

Problem Set 1: Maximum Likelihood and Linear Regression

Prof. Jim Rehg
College of Computing, Georgia Institute of Technology
Monday, Sept. 8, 2008

Due at the beginning of class, Wednesday Sept. 17, 2007

You do not need to typeset your answers, legible hand-written responses are fine.

1. The correlation coefficient for two scalar random variables x and t is defined:

$$\rho_{x,t} = \frac{\text{cov}[x, t]}{\sqrt{\text{var}[x]\text{var}[t]}}$$

Note that the possible range of values for $\rho_{x,t}$ is $[-1, 1]$.

(a) Suppose that $t = ax + b$ for constants a and b . Show that $\rho_{x,t} = \text{sgn}(a)$, where the sgn function returns the sign of its argument (-1 or $+1$).

(b) What is the meaning of the three cases $\rho_{x,t} > 0$, $\rho_{x,t} = 0$, and $\rho_{x,t} < 0$? Suppose we are given n scalar attributes x_i which could potentially be used to predict the value of a target variable t . Describe how the correlation coefficient could be useful in this context. Can you think of any limitations in using correlation to measure the usefulness of an attribute?

2. Download the file: http://www.cc.gatech.edu/~rehg/Classes/ps1_PR.zip. The `housing.csv` file contains housing data from 506 different neighborhoods in the suburbs of Boston. Each row gives the attribute values for a single neighborhood. An explanation of the data fields is given in `housing.txt`. The goal is to use linear regression to predict the target value (the median home value) from the other attributes.

You can use two different software packages to accomplish this. If you have access to the Matlab Statistics Toolbox, then the simplest approach for this problem is to use the built-in `regress` function. An explanation of what this function does can be found at: http://www.mathworks.com/access/helpdesk/help/toolbox/stats/index.html?access/helpdesk/help/toolbox/stats/bq_675g.html under “linear regression.” A simple example of the use of this function is given in <http://www.stanford.edu/class/msande121/Handouts/MatlabRegress.pdf>. Note that a regularized version of the linear regression solver known as *ridge* is also available.

The second option is to use Python. (The easy extensibility of Python, which will make it an attractive choice for future assignments, is not required in this case). There are at least two ways to go here. First, you can write your own regression function using the linear system solver within the Python *SciPy* library. You should avoid explicitly calculating and storing matrix inverses to obtain the solution, as this often leads to numerical instability. Alternatively, you can use the *RPy* package (see <http://rpy.sourceforge.net/>) to call the “linear model” function provided by R, an open source statistics package (see <http://www.r-project.org/>). An example of how to do this is given at: http://www2.warwick.ac.uk/fac/sci/moac/currentstudents/peter_cock/python/lin_reg/#lm (For the purpose of this assignment, it’s probably easier to work directly with R, but the combination of R with Python can be useful in other cases.) In either of these approaches you will likely need the *numPy* package and the *pyLab* package, which provides an interface to *Matplotlib* (see <http://matplotlib.sourceforge.net/> and the examples provided there). This gives an open source environment for data processing which is easily extensible, informed by modern language design, and which replicates some of the important features of Matlab.

(a) Generate a table of scatter plots showing the variation of the target variable (median home values) with each of the scalar attributes. Calculate and display the correlation coefficient in each case. Which single attribute seems to be the most strongly correlated with the target? Which single continuous attribute seems to be the least correlated with the target?

(b) Select the two attributes that are the most correlated with the target, and the two that are the least correlated. Calculate separate scalar regressions for each of these four attributes with the target. Compute the prediction error in each of the four cases. Which attribute is the best predictor, and which attribute is the worst? You must submit a plot for the best and worst attributes, showing the regression fit superimposed on the scatter plot.

(c) Now perform vector linear regression, using of all attributes simultaneously to predict the output under a linear model. Examine the weights in the resulting linear solution and compute the prediction error. How does the vector regression result compare to the solution from part (b) (based on using the single best attribute)?

3. In this exercise we will replicate some of the experiments in section 1.1 of your text that illustrate the effect of overfitting and regularization. In our case, we will use Gaussian basis functions to fit data generated by a sinusoidal curve model.

(a) Write a function that generates a dataset, consisting of sample points from the following additive gaussian measurement model: $t = y(x) + \epsilon$, where $y(x) = \sin(2\pi x)$ and $\epsilon \sim N[0, \sigma^2]$. Your function should allow you to vary n , which is the number of sample points of x , evenly spaced in the interval $[0, 1]$, and σ . Choose a value for σ so that the datasets produced by your function resemble those shown in section 1.1 of your text.

(b) Write a function that generates the design matrix and residual vector for the linear regression problem. Your function should allow you to vary N , the number of datasets, M , the number of Gaussian basis functions, and β the variance of the Gaussian bases. Your basis functions should be distributed evenly on the interval $[0, 1]$. Choose β so that your bases resemble those in Figure 3.1(b) when plotted.

(c) For $n = 25$ and $M = 12$, generate a dataset and perform linear regression. Plot the regression fit along with $y(x)$ and the individual data points (as in Figure 1.6). Superimpose plots of the M functions $w_i\phi_i(x)$ on top of the curves. Use this graph to explain what the rows and columns of the design matrix correspond to.

(d) Build a table which shows the effect of overfitting and the solution produced by regularization. Start by generating 3 datasets of sizes (approximately) $N = 15, 50, 100$ (you may need to adjust these sizes to get the desired effect). Choose a fixed number of basis functions M such that overfitting is exhibited for all datasets except for the largest value of N . Identify a value of the regularization parameter λ which avoids overfitting. Make a 4 by 4 table of plots. Each subplot should consist of the superposition of all of the regression fits from a given dataset (see the first column of plots in Figure 3.5), with the exception of the last column (explained below). The first 3 columns should correspond to your 3 datasets. The first row is the unregularized solution. The next 3 rows show solutions for three choices of λ . The first choice should still exhibit overfitting, the second choice should be the value you selected, the third choice should show increased regularization. The fourth column should replicate the second column in Figure 3.5, showing the single average response for each dataset, superimposed on the ground truth curve.

4. Let y be a scalar random variable with distribution $p(y|n)$. We construct y as follows:

$$y = \sum_{i=1}^n x_i, \quad \text{where } x_i \sim N[x_i|0, 1].$$

In words, y is formed by summing n scalar Gaussian random variables with zero mean and unit variance.

(a) Derive the maximum likelihood estimator for the parameter n . (In your analysis it is OK to treat n as a continuous parameter).

(b) Without doing a complete derivation, describe qualitatively how you would modify part (a) to take the discrete nature of n into account.

(c) **Extra Credit:** Is the maximum likelihood estimate in (a) unbiased? Explain.