ISTA 421 / INFO 521 – Homework 2

Due: Friday, Sept 14, 5pm

15 points total Undergraduate / 20 points total Graduate

Ken Youens-Clark

Graduate

I conferred with students Matthew Miller and Kai Blumberg.

1. [3 points] Adapted from **Exercise 1.2** of FCMA p.35:

   Write a Python script that can find the parameters **w** (a vector of parameters) for an arbitrary dataset of x_n, t_n pairs. You will use this script to complete Exercise 2, which only requires fitting a simple line to the data (i.e., you only need to fit parameters w_0 and w_1); however, in problem 5 you will need to fit higher-order polynomial models (e.g., $t = w_0 + w_1 x + w_2 x^2 + ...$), so you must make your script here generalized to handle higher-order polynomials. The script `fitpoly_incomplete.py` is provided to help get you started. `fitpoly_incomplete.py` provides helper functions to read data, plot data, and plot the model (once you've determined the weight vector **w**), but the function for computing linear least-squares fit, `fitpoly` (starting on line 166), is *incomplete*. `fitpoly` takes as input the (one-dimensional) data vector **x**, the target values vector **t**, and a non-negative integer `model_order`, which represents the highest polynomial order term of the model; `fitpoly` is intended to return the **w** vector (as a numpy array).

   Recommended: (If needed) review the Appendix to HW 1, the brief tutorial to numpy arrays!

   Just to state the obvious: the objective of this exercise is for you to implement the linear least squares fit solution (i.e., the normal equation) in their general linear algebra form. **DO NOT** use existing least squares solvers, such as `numpy.linalg.lstsq`, or scikit learn's `sklearn.linear_model.LogisticRegression` as your implemented solution; however, it is certainly fine to use these to help *verify* your implementation's output.

   You will submit your script as a stand-alone file called `fitpoly.py`.

```
$ ./fitpoly.py -h
usage: fitpoly.py [-h] [-m int] [-t str] [-x str] [-y str] [-o str] [-s] [-q]
                  FILE

Find w-hat

positional arguments:
  FILE                  csv data file

optional arguments:
  -h, --help            show this help message and exit
  -m int, --model_order int
                        Model order (default: 1)
  -t str, --title str   Plot title (default: Data)
  -x str, --xlabel str  X axis label (default: x)
  -y str, --ylabel str  Y axis label (default: t)
  -o str, --outfile str
                        Save output to filename (default: None)
  -s, --scale           Whether to scale the data (default: False)
  -q, --quiet           Do not show debug messages (default: False)
```

2. [2 point] Adapted from **Exercise 1.6** of FCMA p.35:

Table 1.3 (p.13) of FCMA lists the women's 100m gold medal Olympic results – this data is provided in the file `womens100.csv` in the data folder. Using your script from problem 1, find the 1st-order polynomial model (i.e., a line with parameters w_0 and w_1) that minimizes the squared loss of this data. Report the model here as an equation. Plot the data with your best-first model and include the plot in your answer (label axes and include an informative caption!).

Solution.

The weights for the best fit are:

$$\mathbf{w}^\top = \begin{bmatrix} 40.924155 & -0.015072 \end{bmatrix}$$

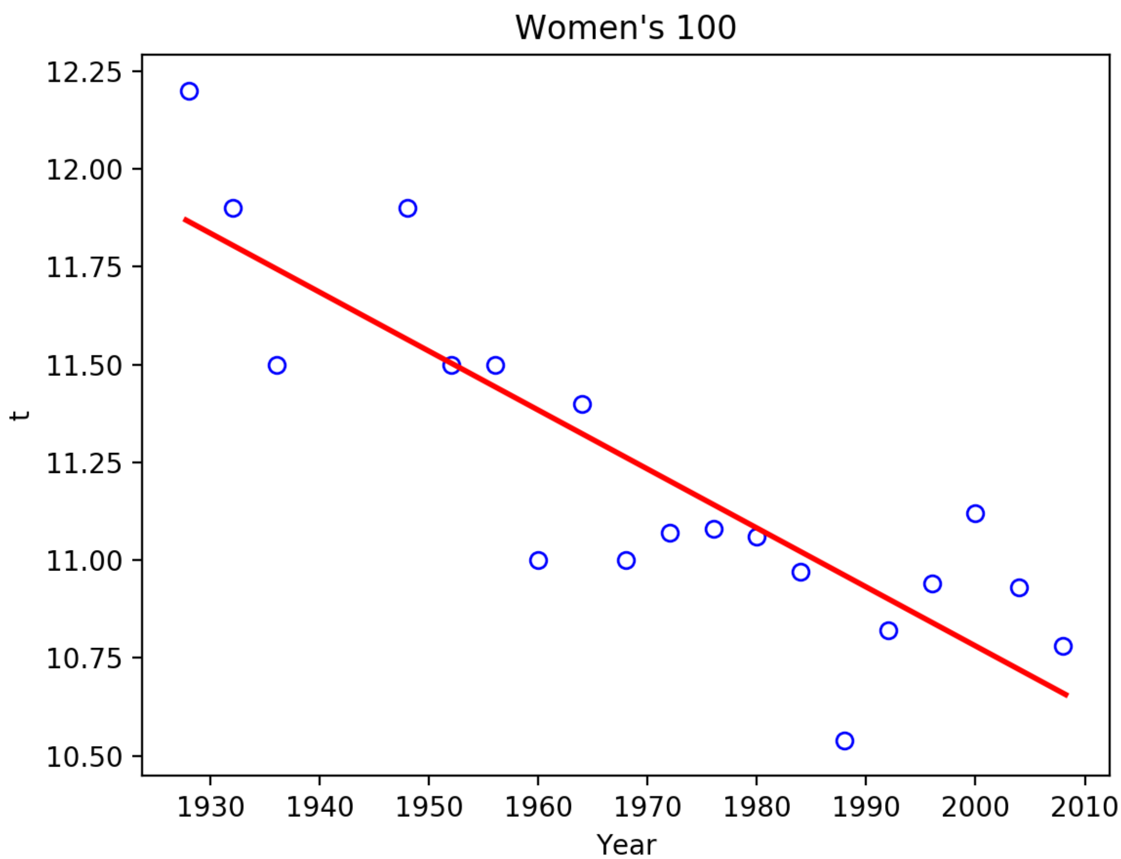

Figure 1: Women's best fit (1st order)

3. [2 point] Adapted from **Exercise 1.9** of FCMA p.36:

   Use your python script from problem 1 to load the data stored in the file `synthdata2018.csv` (in the data folder). Fit a 3rd order polynomial function – $f(x; \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$ – to this data (if you extended the fitpoly_incomplete.py script, then `model_order`= 3). Report the best-fit model parameters as an equation. Plot the data and your model and include the plot in your answer (be sure to include an informative caption to your plot).

Solution.

The weights for the best fit are:

$$\mathbf{w}^\top = \begin{bmatrix} -12.77102013 & 10.19891444 & 5.13913951 & 0.50819653 \end{bmatrix}$$

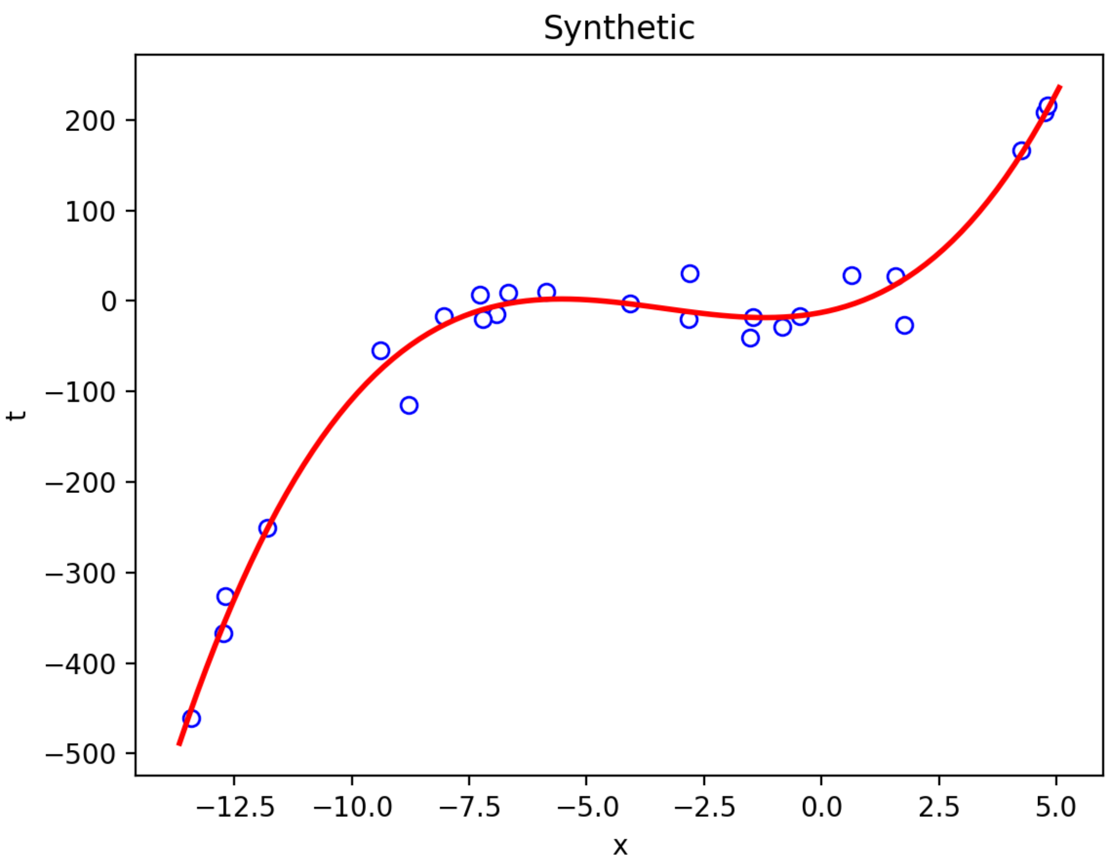

Figure 2: Synthetic best fit (3rd order polynomial)

4. [8 points] Write a script that implements K-fold cross-validation to choose the polynomial order (between orders 0 and 7) with the best predictive error for the `synthdata2018.csv`. The provided script `cv_demo_incomplete.py` implements the synthetic data experiment described in Ch 1 (pp.31-32) of the book; you are welcome to use and adapt any part of this code you like; keep in mind that this script won't successfully execute until you add the general (matrix form) normal equation calculation on line 339. Note also that in the synthetic data experiment in `cv_demo_incomplete.py`, 1000 test data points are generated in addition to the 100 data points used for 10-fold cross-validation in the demo; for this problem (problem 4), you won't have this independent test set, only the data from

synthdata2018.csv on which to perform K-fold cross-validation. Also, for this problem, you will perform 5-fold cross-validation, in addition to Leave-One-Out-CV (LOOCV).

Run your script with 5-fold cross-validation and LOOCV multiple times, each while randomizing the order of the data (see the randomize_data option of the run_cv function). Which model order do the two cross-validation methods predict as the best order for predictive accuracy? Do the two different cross-validation runs always agree?

Report the best-fit model parameters for the best model order according to LOOCV, and plot this model with the data. Include a plot of the training loss and CV-loss for the 8 different (0..7) polynomial model orders for **one example each** of 5-fold cross-validation and LOOCV (i.e., you will include four plots: (1) 5-fold CV and (2) related training loss, (3) LOOCV, and (4) related training loss). You can use the provided plot_cv_results to plot the training loss and CV loss (whether 5-fold or LOOCV) as a pair of plots.

You will submit your script as a stand-alone file called cv.py

Solution.

Both models report order 3 as the best fit. In my runs, they have almost always agreed on this order but sometimes CV would report 4.

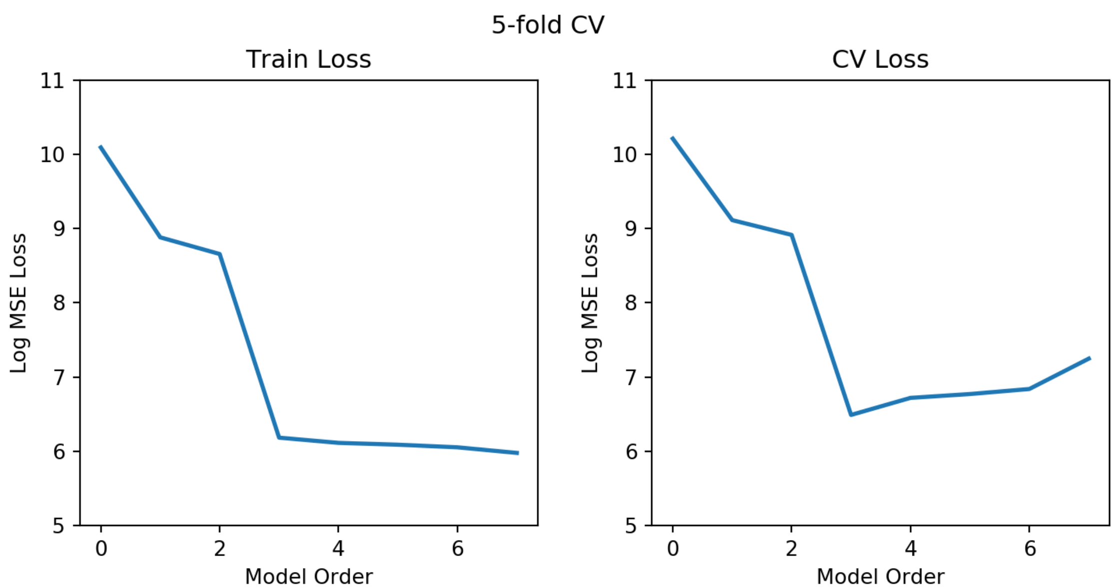

Figure 3: 5-fold cross validation

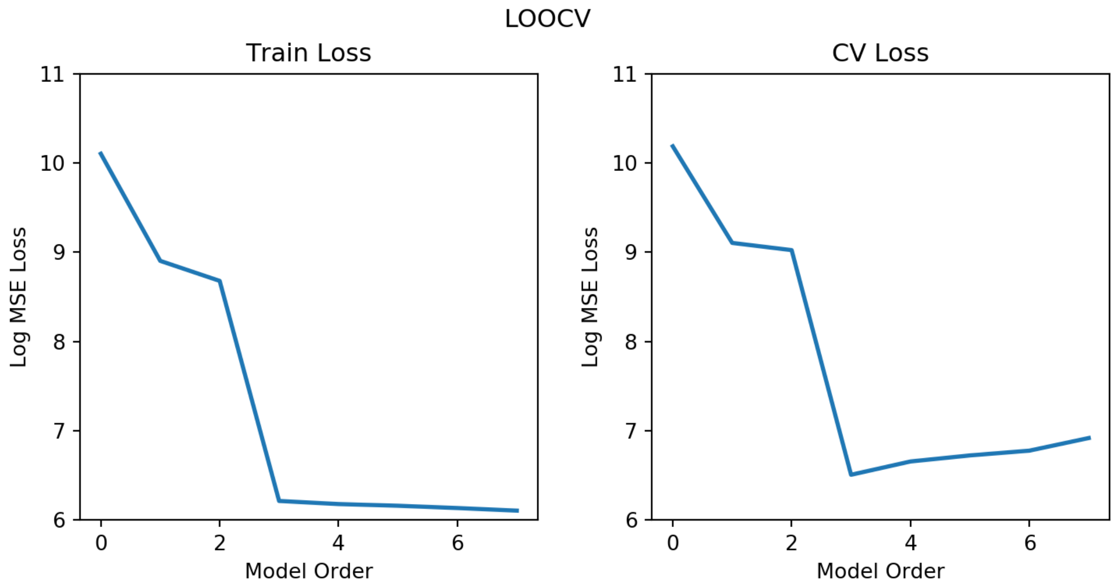

Figure 4: Leave-one-out cross validation

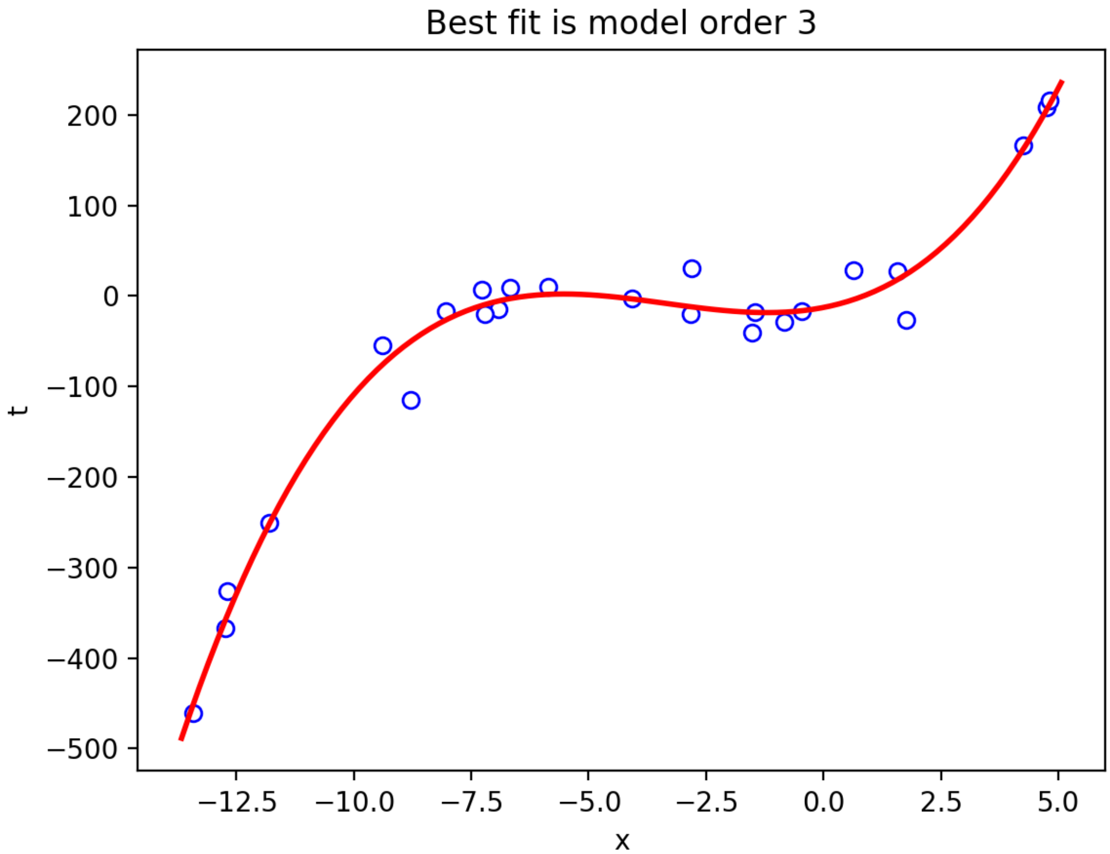

Figure 5: Plot of all data using best order (3rd)

5. [2 points – **Required only for Graduates**] **Exercise 1.10** from FCMA p.36

   Derive the optimal least squares parameter value, $\hat{\mathbf{w}}$, for the total training loss:

$$\mathcal{L} = \sum_{n=1}^{N} \left(t_n - \mathbf{w}^\top \mathbf{x}_n\right)^2$$

   How does the expression compare with that derived from the average (mean) loss? (Hint: Express this loss in the **full** matrix form and derive the normal equation.)

   Solution.

$$
\begin{aligned}
\mathcal{L} &= \sum_{n=1}^{N} \left(t_n - \mathbf{w}^\top \mathbf{x}_n\right)^2 \\
&= (\mathbf{t} - \mathbf{X}\mathbf{w})^\top (\mathbf{t} - \mathbf{X}\mathbf{w}) \\
&= (\mathbf{X}\mathbf{w} - \mathbf{t})^\top (\mathbf{X}\mathbf{w} - \mathbf{t}) \\
&= ((\mathbf{X}\mathbf{w})^\top - \mathbf{t}^\top)(\mathbf{X}\mathbf{w} - \mathbf{t}) \\
&= ((\mathbf{X}\mathbf{w})^\top \mathbf{X}\mathbf{w} - \mathbf{t}^\top \mathbf{X}\mathbf{w} - (\mathbf{X}\mathbf{w})^\top \mathbf{t} + \mathbf{t}^\top \mathbf{t} \\
&= \mathbf{w}^\top \mathbf{X}^\top \mathbf{X}\mathbf{w} - 2\mathbf{w}^\top \mathbf{X}^\top \mathbf{t} + \mathbf{t}^\top \mathbf{t} \\
\frac{\partial \mathcal{L}}{\partial \mathbf{w}} &= 2\mathbf{X}^\top \mathbf{X}\mathbf{w} - 2\mathbf{X}^\top \mathbf{t} = 0 \\
\mathbf{X}^\top \mathbf{X}\mathbf{w} &= \mathbf{X}^\top \mathbf{t} \\
\mathbf{I}\mathbf{w} &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{t}
\end{aligned}
$$

The answer is the same as for the mean, so averaging has no effect on the outcome.

6. [3 points – **Required only for Graduates**] **Exercise 1.11** from FCMA p.36

The following expression is known as the *weighted* average loss:

$$
\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \alpha_n \left(t_n - \mathbf{w}^\top \mathbf{x}_n\right)^2
$$

where the influence of each data point is controlled by its associated parameter. Assuming that each α_n is fixed, derive the optimal least squares parameter value $\hat{\mathbf{w}}$. (Hint: When expressing in the full matrix form, the *alpha*'s become a matrix...)

Solution.

$$
\begin{aligned}
\mathcal{L} &= \frac{1}{N}\sum_{n=1}^{N}\alpha_n\left(t_n - \mathbf{w}^\top\mathbf{x}_n\right)^2 \\
\mathbf{A} &= \begin{bmatrix} \alpha_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \alpha_n \end{bmatrix} \\
\mathcal{L} &= \frac{1}{N}\left((\mathbf{t}-\mathbf{X}\mathbf{w})^\top\mathbf{A}(\mathbf{t}-\mathbf{X}\mathbf{w})\right) \\
&= \frac{1}{N}\left((\mathbf{t}^\top-(\mathbf{X}\mathbf{w})^\top)\mathbf{A}(\mathbf{t}-\mathbf{X}\mathbf{w})\right) \\
&= \frac{1}{N}\left((\mathbf{t}^\top-\mathbf{w}^\top\mathbf{X}^\top)\mathbf{A}(\mathbf{t}-\mathbf{X}\mathbf{w})\right) \\
&= \frac{1}{N}\left((\mathbf{t}^\top\mathbf{A}-\mathbf{w}^\top\mathbf{X}^\top\mathbf{A})(\mathbf{t}-\mathbf{X}\mathbf{w})\right) \\
&= \frac{1}{N}\left(\mathbf{t}^\top\mathbf{A}\mathbf{t}-\mathbf{t}^\top\mathbf{A}\mathbf{X}\mathbf{w}-\mathbf{w}^\top\mathbf{X}^\top\mathbf{A}\mathbf{t}+\mathbf{w}^\top\mathbf{X}^\top\mathbf{A}\mathbf{X}\mathbf{w}\right) \\
&= \frac{1}{N}\left(\mathbf{t}^\top\mathbf{A}\mathbf{t}-2\mathbf{w}^\top\mathbf{X}^\top\mathbf{A}\mathbf{t}+\mathbf{w}^\top\mathbf{X}^\top\mathbf{A}\mathbf{X}\mathbf{w}\right) \\
&= \frac{1}{N}\mathbf{t}^\top\mathbf{A}\mathbf{t}-\frac{2}{N}\mathbf{w}^\top\mathbf{X}^\top\mathbf{A}\mathbf{t}+\frac{1}{N}\mathbf{w}^\top\mathbf{X}^\top\mathbf{A}\mathbf{X}\mathbf{w} \\
\frac{\partial\mathcal{L}}{\partial\mathbf{w}} &= -\frac{2}{N}\mathbf{X}^\top\mathbf{A}\mathbf{t}+\frac{2}{N}\mathbf{X}^\top\mathbf{A}\mathbf{X}\mathbf{w}=0 \\
\mathbf{X}^\top\mathbf{A}\mathbf{X}\mathbf{w} &= \mathbf{X}^\top\mathbf{A}\mathbf{t} \\
\mathbf{I}\mathbf{w} &= (\mathbf{X}^\top\mathbf{A}\mathbf{X})^{-1}(\mathbf{X}^\top\mathbf{A}\mathbf{t})
\end{aligned}
$$