CPSC 483- Data Mining and Pattern Recognition

Fall 2017 - Project 1, due September 21

In this first project, you will use scikit-learn and related Python libraries to perform several regression experiments against the Olympics data from Chapter 1 of the textbook

Setup

For this project you will need

- The Python programming language (either Python 2 or 3 is fine)
- scikit-learn and its prerequisites (Numpy, Scipy, and matplotlib)
- The file `olympics.mat` from the textbook authors' website.

Recall from the Syllabus that an easy way to get up and running quickly is the Anaconda Python distribution.

Allocating time for this project

The difficulty level of this project will depend largely on how comfortable you are with Python and scikit-learn. It may take anywhere from a few hours to several days to complete. Be prepared to read the documentation for scikit-learn, NumPy, SciPy, and matplotlib and consult the Safari library.

You are welcome to discuss the project with other students in the course and to offer and solicit advice. You may also, of course search for help on the Internet. In all cases, however, remember that any code you write must be your own.

Experiments

1. Load the data in `olympics.mat` into Python. This file contains several datasets: `male100`, `male200`, `male400`, `female100`, `female200`, and `female400`. For each dataset, the first column is the year, and second column is the time in seconds. You may ignore the other columns.

2. Use `matplotlib.pyplot` to plot `male100`, reproducing Figure 1.1 in the textbook.

3. Use `sklearn.linear_model.LinearRegression` to fit `male100`. List the coefficients, then predict the values for *x* = 2012 and *x* = 2016. Compare your results to those found in Section 1.2.

4. Plot `male100` and the line you fit in experiment *(3)* above, reproducing Figure 1.5 in the textbook.

5. Use linear regression to fit a line to `female400`. How does the error for this model compare to the error when fitting a line to `male100`?

6. Fit a 3rd order polynomial to `female400`. Does the error improve?

7. Fit a 5th order polynomial to `female400`. Does the error improve?

8. Use LOOCV for both the 3rd and 5th order polynomials. (Hint: use `sklearn.model_selection.LeaveOneOut`.) Which polynomial is a better choice?

9. Use `sklearn.linear_model.Ridge` with $\alpha = 0.1$ to fit a 5th order polynomial to `female400`. How do the coefficients compare to those found with linear regression?

10. Use `sklearn.linear_model.RidgeCV` to find the best value for α across the range 0.001, 0.002, 0.004, 0.01, 0.02, 0.04, 0.1, 0.2, 0.4, 1.0.

Documentation

In your Python code, use comments to clearly identify

- which experiment is being performed
- the output produced by each command
- the answers to the questions asked in experiments 5-9.

Submission

Turn in the Python code for your project by placing it in the `project1/` subdirectory of the folder that will be shared with you on Dropbox.