning of the week 5 lecture (Oct 30).** Submissions should be made on **gradescope**. Please complete homework **individually**.

You will need the following files:

Logistic regression and validation code stub :

`http://jmcauley.ucsd.edu/code/homework2_starter.py`

Executing the code requires a working install of Python 2.7 or Python 3 with the scipy packages installed. **Please include the code of (the important parts of) your solutions.**

Tasks (Classifier evaluation)

Similar to the classifier we built in the last homework, a stub has been provided that runs a logistic regressor on the beer rating data (see link above). The stub predicts whether a beer has an ABV ≥ 6.5 based on its five rating scores:

$$p(\text{positive label}) = \sigma(\theta_0 + \theta_1 \times \text{`review/taste'} + \theta_2 \times \text{`review/appearance'} + \theta_3 \times \text{`review/aroma'} + \theta_4 \times \text{`review/palate'} + \theta_5 \times \text{`review/overall'})$$

The stub runs logistic regression with a hyperparameter $\lambda = 1.0$. We will use this stub to further improve and evaluate our classifier.

1. The code currently does not perform any train/test splits. Split the data into training, validation, and test sets, via 1/3, 1/3, 1/3 splits. Use the first third, second third, and last third of the data (respectively). After training on the training set, report the accuracy of the classifier on the validation and test sets (1 mark).

2. Let's come up with a more accurate classifier[1] based on a few common words in the review. Build a feature vector to implement a classifier of the form

   $$p(\text{positive label}) = \sigma(\theta_0 + \theta_1 \times \#\text{`lactic'} + \theta_2 \times \#\text{`tart'...}),$$

   where each feature corresponds to the number of times a particular word appears. Base your feature on the following 10 words: "lactic," "tart," "sour," "citric," "sweet," "acid," "hop," "fruit," "salt," "spicy." **Convert the reviews to lowercase before counting.**

3. Report the number of true positives, true negatives, false positives, false negatives, and the *Balanced Error Rate* of the classifier on the test set (1 mark).

4. (**Hard**) Our classifier is possibly less effective than it could be due to the issue of *class imbalance* (i.e., an uneven number of the datapoints have a positive label). Show how you would adjust the gradient ascent code provided such that the classifier would be approximately 'balanced' between the positive and negative classes. Report the Balanced Error Rate (on the train/validation/test sets) for the new classifier (1 mark).

5. Implement a training/validation/test pipeline so that you can select the best model based on its performance on the *validation* set. Try models with $\lambda \in \{0, 0.01, 0.1, 1, 100\}$. Report the performance on the training/validation/test sets for the best value of λ (1 mark).

Tasks (Dimensionality reduction):

Next, we'll run dimensionality reduction on the same data, using the word features from the previous question (you can drop the constant feature). Specifically we'll try to find the principal components of our 10 word features. For this question, use the *training* set constructed from the initial 1/3, 1/3, 1/3 splits of the data.

6. Find and report the PCA components (i.e., the transform matrix) using the week 3 code (1 mark).

7. Suppose we want to compress the data using just two PCA dimensions. How large is the reconstruction error when doing so (1 mark)?[2]

[1] Maybe less accurate actually!

[2] Hint: You should be able to solve this *without* explicitly computing the reconstruction.

8. Looking at the first two dimensions of our data in the PCA basis is an effective way to 'summarize' the data via a 2-d plot. Using a plotting program of your choice, make a 2-d scatterplot showing the difference between 'American IPA' style beers versus all other styles (e.g. plot American IPAs in red and other styles in blue) (1 mark).