CSCI 5521 Denis Rybkin
Section 2 rybki001@umn.edu

Homework 1
Due Monday, October 05

1.

(i) Consider empirical loss:

$E(w_1, w_0|\mathcal{Z}_{train}) = \frac{1}{N}\sum_{t=1}^{N}(r_t - (w_1 x_t + w_0))^2$

First, we take partial derivative with respect to w_0:

$\frac{\partial}{\partial w_0}\frac{1}{N}\sum_{t=1}^{N}(r_t - (w_1 x_t + w_0))^2 = \frac{1}{N}\sum_{t=1}^{N}\frac{\partial}{\partial w_0}(r_t - (w_1 x_t + w_0))^2 =$

$= \frac{1}{N}\sum_{t=1}^{N} -2(r_t - (w_1 x_t + w_0)) = \frac{-2}{N}\sum_{t=1}^{N}(r_t - (w_1 x_t + w_0))$

Similarly, for w_1:

$\frac{\partial}{\partial w_1}\frac{1}{N}\sum_{t=1}^{N}(r_t - (w_1 x_t + w_0))^2 = \frac{1}{N}\sum_{t=1}^{N}\frac{\partial}{\partial w_1}(r_t - (w_1 x_t + w_0))^2 =$

$= \frac{1}{N}\sum_{t=1}^{N} 2(r_t - (w_1 x_t + w_0))(-x_t) = \frac{-2}{N}\sum_{t=1}^{N} x_t(r_t - (w_1 x_t + w_0))$

Now we set partial derivatives equal to 0:

$\begin{cases} \frac{-2}{N}\sum_{t=1}^{N}(r_t - (w_1 x_t + w_0)) = 0 \\ \frac{-2}{N}\sum_{t=1}^{N} x_t(r_t - (w_1 x_t + w_0)) = 0 \end{cases}$

We can get rid of $\frac{-2}{N}$:

$\begin{cases} \sum_{t=1}^{N}(r_t - (w_1 x_t + w_0)) = 0 \\ \sum_{t=1}^{N} x_t(r_t - (w_1 x_t + w_0)) = 0 \end{cases}$

Solve for w_0:

$\sum_{t=1}^{N}(r_t - (w_1 x_t + w_0)) = 0$

$\sum_{t=1}^{N} r_t - \sum_{t=1}^{N} w_1 x_t - \sum_{t=1}^{N} w_0 = 0$

$\sum_{t=1}^{N} r_t - w_1 \sum_{t=1}^{N} x_t - N w_0 = 0$

$N w_0 = \sum_{t=1}^{N} r_t - w_1 \sum_{t=1}^{N} x_t$

$w_0 = \frac{1}{N} \sum_{t=1}^{N} r_t - \frac{w_1}{N} \sum_{t=1}^{N} x_t$

Or equivalently:

$w_0 = \bar{r} - w_1 \bar{x}$

where $\bar{r} = \frac{1}{N} \sum_{t=1}^{N} r_t$ and $\bar{x} = \frac{1}{N} \sum_{t=1}^{N} x_t$.

Solve for w_1:

$\sum_{t=1}^{N} x_t(r_t - (w_1 x_t + w_0)) = 0$

Substitute $\bar{r} - w_1 \bar{x}$ for w_0:

$\sum_{t=1}^{N} x_t(r_t - (w_1 x_t + \bar{r} - w_1 \bar{x})) = 0$

$\sum_{t=1}^{N} x_t(r_t - \bar{r} - w_1(x_t - \bar{x})) = 0$

$\sum_{t=1}^{N} x_t(r_t - \bar{r}) - \sum_{t=1}^{N} w_1 x_t(x_t - \bar{x}) = 0$

$\sum_{t=1}^{N} x_t(r_t - \bar{r}) = w_1 \sum_{t=1}^{N} x_t(x_t - \bar{x})$

$w_1 = \frac{\sum_{t=1}^{N} x_t(r_t - \bar{r})}{\sum_{t=1}^{N} x_t(x_t - \bar{x})}$

$w_1 = \frac{\sum_{t=1}^{N} (x_t - \bar{x})(r_t - \bar{r})}{\sum_{t=1}^{N} (x_t - \bar{x})^2}$

Hence, we have that the optimal value of w_1 is $\frac{\sum_{t=1}^{N} (x_t - \bar{x})(r_t - \bar{r})}{\sum_{t=1}^{N} (x_t - \bar{x})^2}$ and the optimal value of w_0 is $\bar{r} - w_1 \bar{x}$ where $\bar{r} = \frac{1}{N} \sum_{t=1}^{N} r_t$ and $\bar{x} = \frac{1}{N} \sum_{t=1}^{N} x_t$.

(ii) Consider empirical loss:

$E(v_2, v_1, v_0 | \mathcal{Z}_{train}) = \frac{1}{N} \sum_{t=1}^{N} (r_t - (v_2 x_t^{20} + v_1 x_t^3 + v_0))^2$

First, we take partial derivative with respect to v_0:

$\frac{\partial}{\partial v_0}\frac{1}{N}\sum_{t=1}^{N}(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0))^2 = \frac{1}{N}\sum_{t=1}^{N}\frac{\partial}{\partial v_0}(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0))^2$

$= \frac{1}{N}\sum_{t=1}^{N} -2(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0)) = \frac{-2}{N}\sum_{t=1}^{N}(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0))$

Similarly, for v_1:

$\frac{\partial}{\partial v_1}\frac{1}{N}\sum_{t=1}^{N}(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0))^2 = \frac{1}{N}\sum_{t=1}^{N}\frac{\partial}{\partial v_1}(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0))^2$

$= \frac{1}{N}\sum_{t=1}^{N} -2x_t^3(r_t - (v_2x_t^20 + v_1x_t^3 + v_0)) = \frac{-2}{N}\sum_{t=1}^{N} x_t^3(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0))$

And for v_2:

$\frac{\partial}{\partial v_2}\frac{1}{N}\sum_{t=1}^{N}(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0))^2 = \frac{1}{N}\sum_{t=1}^{N}\frac{\partial}{\partial v_2}(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0))^2$

$= \frac{1}{N}\sum_{t=1}^{N} -2x_t^{20}(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0)) = \frac{-2}{N}\sum_{t=1}^{N} x_t^{20}(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0))$

Now we set partial derivatives equal to 0:

$$\begin{cases} \frac{-2}{N}\sum_{t=1}^{N}(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0)) = 0 \\ \frac{-2}{N}\sum_{t=1}^{N} x_t^3(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0)) = 0 \\ \frac{-2}{N}\sum_{t=1}^{N} x_t^{20}(r_t - (v_2x_t^20 + v_1x_t^3 + v_0)) = 0 \end{cases}$$

We can get rid of $\frac{-2}{N}$:

$$\begin{cases} \sum_{t=1}^{N}(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0)) = 0 \\ \sum_{t=1}^{N} x_t^3(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0)) = 0 \\ \sum_{t=1}^{N} x_t^{20}(r_t - (v_2x_t^{20} + v_1x_t^3 + v_0)) = 0 \end{cases}$$

From here, we have a system of linear equations, where $\mathbf{b} = [0, 0, 0]^T$, $\mathbf{v} = [v_2, v_1, v_0]^T$ and A is a 3 by 3 matrix with non-variable values in it (can be obtained by slightly rewriting the system of linear equations above and taking constant values as fields of the matrix). It can be easily solved the same way as in (i). First, we would solve one of the variables in terms of two others, then substitute first variable into second equation and solve it in terms of remaining variable. For the last one, we will substitute the first and second equations and solve for 0. This way, we'll get the actual value from the third equation. From here, we could solve for second and first variables.

(iii) Consider the following:

$$E(v_2^*, v_1^*, v_0^* | \mathcal{Z}_{train}) \leq E(w_1^*, w_0^* | \mathcal{Z}_{train})$$

Professor Gopher is correct.

From the properties of polynomials, we know that the higher degree polynomial can estimate the curve much more accurately than the lower one. Hence, for arbitrary data points, empirical error $E(v_2^*, v_1^*, v_0^* | \mathcal{Z}_{train})$ should output lower value, because it can fit the data more accurately.

2. Consider A = $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{bmatrix}$

(i)

tr(A) $= 1 + 2 + 9 + 64 = 76$

$$A^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \\ 1 & 8 & 27 & 64 \end{bmatrix}$$

tr(A^T) $= 1 + 2 + 9 + 64 = 76$

$$A^T A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \\ 1 & 8 & 27 & 64 \end{bmatrix} = \begin{bmatrix} 4 & 10 & 30 & 100 \\ 10 & 30 & 100 & 354 \\ 30 & 100 & 354 & 1300 \\ 100 & 354 & 1300 & 4890 \end{bmatrix}$$

$\mathrm{tr}(A^T A) = 4 + 30 + 354 + 4890 = 5278$

$$AA^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \\ 1 & 8 & 27 & 64 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{bmatrix} = \begin{bmatrix} 4 & 15 & 40 & 85 \\ 15 & 85 & 259 & 585 \\ 40 & 259 & 820 & 1885 \\ 85 & 585 & 1885 & 4369 \end{bmatrix}$$

$\mathrm{tr}(AA^T) = 4 + 85 + 820 + 4369 = 5278$

(ii) Geometrically, we can think of $|A|$ as a volume of n-dimensional parallelepiped constructed from vectors of an n-dimensional matrix (area for 2-dimensional matrix). Hence, we can compute a determinant $|A|$ by just computing area/volume of the parallelepiped formed from rows of a matrix.

(iii) The matrix is linearly independent if the only solution of $c_1\mathbf{v_1} + c_2\mathbf{v_2} + c_3\mathbf{v_3} + c_4\mathbf{v_4} = 0$ is $c_1, c_2, c_3, c_4 = 0$. In other words, the matrix is independent if no column is a multiple of the others.

In order to show that rows of A are linearly independent, all we need to do is to compute A's row echelon form. Using numpy's M.rref() on A, we get:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(could be done by hand, but it is tedious and not the point of the class)

Hence,

$$\begin{cases} 1c_1 = 0 \\ 1c_2 = 0 \\ 1c_3 = 0 \\ 1c_4 = 0 \end{cases}$$

Therefore, A is linearly independent.

3.

(i)

- **Code:** Provided in q3i.py and my_cross_val.py files

- **Summary of results:**

Error rates for LinearSVC with Boston50											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.31	0.08	0.06	0.20	0.47	0.20	0.3	0.44	0.24	0.16	0.25	0.13

Error rates for LinearSVC with Boston25											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.41	0.08	0.16	0.25	0.12	0.12	0.10	0.20	0.08	0.16	0.17	0.10

Error rates for LinearSVC with Digits											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.05	0.05	0.08	0.04	0.05	0.05	0.03	0.04	0.04	0.06	0.05	0.01

Error rates for SVC with Boston50											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.25	0.18	0.22	0.24	0.25	0.22	0.30	0.28	0.08	0.22	0.22	0.06

Error rates for SVC with Boston25											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.18	0.16	0.16	0.16	0.10	0.12	0.16	0.08	0.06	0.12	0.13	0.04

Error rates for SVC with Digits											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.01	0.01	0.02	0.00	0.00	0.01	0.00	0.02	0.01	0.01	0.01	0.01

Error rates for Logistic Regression with Boston50											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.10	0.12	0.18	0.08	0.12	0.16	0.20	0.14	0.20	0.16	0.14	0.04

Error rates for Logistic Regression with Boston25											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.06	0.06	0.06	0.04	0.12	0.08	0.12	0.12	0.08	0.08	0.08	0.03

Error rates for Logistic Regression with Digits											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.03	0.02	0.02	0.05	0.04	0.04	0.05	0.04	0.05	0.02	0.04	0.01

(ii)

- **Code:** Provided in q3ii.py and my_train_test.py files

- **Summary of results:**

Error rates for LinearSVC with Boston50											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.31	0.19	0.31	0.17	0.27	0.41	0.24	0.39	0.17	0.4	0.29	0.09

Error rates for LinearSVC with Boston25											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.08	0.14	0.09	0.07	0.46	0.15	0.16	0.15	0.16	0.13	0.16	0.10

Error rates for LinearSVC with Digits											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.07	0.06	0.07	0.05	0.06	0.06	0.05	0.06	0.06	0.06	0.06	0.01

Error rates for SVC with Boston50											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.23	0.24	0.24	0.20	0.28	0.20	0.24	0.17	0.28	0.27	0.24	0.03

Error rates for SVC with Boston25											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.18	0.13	0.16	0.20	0.16	0.23	0.20	0.14	0.11	0.09	0.16	0.04

Error rates for SVC with Digits											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.02	0.01	0.01	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01

Error rates for Logistic Regression with Boston50											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.13	0.16	0.09	0.15	0.14	0.15	0.17	0.14	0.15	0.16	0.14	0.02

Error rates for Logistic Regression with Boston25											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.06	0.13	0.10	0.09	0.07	0.13	0.09	0.11	0.08	0.09	0.10	0.02

Error rates for Logistic Regression with Digits											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.03	0.03	0.04	0.03	0.03	0.05	0.03	0.03	0.05	0.05	0.04	0.01

4.

- **Code:** Provided in q4.py, rand_proj.py and quad_proj.py files

- **Summary of results:**

Error rates for LinearSVC with $\widetilde{X_1}$											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.14	0.07	0.08	0.07	0.08	0.12	0.08	0.13	0.05	0.11	0.09	0.03

Error rates for LinearSVC with $\widetilde{X_2}$											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.01	0.03	0.02	0.01	0.01	0.02	0.03	0.01	0.00	0.00	0.01	0.01

Error rates for SVC with $\widetilde{X_1}$											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.03	0.01	0.04	0.04	0.01	0.00	0.02	0.03	0.02	0.01	0.02	0.01

Error rates for SVC with $\widetilde{X_2}$											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.00	0.01	0.00	0.01	0.01	0.02	0.01	0.02	0.01	0.02	0.01	0.01

Error rates for Logistic Regression with $\widetilde{X_1}$											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.04	0.04	0.07	0.08	0.07	0.06	0.06	0.06	0.09	0.07	0.06	0.01

Error rates for Logistic Regression with $\widetilde{X_2}$											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.02	0.01	0.02	0.00	0.02	0.01	0.00	0.03	0.02	0.01	0.01	0.01