CSE 258, Winter 2017: Homework 2

Instructions

Please submit your solution **by the beginning of the week 5 lecture (Feb 6).** Submissions should be made on **gradescope**. Please complete homework **individually**.

You will need the following files:

UCI Wine Quality Dataset :

`http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv`

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 *white wine* rating data (see link above). Wines rated > 5 are treated as positive instances, others as negative instances. That is, we have a predictor of the form

$$p(\text{positive label}) = \sigma(\theta_0 + \theta_1 \times \text{`fixed acidity'} + \theta_2 \times \text{`volatile acidity'} + \ldots + \theta_{11} \times \text{`alcohol'}.$$

The stub provided runs the regularization pipeline to select a model, using logistic regression (implemented via gradient ascent). We will use this stub to further improve and evaluate our classifier.

1. The code splits the data into train, validation, and test sets, via 1/3, 1/3, 1/3 splits. This could be risky (and violate some of our assumptions!) if the data are not randomly sorted. Randomly re-shuffle the data, and report the train/validation/test performance for $\lambda \in \{0, 0.01, 1.0, 100.0\}$.

From now on we'll perform experiments with $\lambda = 0.01$. Let's consider some other performance measures besides just accuracy. Use only the test set from the original 1/3, 1/3, 1/3 split for this question.

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

3. Suppose we care about using our model to rank predictions, i.e., we care about the predictions where the classifier is most confident of a positive label. By sorting the predictions by confidence, compute the precision and the recall when returning the top 10, 500, and 1000 predictions (1 mark).

4. Plot precision versus recall as the number of results considered varies (from 1 to len(y_test)). You may use any plotting library, e.g. `http://matplotlib.org/api/pyplot_api.html` (1 mark).

Tasks (Dimensionality reduction):

Next, we'll run dimensionality reduction on the same data. Specifically we'll try to find the principal components of our 11 wine-related features. For this question, use the *training* set constructed from the initial 1/3, 1/3, 1/3 splits of the data.

5. Suppose we wanted to 'compress' our data just by replacing each of the points with their mean vector.[1] What is the 'reconstruction error', here defined as

$$\sum_{x \in X} \|\bar{x} - x\|_2^2$$

for the compressed data (recall that $\|y\|_2^2 = \sum_i y_i^2$) (1 mark)?

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

[1] i.e., for each of the 11 dimensions, replace it by the mean of the 4898/3 values for that dimension.

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

We now have a new representation for our data, such that its first dimensions best explain its variance. But if we discard some of the low variance dimensions, we'd like to find out whether the reduced dataset is still useful for prediction.

8. Train a simple linear regressor (no regularization, using the training set) to predict the quality score, using increasingly many PCA dimensions, i.e.,

$$\begin{aligned} \text{model 1:} \quad \text{quality} &= \theta_0 + \theta_1 \times (\text{first pca dimension}) \\ \text{model 2:} \quad \text{quality} &= \theta_0 + \theta_1 \times (\text{first pca dimension}) + \theta_2 \times (\text{second pca dimension}) \\ \textit{etc.} & \end{aligned}$$

Write (or plot) how the MSE changes (on the train and test sets) as more and more dimensions are used (1 mark).

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