CS 678 Machine Learning
Programming Project #1

Data Visualization and Analysis

Overview

This is a very simple "warm-up" exercise. The main purpose of the assignment is to begin thinking about and analyzing data and manipulating it via a program.

Background

Suppose you have started a new blog on Medium. Their Analytics division provides you with a traffic summary of visitors to your site. This datafile consists simply of a list of #hits per hour for the entire previous month (31 days).

You would like to get a feel for the popularity of your site. Are more people reading it the longer it's been active? Has "word of mouth" had any effect, or has interest in it begun to tail off? What kind of traffic can you expect in the future?

Specifications

You decide to download the analytics data, visualize it, and produce a trend line in order to predict future performance. That's basically the assignment.

1. Pre-processing: read in and clean the data

The datafile (`hits.txt`) comes as a comma-separated list of values: each line contains the hour of the month and the number of visits that occurred during that hour (e.g. 1,2272). There are 24×31=744 total lines. A quick glance at the file shows that some type of error has prevented the data from being measured and/or recorded at certain times, represented as a '`nan`' ("not a number") value in the datafile. You're going to have to do something to deal with this problem.

- Develop, document, and justify a solution.

2. Visualization: display the data

In order to get an initial feel for the data, the next step is to visualize it. Create a scatterplot (a Cartesian display of two-variable data) within your program, using a Python graphics library (e.g. matplotlib).

- Document and describe your approach (i.e. technologies used).
- What does the visualization tell you about visits to your site?

3. Analysis and discussion: perform simple linear regression on the data

You decide to begin with the simplest analysis: linear regression. Linear regression is a method for fitting a curve, in this case a straight line, to a set of points. The *slope* of the line represents the correlation between the *x* and *y* values; the *intercept* gives the center of mass of the data points.

There are several different ways of performing a linear regression, typically based on the *least-squares* method that attempts to minimize the sum of squared residuals (i.e. the error). A simple method follows.

Obtain/calculate:

- Σ X: the sum of all X values
- Σ Y: the sum of all Y values
- Σ XY: the sum of the products of each X,Y pair
- Σ X^2: the sum of the squares of every X value
- Σ Y^2: the sum of the squares of every Y value

Suppose N is the number of data points. Then the relevant calculations are:

$$\text{slope} = \frac{(N\sum XY) - (\sum X \sum Y)}{(N\sum X^2) - (\sum X)^2}$$

$$\text{intercept} = \frac{\sum Y - (\text{slope} \sum X)}{N}$$

With these values you can create the regression equation:

$$Y = \text{intercept} + \text{slope} * X$$

and use it to make predictions about the future.

Perform the following:

- Create a visualization of the regression analysis (i.e. plot the trendline over the scatter plot of the data).
- Assuming the regression equation accurately captures current and expected visitor behavior, how many visits would you expect at Noon on the fifth day of the next month?

Discussion: Scientific analyses often include a discussion of the possible weaknesses of the presented approach, and suggestions for future work.

For example: apparently a well-read blogger made a favorable mention of your site towards the end of the month.

- Do you think a simple linear analysis captures the expected popularity of your site?
- What other analytical approaches might produce a better model?

Notes:

- This, and all programming assignments, must be performed in Python. All computations of the basic assignment should be performed by your program, not by a built-in library routine. However, for validation you are encouraged to compare your results to packaged routines (e.g. Excel, R. SAS).
- Be sure to demonstrate good programming style and practices.
- You may work together on this assignment.

Deliverables

- Submit a single PDF containing your source-code, sample output, graphs, and all documentation describing your approach, design decisions, and answers to all questions.
- Be prepared to present and discuss your solution in class. E.g. what data structures did you employ? What graphing package/API did you use? What interesting problems did you encounter and how did you address them? What alternative analyses did you attempt?