

Homework 01

1) Cross-validation for regression.

Using the formula for least-squares regression derived in class, find the best-fit polynomial for the provided data. Use cross-validation to determine the optimal degree for the polynomial, with a suggested 50/50% training/testing split of the data.

Provide a plot of the training and test error as a function of the polynomial degree which indicates the optimal degree as that with minimal test error. For this optimal degree, also provide a scatter plot of the data with the best-fit model overlaid. Report the coefficients for the best-fit model.

The data are in the tab-separated file "homework_01.tsv". The first line (the "header") species the column names, here "x", the input, and "y", the output. $N = 200$ points are given.

Recall from class that the solution for the best-fit parameters θ is given by:

$$\hat{\theta} = (X^T X)^{-1} X^T y \quad (1)$$

where y is the $N \times 1$ vector of outputs and X is an $N \times (K + 1)$ matrix with elements $X_{ik} = (x_i)^k$, i.e. the k -th power of the i -th input x_i for $k = 0, \dots, K$. The vector θ is length $(K + 1) \times 1$ and gives estimates of the degree- K polynomial coefficients for the function

$$f_K(x; \theta) = \sum_{k=0}^K \theta_k x^k. \quad (2)$$

Least-squared loss measures the sum-of-squared differences between the actual and predicted values:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N (y_i - f_K(x; \theta))^2. \quad (3)$$

2) Installing/using Bash and Python.

Whether on Linux, Mac, or Windows, make sure you have access to a Bash¹ shell – fully stocked with the command line tools "sed", "awk", and "grep" – and a distribution of Python². (Note: you're not expected/required to know

¹If you're experienced with the command line and prefer an alternative shell, e.g. Tcsh, Zsh, etc., that's perfectly fine.

²Python isn't required if you're comfortable in a different scripting environment, but many class examples will be written in Python, so it may be useful to have a working version available.

how to *use* these tools yet, just to make sure they're *installed* and *working*.)

If you're using Linux your system should have Bash and Python, and it's assumed that you know how to access and use the command line.

If you're on a Mac, Bash and Python are also part of the default installation. You can access the command line by going to Applications → Utilities → Terminal. You can verify that Python's installed by entering the command "python -version".

If you run Windows, you'll need to install a tool like Cygwin to obtain a Bash shell. Cygwin is freely available here:

<http://www.cygwin.com/>

Download and run setup.exe and follow the instructions. Documentation is available, e.g.:

<http://cygwin.com/cygwin-ug-net/ov-ex-win.html>

You can either install Python as part of cygwin, by selecting it as an optional package, or with the latest release of Python 2.x from python.org:

<http://www.python.org/download/releases/2.6.2/>

You can verify that Python's installed by entering the command "python -version".

Finally, if you're using university machines (e.g. cunix or a similar unix variant) you should have access to Bash and Python.

If you're unfamiliar with Bash or Python, the following links may provide a useful reference/introduction:

<http://freeengineer.org/learnUNIXin10minutes.html>

<http://www.pixelbeat.org/cmdline.html>

<http://docs.python.org/tutorial/index.html>

<http://software-carpentry.org/> (Lectures 2,3,4)

<http://www.diveintopython.org/toc/index.html> (Chapters 1,2,3)