Homework 1
MLE, MAP estimates; Linear and Logistic Regression

CMU 10-701: Machine Learning (Spring 2017)

OUT: Jan 31
DUE: Feb 10, 11:59 PM

START HERE: Instructions

- **Collaboration policy:** Collaboration on solving the homework is allowed, after you have thought about the problems on your own. It is also OK to get clarification (but not solutions) from books or online resources, again after you have thought about the problems on your own. There are two requirements: first, cite your collaborators fully and completely (e.g., "Jane explained to me what is asked in Question 3.4"). Second, write your solution *independently*: close the book and all of your notes, and send collaborators out of the room, so that the solution comes from you only.
- **Submitting your work:** Assignments should be submitted as PDFs using Gradescope unless explicitly stated otherwise. Each derivation/proof should be completed on a separate page. Submissions can be handwritten, but should be labeled and clearly legible. Else, submissions can be written in LaTeX. Upon submission, label each question using the template provided by Gradescope.
- **Programming**: All programming portions of the assignments should be submitted to Gradescope as well. We will not be using this for autograding, but rather for plagiarism detection and to make it simpler to submit code. You may use any language which you like to submit unless the problem states otherwise. There is a separate 'programming assignment' that should allow you to upload your code easily. Code should be uploaded to this separate programming assignment, while visualizations and written answers should still be submitted within the primary Gradescope assignment. In your code, **please make it clear in the comments which are the primary functions to compute the answers to each question.**

Part A: Multiple Choice Questions [7 Points] (Yiting)

- There will be only one right answer. Please explain your choice in one or two sentences.

1. [**4 Points**] For each case listed below, what type of machine learning problem does it belong to?

   (a) Advertisement selection system, which can predict the probability whether a customer will click on an ad or not based on the search history

   (b) U.S post offices use a system to automatically recognize handwriting on the envelope

   (c) Reduce dimensionality using principal components analysis (PCA)

   (d) Trading companies try to predict future stock market based on current market conditions

   (e) Repair a digital image that has been partially damaged

   A. Supervised learning: Classification
   B. Supervised learning: Regression
   C. Unsupervised learning

   Solution: (a):A; (b): A; (c): C; (d) B; (e) A or B or C

2. [**1 Point**] For four statements below, which one is wrong?

   A. In maximum a posterior (MAP) estimate, data overwhelms the prior if we have enough data
   B. There are no parameters in non-parametirc models
   C. $P(X \bigcap Y \bigcap Z) = P(Z|X \bigcap Y)P(Y|X)P(X)$
   D. Compared with parametric models, non-parameter models are flexible, since they don't make strong assumptions

   Solution: A (False if prior is concentrated on one value) or B

3. [**1 Point**] There are about 12% people in U.S. having breast cancer during their lifetime. One patient has a positive result for the medical test. Suppose the sensitivity of this test is 90%, meaning the test will be positive with probability 0.9 if one really has cancer. The false positive is likely to be 2%. Then what is the probability this patient actually having cancer based on Bayes Theorem?

   A. 90% B. 86% C. 12% D. 43%

   Solution: B

4. [**1 Point**] What is the most suitable error function for gradient **descent** using logistic regression?

   A. The negative log-likelihood function
   B. The number of mistakes
   C. The squared error
   D. The log-likelihood function

   Solution: A

Part B, Problem 1: Bias-Variance Decomposition [20 Points] (Calvin and Hao)

Consider a p-dimensional vector $\boldsymbol{x} \in \mathbb{R}^p$ drawn from a Gaussian distribution with an identity covariance matrix $\boldsymbol{\Sigma} = \mathbf{I}_p$ and an unknown mean $\boldsymbol{\mu}$, i.e. $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I}_p)$. Our goal is to evaluate the effectiveness of an estimator $\hat{\boldsymbol{\mu}} = \boldsymbol{f}(\boldsymbol{x})$ of the mean from only a single sample (i.e. $n = 1$) by measuring its mean squared error $\mathbb{E}[\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2]$, where $\|\cdot\|^2$ is the squared Euclidean norm and the expectation is taken over the data generating distribution.

Note that for any estimator $\hat{\boldsymbol{\theta}}$ of a parameter vector $\boldsymbol{\theta}$, its mean squared error can be decomposed as:

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right\|^2 \right] = \left\|\text{Bias}[\hat{\boldsymbol{\theta}}]\right\|^2 + \text{Trace}(\text{Var}[\hat{\boldsymbol{\theta}}]), \text{ where:}$$

$$\text{Bias}[\hat{\boldsymbol{\theta}}] = \mathbb{E}[\hat{\boldsymbol{\theta}}] - \boldsymbol{\theta} \quad \text{and} \quad (\text{Var}[\hat{\boldsymbol{\theta}}])_{j,j} = \text{Var}[\hat{\theta}_j] = \mathbb{E}[(\hat{\theta}_j - \mathbb{E}[\hat{\theta}_j])^2]$$

Here, $\text{Trace}(\cdot)$ denotes the sum of the diagonal elements of a square matrix, $(\text{Var}[\hat{\boldsymbol{\theta}}])_{j,j}$ denotes the jth diagonal element of $\text{Var}[\hat{\boldsymbol{\theta}}]$, and $\hat{\theta}_j$ denotes the jth element of $\hat{\boldsymbol{\theta}}$.

First, we derive this identity:

$$\begin{aligned}
\mathbb{E}[\left\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right\|^2] &= \sum_{j=1}^{p} \mathbb{E}[(\hat{\theta}_j - \theta_j)^2] \\
\mathbb{E}[(\hat{\theta}_j - \theta_j)^2] &= \mathbb{E}\left[\left(\hat{\theta}_j - \mathbb{E}[\hat{\theta}_j] + \mathbb{E}[\hat{\theta}_j] - \theta_j\right)^2\right] \\
&= \mathbb{E}\left[\left(\left(\hat{\theta}_j - \mathbb{E}[\hat{\theta}_j]\right) + \left(\mathbb{E}[\hat{\theta}_j] - \theta_j\right)\right)^2\right] \\
&= \mathbb{E}\left[\left(\mathbb{E}[\hat{\theta}_j] - \theta_j\right)^2\right] + \mathbb{E}\left[\left(\hat{\theta}_j - \mathbb{E}[\hat{\theta}_j]\right)^2\right] + 2\mathbb{E}\left[\left(\hat{\theta}_j - \mathbb{E}[\hat{\theta}_j]\right)\left(\mathbb{E}[\hat{\theta}_j] - \theta_j\right)\right] \\
&= \mathbb{E}\left[\text{Bias}[\hat{\theta}_j]^2\right] + \text{Var}[\hat{\theta}_j] + 2\left(\mathbb{E}[\hat{\theta}_j\mathbb{E}[\hat{\theta}_j]] - \mathbb{E}[\mathbb{E}[\hat{\theta}_j]^2] - \mathbb{E}[\hat{\theta}_j\theta_j] + \mathbb{E}[\mathbb{E}[\hat{\theta}_j]\theta_j]\right) \\
&= \text{Bias}[\hat{\theta}_j]^2 + \text{Var}[\hat{\theta}_j] + 2\left(\mathbb{E}[\hat{\theta}_j]^2 - \mathbb{E}[\hat{\theta}_j]^2 - \theta_j\mathbb{E}[\hat{\theta}_j] + \theta_j\mathbb{E}[\hat{\theta}_j]\right) \\
&= \text{Bias}[\hat{\theta}_j]^2 + \text{Var}[\hat{\theta}_j] \\
\mathbb{E}[\left\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right\|^2] &= \sum_{j=1}^{p}\left(\text{Bias}[\hat{\theta}_j]^2 + \text{Var}[\hat{\theta}_j]\right) \\
&= \left\|\text{Bias}[\hat{\boldsymbol{\theta}}]\right\|^2 + \text{Trace}(\text{Var}[\hat{\boldsymbol{\theta}}])
\end{aligned}$$

1. **[4 Points]** Derive the maximum likelihood estimator:

$$\hat{\boldsymbol{\mu}}_{\text{MLE}} = \arg\max_{\boldsymbol{\mu}} P(\boldsymbol{x}; \boldsymbol{\mu}).$$

   What is its mean squared error?

   Solution:

$$
\begin{aligned}
\hat{\boldsymbol{\mu}}_{\text{MLE}} &= \arg\max_{\boldsymbol{\mu}} P(\boldsymbol{x};\boldsymbol{\mu}) \\
&= \arg\min_{\boldsymbol{\mu}} -\log P(\boldsymbol{x};\boldsymbol{\mu}) \\
&= \arg\min_{\boldsymbol{\mu}} -\log\left((2\pi)^{-p/2}\exp\left(-\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{\mu}\|^2\right)\right) \\
&= \arg\min_{\boldsymbol{\mu}} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{\mu}\|^2 \\
0 &= \frac{d}{d\boldsymbol{\mu}}\left(\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{\mu}\|^2\right)\Big|_{\boldsymbol{\mu}=\hat{\boldsymbol{\mu}}_{\text{MLE}}} \\
0 &= \hat{\boldsymbol{\mu}}_{\text{MLE}} - \boldsymbol{x} \\
\hat{\boldsymbol{\mu}}_{\text{MLE}} &= \boldsymbol{x}
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}[\|\hat{\boldsymbol{\mu}}_{\text{MLE}}-\boldsymbol{\mu}\|^2] &= \mathbb{E}[\|\boldsymbol{x}-\boldsymbol{\mu}\|^2] \\
&= \|\text{Bias}(\boldsymbol{x})\|^2 + \text{Trace}(\text{Var}(\boldsymbol{x})) \\
&= \|\mathbb{E}(\boldsymbol{x})-\boldsymbol{\mu}\|^2 + \text{Trace}(\mathbf{I}_p) \\
&= \|\boldsymbol{\mu}-\boldsymbol{\mu}\|^2 + p \\
&= p
\end{aligned}
$$

2. **[4 Points]** Derive the ℓ_2-regularized maximum likelihood estimator:

$$\hat{\boldsymbol{\mu}}_{\text{RMLE}} = \arg\max_{\boldsymbol{\mu}} \log P(\boldsymbol{x};\boldsymbol{\mu}) - \lambda\|\boldsymbol{\mu}\|^2 .$$

What is its mean squared error?

Solution:

$$
\begin{aligned}
\hat{\boldsymbol{\mu}}_{\text{RMLE}} &= \arg\min_{\boldsymbol{\mu}} -\log P(\boldsymbol{x};\boldsymbol{\mu}) + \lambda\|\boldsymbol{\mu}\|^2 \\
&= \arg\min_{\boldsymbol{\mu}} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{\mu}\|^2 + \lambda\|\boldsymbol{\mu}\|^2 \\
0 &= \frac{d}{d\boldsymbol{\mu}}\left(\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{\mu}\|^2 + \lambda\|\boldsymbol{\mu}\|^2\right)\Big|_{\boldsymbol{\mu}=\hat{\boldsymbol{\mu}}_{\text{MLE}}} \\
0 &= \hat{\boldsymbol{\mu}}_{\text{MLE}} + 2\lambda\hat{\boldsymbol{\mu}}_{\text{MLE}} - \boldsymbol{x} \\
(1+2\lambda)\,\hat{\boldsymbol{\mu}}_{\text{MLE}} &= \boldsymbol{x} \\
\hat{\boldsymbol{\mu}}_{\text{MLE}} &= \frac{1}{1+2\lambda}\boldsymbol{x} \\
&= c\boldsymbol{x}, \quad \text{where } c = \frac{1}{1+2\lambda}
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}[\|\hat{\boldsymbol{\mu}}_{\text{MLE}}-\boldsymbol{\mu}\|^2] &= \mathbb{E}[\|c\boldsymbol{x}-\boldsymbol{\mu}\|^2] \\
&= \|\text{Bias}(c\boldsymbol{x})\|^2 + \text{Trace}(\text{Var}(c\boldsymbol{x})) \\
&= \|c\mathbb{E}(\boldsymbol{x})-\boldsymbol{\mu}\|^2 + c^2\text{Trace}(\mathbf{I}_p) \\
&= (c-1)^2\|\boldsymbol{\mu}\|^2 + c^2 p \\
&= \left(\frac{1}{1+2\lambda}-1\right)^2\|\boldsymbol{\mu}\|^2 + \left(\frac{1}{1+2\lambda}\right)^2 p
\end{aligned}
$$

3. **[4 Points]** Consider an estimator of the form $\hat{\boldsymbol{\mu}}_{\text{SCALE}} = c\boldsymbol{x}$ where $c \in \mathbb{R}$ is a constant scaling factor. Find the value c^* that minimizes its mean squared error:

$$c^* = \arg\min_c \mathbb{E}[\|c\boldsymbol{x} - \boldsymbol{\mu}\|^2].$$

What is the corresponding minimum mean squared error?

Solution:

$$\begin{aligned}
c^* &= \arg\min_c \mathbb{E}[\|c\boldsymbol{x} - \boldsymbol{\mu}\|^2] \\
&= \arg\min_c \left[\|\text{Bias}(c\boldsymbol{x})\|^2 + \text{Trace}(\text{Var}(c\boldsymbol{x}))\right] \\
&= \arg\min_c \left[\|c\mathbb{E}(\boldsymbol{x}) - \boldsymbol{\mu}\|^2 + c^2\text{Trace}(\mathbf{I}_p)\right] \\
&= \arg\min_c \left[(c-1)^2 \|\boldsymbol{\mu}\|^2 + c^2 p\right] \\
0 &= \frac{d}{dc}\left((c-1)^2 \|\boldsymbol{\mu}\|^2 + c^2 p\right)\Big|_{c=c^*} \\
&= (2c^* - 2)\|\boldsymbol{\mu}\|^2 + 2c^* p \\
&= c^*\left(\|\boldsymbol{\mu}\|^2 + p\right) - \|\boldsymbol{\mu}\|^2 \\
c^* &= \frac{\|\boldsymbol{\mu}\|^2}{\|\boldsymbol{\mu}\|^2 + p}
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[\|c^*\boldsymbol{x} - \boldsymbol{\mu}\|^2] &= (c^* - 1)^2 \|\boldsymbol{\mu}\|^2 + c^{*2} p \\
&= \left(\frac{\|\boldsymbol{\mu}\|^2}{\|\boldsymbol{\mu}\|^2 + p} - 1\right)^2 \|\boldsymbol{\mu}\|^2 + \left(\frac{\|\boldsymbol{\mu}\|^2}{\|\boldsymbol{\mu}\|^2 + p}\right)^2 p \\
&= \left(-\frac{p}{\|\boldsymbol{\mu}\|^2 + p}\right)^2 \|\boldsymbol{\mu}\|^2 + \left(\frac{\|\boldsymbol{\mu}\|^2}{\|\boldsymbol{\mu}\|^2 + p}\right)^2 p \\
&= \frac{p^2\|\boldsymbol{\mu}\|^2}{\left(\|\boldsymbol{\mu}\|^2 + p\right)^2} + \frac{\|\boldsymbol{\mu}\|^4 p}{\left(\|\boldsymbol{\mu}\|^2 + p\right)^2} \\
&= \frac{p\|\boldsymbol{\mu}\|^2\left(\|\boldsymbol{\mu}\|^2 + p\right)}{\left(\|\boldsymbol{\mu}\|^2 + p\right)^2} \\
&= \frac{p\|\boldsymbol{\mu}\|^2}{\|\boldsymbol{\mu}\|^2 + p}
\end{aligned}$$

4. Consider the James-Stein estimator:

$$\hat{\boldsymbol{\mu}}_{\text{JS}} = \left(1 - \frac{p-2}{\|\boldsymbol{x}\|^2}\right)\boldsymbol{x}.$$

Note that $\hat{\boldsymbol{\mu}}_{\text{JS}}$ can be written as $\boldsymbol{x} - \boldsymbol{g}(\boldsymbol{x})$ where $\boldsymbol{g}(\boldsymbol{x}) = \frac{p-2}{\|\boldsymbol{x}\|^2}\boldsymbol{x}$. This allows us to separate the mean squared error into three parts:

$$\begin{aligned}
\mathbb{E}[\|\hat{\boldsymbol{\mu}}_{\text{JS}} - \boldsymbol{\mu}\|^2] &= \mathbb{E}[\|\boldsymbol{x} - \boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{\mu}\|^2] \\
&= \mathbb{E}[\boldsymbol{x}^\intercal\boldsymbol{x} - 2\boldsymbol{x}^\intercal\boldsymbol{\mu} + \boldsymbol{\mu}^\intercal\boldsymbol{\mu} + \boldsymbol{g}(\boldsymbol{x})^\intercal\boldsymbol{g}(\boldsymbol{x}) - 2\boldsymbol{x}^\intercal\boldsymbol{g}(\boldsymbol{x}) + 2\boldsymbol{\mu}^\intercal\boldsymbol{g}(\boldsymbol{x})] \\
&= \mathbb{E}[\|\boldsymbol{x} - \boldsymbol{\mu}\|^2] + \mathbb{E}[\|\boldsymbol{g}(\boldsymbol{x})\|^2] - 2\mathbb{E}[(\boldsymbol{x} - \boldsymbol{\mu})^\intercal\boldsymbol{g}(\boldsymbol{x})]
\end{aligned}$$

Furthermore, from Stein's lemma, we know that:

$$\mathbb{E}[(\boldsymbol{x}-\boldsymbol{\mu})^\intercal \boldsymbol{g}(\boldsymbol{x})] = \mathbb{E}\Big[\sum_{j=1}^{p} \frac{\partial}{\partial x_j} g_j(\boldsymbol{x})\Big]$$

where x_j is the jth element of $\boldsymbol{x}$ and $g_j(\boldsymbol{x})$ is the jth element of $\boldsymbol{g}(\boldsymbol{x})$.

(a) **[1 Point]** Find $\mathbb{E}[\|\boldsymbol{x}-\boldsymbol{\mu}\|^2]$.

Solution:

$$\mathbb{E}[\|\boldsymbol{x}-\boldsymbol{\mu}\|^2] = p \text{ (from question 1)}$$

(b) **[1 Point]** Find $\mathbb{E}[\|\boldsymbol{g}(\boldsymbol{x})\|^2]$. (Hint: your answer will include $\mathbb{E}[\|\boldsymbol{x}\|^{-2}]$.)

Solution:

$$\begin{aligned}
\mathbb{E}[\|\boldsymbol{g}(\boldsymbol{x})\|^2] &= \mathbb{E}\left[\left\|\frac{p-2}{\|\boldsymbol{x}\|^2}\boldsymbol{x}\right\|^2\right] \\
&= (p-2)^2\,\mathbb{E}\left[\left(\frac{1}{\|\boldsymbol{x}\|^2}\boldsymbol{x}\right)^\intercal\left(\frac{1}{\|\boldsymbol{x}\|^2}\boldsymbol{x}\right)\right] \\
&= (p-2)^2\,\mathbb{E}\left[\frac{\boldsymbol{x}^\intercal\boldsymbol{x}}{\|\boldsymbol{x}\|^4}\right] \\
&= (p-2)^2\,\mathbb{E}\left[\frac{\|\boldsymbol{x}\|^2}{\|\boldsymbol{x}\|^4}\right] \\
&= (p-2)^2\,\mathbb{E}[\|\boldsymbol{x}\|^{-2}]
\end{aligned}$$

(c) **[1 Point]** Show that:

$$\frac{\partial}{\partial x_j} g_j(\boldsymbol{x}) = (p-2)\frac{\|\boldsymbol{x}\|^2 - 2x_j^2}{\|\boldsymbol{x}\|^4}$$

Solution:

$$\begin{aligned}
\frac{\partial}{\partial x_j} g_j(\boldsymbol{x}) &= \frac{\partial}{\partial x_j}\left[\frac{(p-2)\,x_j}{\boldsymbol{x}^\intercal\boldsymbol{x}}\right] \\
&= (p-2)\,\frac{\partial}{\partial x_j}\left[\frac{x_j}{x_j^2+\sum_{k\neq j} x_k^2}\right] \\
&= (p-2)\,\frac{\left(\frac{\partial}{\partial x_j}x_j\right)\boldsymbol{x}^\intercal\boldsymbol{x} - \left(\frac{\partial}{\partial x_j}x_j^2\right)x_j}{(\boldsymbol{x}^\intercal\boldsymbol{x})^2} \\
&= (p-2)\,\frac{\|\boldsymbol{x}\|^2 - 2x_j^2}{\|\boldsymbol{x}\|^4}
\end{aligned}$$

(d) **[1 Point]** What is the resulting mean squared error. (Hint: your answer will include $\mathbb{E}[\|\boldsymbol{x}\|^{-2}]$.)

Solution:

$$
\begin{aligned}
\mathbb{E}[\|\hat{\boldsymbol{\mu}}_{\text{JS}} - \boldsymbol{\mu}\|^2] &= \mathbb{E}[\|\boldsymbol{x} - \boldsymbol{\mu}\|^2] + \mathbb{E}[\|\boldsymbol{g}(\boldsymbol{x})\|^2] - 2\mathbb{E}[(\boldsymbol{x} - \boldsymbol{\mu})^\intercal \boldsymbol{g}(\boldsymbol{x})] \\
&= p + (p-2)^2\, \mathbb{E}[\|\boldsymbol{x}\|^{-2}] - 2\mathbb{E}[\sum_{j=1}^{p} \frac{\partial}{\partial x_j} g_j(\boldsymbol{x})] \\
&= p + (p-2)^2\, \mathbb{E}[\|\boldsymbol{x}\|^{-2}] - 2\,(p-2)\,\mathbb{E}\left[\sum_{j=1}^{p} \frac{\|\boldsymbol{x}\|^2 - 2x_j^2}{\|\boldsymbol{x}\|^4}\right] \\
&= p + (p-2)^2\, \mathbb{E}[\|\boldsymbol{x}\|^{-2}] - 2\,(p-2)\,\mathbb{E}\left[\frac{1}{\|\boldsymbol{x}\|^4}\left(p\,\|\boldsymbol{x}\|^2 - 2\sum_{j=1}^{p} x_j^2\right)\right] \\
&= p + (p-2)^2\, \mathbb{E}[\|\boldsymbol{x}\|^{-2}] - 2\,(p-2)\,\mathbb{E}\left[\frac{p\,\|\boldsymbol{x}\|^2 - 2\,\|\boldsymbol{x}\|^2}{\|\boldsymbol{x}\|^4}\right] \\
&= p + (p-2)^2\, \mathbb{E}[\|\boldsymbol{x}\|^{-2}] - 2\,(p-2)\,\mathbb{E}\left[\frac{p-2}{\|\boldsymbol{x}\|^2}\right] \\
&= p + (p-2)^2\, \mathbb{E}[\|\boldsymbol{x}\|^{-2}] - 2\,(p-2)^2\, \mathbb{E}[\|\boldsymbol{x}\|^{-2}] \\
&= p - (p-2)^2\, \mathbb{E}[\|\boldsymbol{x}\|^{-2}]
\end{aligned}
$$

5. **[4 Points]** Qualitatively compare these estimators, noting any similarities between them. How does regularization affect an estimator's bias and variance? Which estimator would you choose to approximate $\boldsymbol{\mu}$ from real data about which you have no prior knowledge? How does the data dimensionality p affect your answer?

   Solution:

   The MLE is the standard, unbiased sample average, while the other estimators all multiply this estimator by a scaling factor. The optimal scaling factor c^* includes information about the unknown mean $\boldsymbol{\mu}$, so it cannot be used directly in a practical estimator. However, its value is less than 1 indicating that the mean squared error can be improved by shrinking the MLE towards zero, which is equivalent to ℓ_2 regularization with some $\lambda > 0$. On the other hand, the James-Stein estimator only uses known information about $\boldsymbol{x}$.

   Furthermore, since $\mathbb{E}[\|\boldsymbol{x}\|^{-2}] > 0$, its mean squared error is always less than that of the MLE when $p > 2$. When $p = 2$, the James-Stein estimator reduces to the MLE, and when $p = 1$, the MSE of $\hat{\mu}_{\text{JS}} = x + \frac{1}{x}$ is always greater than that of the MLE. Therefore, if $p \geq 3$, choose the James-Stein estimator since it gives a lower MSE, and if $p \leq 2$, choose the MLE since it gives a lower MSE.

   This result is counter-intuitive; even though there is no prior knowledge about any of the unknown variables μ_j (which could be completely independent of eachother), regularization using information about all of them can reduce the MSE. But only when estimating more than two things at a time...

Part B, Problem 2: Linear Regression [18 Points] (Adams and Weixiang)

Suppose we observe N data pairs $\{(x_i, y_i)\}_{i=1}^N$, where y_i is generated by the following rule:

$$y_i = x_i^\top \beta + \epsilon_i,$$

where $x_i, \beta \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, and ϵ_i is an i.i.d random noise drawn from the Gaussian Distribution:

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

with a known constant σ. We further denote $Y = [y_1, y_2, ..., y_N]^\top$ and $X = [x_1, x_2, ..., x_N]^\top$.

Now, we are interested in estimating β from the observed data.

1. **[3 Points]** Derive the likelihood function $\mathcal{L}(\beta)$.
   Solution:

   Since $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, we have $y_i \sim \mathcal{N}(x_i^\top \beta, \sigma^2)$. So the likelihood function is

   $$\mathcal{L}(\beta) = p(X, Y|\beta) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x_i^\top \beta - y_i)^2}{2\sigma^2}\right\} = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N \exp\left\{-\frac{\|Y - X\beta\|_2^2}{2\sigma^2}\right\}.$$

2. **[5 Points]** Show that the MLE estimator $\hat{\beta}_{\text{ml}}$ of β is equivalent to the solution of the following linear regression problem:

   $$\min_\beta \frac{1}{2}\|Y - X\beta\|_2^2 \tag{1}$$

   Solution: Maximizing $\log \mathcal{L}(\beta)$ over β is equivalent to maximizing $-\frac{\|Y - X\beta\|_2^2}{2\sigma^2}$, which is further equivalent to minimizing $\frac{1}{2}\|Y - X\beta\|_2^2$. We ignore some constants of the optimization problem as they do not change the solution.

3. **[5 Points]** Now we suppose β is not a deterministic parameter, but a random variable having a Gaussian prior distribution:

   $$p(\beta) \sim \mathcal{N}(0, \frac{\sigma^2}{2\lambda} I),$$

   where I is a $d \times d$ identity matrix and $\lambda > 0$ is a known parameter. Show that the MAP estimation $\hat{\beta}_{\text{map}}$ of β is equivalent to the solution of the following ridge regression problem:

   $$\min_\beta \frac{1}{2}\|Y - X\beta\|_2^2 + \lambda\|\beta\|_2^2 \tag{2}$$

   Solution: The posterior of β is

   $$p(\beta|X, Y) = p(\beta)p(X, Y|\beta) \propto \exp\left\{-\frac{\|Y - X\beta\|_2^2}{2\sigma^2} - \frac{\lambda\|\beta\|_2^2}{\sigma^2}\right\}.$$

   So maximizing $p(\beta|X, Y)$ over β is equivalent to minimizing $\frac{1}{2}\|Y - X\beta\|_2^2 + \lambda\|\beta\|_2^2$.

4. **[5 Points]** Refer to the closed form solutions of (1) and (2) in the lecture slides, what might be an issue of $\hat{\beta}_{\text{ml}}$ if $d \gg N$? How can $\hat{\beta}_{\text{map}}$ possibly address it?
   Solution:

   We know $\hat{\beta}_{\text{ml}} = (X^\top X)^{-1} X^\top Y$, $\hat{\beta}_{\text{map}} = (X^\top X + \lambda I)^{-1} X^\top Y$. If $d \gg N$, $X^\top X \in \mathbb{R}^{d \times d}$ might not be invertible, as $\text{Rank}(X^\top X) \leq N \ll d$. In that case, we cannot compute $\hat{\beta}_{\text{ml}}$. On the other hand, any $\lambda > 0$ will ensure $(X^\top X + \lambda I)$ is invertible. In fact, we can show that $v^\top (X^\top X + \lambda I)v = \|Xv\|_2^2 + \lambda\|v\|_2^2 > 0$ for any none-zero vector v, so $(X^\top X + \lambda I)$ is positive definite and thus invertible. Now we are able to compute $\hat{\beta}_{\text{map}}$.

Part B, Problem 3: MLE, MAP and Logistic Regression

(Prakhar Naval and Yichong Xu)

We learnt about Maximum Likelihood estimation in class. For a fixed set of data and underlying statistical model, the method of maximum likelihood selects the set of values of the model parameters that maximises the likelihood function.

In this problem, we will look at two different ways of estimating parameters in a probability distribution. Suppose we observe n iid random variables $X_1, \ldots, X_n$, drawn from a distribution with parameter θ. That is, for each X_i and a natural number k,

$$P(X_i = k) = (1-\theta)^k \theta$$

Given some observed values of X_1 to X_n, we want to estimate the value of θ.

3.1. Maximum Likelihood Estimation [9 pts]

The first kind of estimator for θ we will consider is the Maximum Likelihood Estimator (MLE). The probability of observing given data is called the likelihood of the data, and the function that gives the likelihood for a given parameter $\hat{\theta}$ (which may or may not be equal to the true parameter θ) is called the likelihood function, written as $L(\hat{\theta})$. When we use MLE, we estimate θ by choosing the $\hat{\theta}$ that maximizes the likelihood.

$$\hat{\theta}^{\mathrm{MLE}} = \operatorname*{argmax}_{\hat{\theta}} L(\hat{\theta})$$

It is often convenient to deal with the log-likelihood ($\ell(\hat{\theta}) = \log L(\hat{\theta})$) instead, and since log is an increasing function, the argmax also applies in the log space:

$$\hat{\theta}^{\mathrm{MLE}} = \operatorname*{argmax}_{\hat{\theta}} \ell(\hat{\theta})$$

1. **[4 points]** Given a dataset $\mathcal{D}$, containing observations $\{X_1 = k_1, X_2 = k_2, \ldots, X_n = k_n\}$, write an expression for $\ell(\hat{\theta})$ as a function of $\mathcal{D}$ and $\hat{\theta}$. How does the order of the variables affect the function?
   Solution:
   The likelihood function is given by,

$$L(\hat{\theta}) = \prod_{i=1}^{n} P(X_i = k_i)$$

$$\implies L(\hat{\theta}) = ((1-\hat{\theta})^{X_1}\hat{\theta}) \times ((1-\hat{\theta})^{X_2}\hat{\theta})........((1-\hat{\theta})^{X_n}\hat{\theta})$$

$$= (1-\hat{\theta})^{(\sum_{i=1}^{n} X_i)} \times \hat{\theta}^n$$

   Taking log on both sides, the log-likelihood is given by,

$$\ell(\hat{\theta}) = \log L(\hat{\theta}) = (\sum_{i=1}^{n} X_i) \times \log(1-\hat{\theta}) + n \times \log(\hat{\theta})$$

   We know that, the mean/average of dataset $\mathcal{D}$ is given by,

$$\bar{\mathcal{D}} = \frac{\sum_{i=1}^{n} X_i}{n}$$

$$\implies \sum_{i=1}^{n} X_i = n\bar{\mathcal{D}}$$

$$\ell(\hat{\theta}) = n\bar{\mathcal{D}} \times \log(1-\hat{\theta}) + n \times \log(\hat{\theta})$$

   which is the required expression for $\ell(\hat{\theta})$.
   The order of the variables considered will not have any effect on the function $\ell(\hat{\theta})$.

2. **[5 points]** Derive an expression for the maximum likelihood estimate.
   Solution:
   Differentiating the expression obtained above and equating it to zero gives us,

$$\frac{\partial(n\bar{\mathcal{D}} \times \log(1-\hat{\theta}) + n \times \log(\hat{\theta}))}{\partial(\hat{\theta})} = 0$$

$$\implies \hat{\theta} = \frac{n}{n + n\bar{\mathcal{D}}}$$

3.2. Maximum a Posteriori Estimation [11 pts]

Now we assume that we have some prior knowledge about the true parameter θ. We express it by treating θ itself as a random variable and definining a prior probability distribution over it. Precisely, we suppose that the data $X_1, \ldots, X_n$ are drawn as follows:

- θ is drawn from the prior probability distribution
- Then $X_1, \ldots, X_n$ are drawn independently from a Geometric distribution with θ as the parameter.

Now both X_i and θ are random variables, and they have a joint probability distribution. We now estimate θ as follows

$$\hat{\theta}^{\text{MAP}} = \underset{\hat{\theta}}{\text{argmax}}\, P(\theta = \hat{\theta} | X_1, \ldots, X_n)$$

This is called Maximum a Posteriori (MAP) estimation. Using Bayes rule, we can rewrite the posterior probability as follows.

$$P(\theta = \hat{\theta}) = \frac{P(X_i, \ldots, X_n | \theta = \hat{\theta}) P(\theta = \hat{\theta})}{P(X_1, \ldots, X_n)}$$

Applying this to the MAP estimate, we get the following expression. Notice that we can ignore the denominator since it is not a function of $\hat{\theta}$.

$$\begin{aligned}
\hat{\theta}^{\text{MAP}} &= \underset{\hat{\theta}}{\text{argmax}}\, P(X_1, \ldots, X_n | \theta = \hat{\theta}) P(\theta = \hat{\theta}) \\
&= \underset{\hat{\theta}}{\text{argmax}}\, L(\hat{\theta}) P(\theta = \hat{\theta}) \\
&= \underset{\hat{\theta}}{\text{argmax}} \left(\ell(\hat{\theta}) + \log P(\theta = \hat{\theta}) \right)
\end{aligned}$$

Thus, the MAP estimator maximizes the sum of the log-likelihood and the log-probability of the prior distribution on θ. When the prior is a continuous distribution with density function p, we have

$$\hat{\theta}^{\text{MAP}} = \underset{\hat{\theta}}{\text{argmax}} \left(\ell(\hat{\theta}) + \log p(\hat{\theta}) \right)$$

For this problem, we will use the Beta distribution (a popular choice when the data distribution is Geometric or Bernoulli) as the prior, and the density function is given by

$$p(\hat{\theta}) = \frac{\hat{\theta}^{\alpha-1}(1-\hat{\theta})^{\beta-1}}{B(\alpha, \beta)}$$

where $B(\alpha, \beta)$ is the beta function.

4. **[5 points]** Derive a close form expression for the maximum a posteriori estimate. (hint: If x^* maximizes f, $f'(x^*) = 0$).
   Solution:
   From question 3.1.1, we know that,

$$\ell(\hat{\theta}) = n\bar{\mathcal{D}} \times \log(1-\hat{\theta}) + n \times \log(\hat{\theta})$$

   Adding the log posterior function, we get

$$\implies n\bar{\mathcal{D}} \times \log(1-\hat{\theta}) + n \times \log(\hat{\theta}) + \log(p(\hat{\theta}))$$

$$\implies n\bar{\mathcal{D}} \times \log(1-\hat{\theta}) + n \times \log(\hat{\theta}) + \log(\frac{\hat{\theta}^{\alpha-1}(1-\hat{\theta})^{\beta-1}}{B(\alpha,\beta)})$$

$$\implies n\bar{\mathcal{D}} \times \log(1-\hat{\theta}) + n \times \log(\hat{\theta}) + (\alpha-1) \times \log(\hat{\theta}) + (\beta-1) \times \log(1-\hat{\theta}) - log(B(\alpha,\beta))$$

$$\implies (n\bar{\mathcal{D}} + \beta - 1) \times \log(1-\hat{\theta}) + (n+\alpha-1) \times \log(\hat{\theta}) - log(B(\alpha,\beta))$$

   Differentiating the above equation and equating it to zero gives us,

$$\frac{\partial((n\bar{\mathcal{D}} + \beta - 1) \times \log(1-\hat{\theta}) + (n+\alpha-1) \times \log(\hat{\theta}) - log(B(\alpha,\beta)))}{\partial(\hat{\theta})} = 0$$

$$\implies \frac{-1 \times (n\bar{\mathcal{D}} + \beta - 1)}{(1-\hat{\theta})} + \frac{(n+\alpha-1)}{(\hat{\theta})} = 0$$

$$\implies \frac{(n+\alpha-1)}{(\hat{\theta})} = \frac{(n\bar{\mathcal{D}} + \beta - 1)}{(1-\hat{\theta})}$$

   Thus,

$$\hat{\theta} = \frac{n + (\alpha-1)}{n + n\bar{\mathcal{D}} + (\alpha-1) + (\beta-1)} \implies \text{where } \bar{\mathcal{D}} = \frac{\sum_{i=1}^{n} X_i}{n}$$

$$\implies \text{where } \bar{\mathcal{D}} = \frac{\sum_{i=1}^{n} X_i}{n}$$

5. **[3 points]** Is the bias of Maximum Likelihood Estimate (MLE) typically greater than or equal to the bias of Maximum A Posteriori (MAP) estimate? (Explain your answer in a sentence)
   Solution: The bias of Maximum Likelihood Estimate (MLE) typically less than or equal to the bias of Maximum A Posteriori (MAP) estimate since the MAP estimate injects some prior knowledge and typically adds bias.

6. **[3 points]** What can you say about the value of Maximum Likelihood Estimate (MLE) as compared to the value of Maximum A Posteriori (MAP) estimate with a uniform prior? Why?
   Solution:
   We know the posterior is proportional to the product of likelihood and prior, i.e.,

$$p(\theta|x) \propto p(x|\theta) \times p(\theta)$$

   Since the uniform prior gives us a constant value on $p(\theta)$, after proper normalization, we know that the likelihood of MLE and the posterior of MAP are the same. Thus, the MLE and MAP estimators are also the same.

3.3. Logistic Regression [10 pts]

In class, we will learn about MLE of parameters in logistic regression. For a given data $x \in \mathbb{R}^p$, the probability of Y being 1 in logistic regression is

$$P(Y = 1|X = x) = \frac{\exp(w_0 + x^T w)}{1 + \exp(w_0 + x^T w)}, \tag{3}$$

where w_0 and $w = (w_1, w_2, ..., w_p)^T$ are model parameters. In this problem, we consider the maximum a posteriori setting, where we put a Gaussian prior on the parameters:

$$w_i \sim \mathcal{N}(\mu, 1)$$

for $i = 0, 1, 2, ..., p$.

(a) **[5 Points]** Choose a conjugate prior for Gaussian on μ (choose any higher parameters as you want to ease the computation). Assuming you are given a dataset with n training examples and p features, write down a formula for the conditional log posterior likelihood of the training data in terms of the the class labels $y^{(i)}$, the features $x_1^{(i)}, \ldots, x_p^{(i)}$, and the parameters $w_0, w_1, \ldots, w_p$, where the superscript (i) denotes the sample index. This will be your objective function for gradient ascent.
Solution: I choose the prior as $\mu \sim \mathcal{N}(0, 1)$. The likelihood of the data is

$$\begin{aligned} P(w, w_0, \mu|Y, X) &\propto P(Y|X, w, w_0, \mu)P(w_0, w|\mu)P(\mu) \\ &= \prod_{j=1}^{n} \frac{\exp(y^{(j)}(w_0 + (x^{(j)})^T w))}{1 + \exp(w_0 + (x^{(j)})^T w)} \prod_{i=0}^{p} \left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(w_i - \mu)^2}{2} \right) \right) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\mu^2}{2} \right) \end{aligned}$$

The log posterior likelihood is thus

$$f = C + \sum_j \left(y^{(j)}(w_0 + (x^{(j)})^T w) - \ln(1 + \exp(w_0 + (x^{(j)})^T w)) \right) - \sum_{i=0}^{p} \frac{(w_i - \mu)^2}{2} - \frac{\mu^2}{2}.$$

where C is a constant independent with w_0, w, μ.

(b) **[5 Points]** Compute the partial derivative of the objective with respect to w_0, to an arbitrary w_i and μ, i.e. derive $\partial f / \partial w_0$, $\partial f / \partial w_i$, $\partial f / \partial \mu$ where f is the objective that you provided above. Use (3) to simplify the formula. What is the MAP estimation of μ given w_0 and w?
Solution: We have

$$\frac{\partial f}{\partial w_0} = \sum_j \left(y^{(j)} - P(Y = 1|X = x^{(j)}) \right) - (w_0 - \mu).$$

$$\frac{\partial f}{\partial w_i} = \sum_j x_i^{(j)} \left(y^{(j)} - P(Y = 1|X = x^{(j)}) \right) - (w_i - \mu).$$

And

$$\frac{\partial f}{\partial \mu} = \sum_{i=0}^{p} (w_i - \mu) - \mu.$$

Letting it be 0, the MAP estimation of μ given w_0 and w is

$$\mu = \frac{\sum_{i=0}^{p} w_i}{p + 2}.$$

Part C: Programming Exercise [25 Points] (Dan and Danish)

Note: Your code for all of the programming exercises should be submitted to Gradescope. There is a separate 'programming assignment' that should allow you to upload your code easily. Code should be uploaded to this separate programming assignment, while visualizations and written answers should still be submitted within the primary Gradescope assignment. In your code, **Please make it clear in the comments which are the primary functions to compute the answers to each question.**

Exploring The Effect of Priors in Batting Average Estimation: Dan [10 Points]

In this problem, you will explore how prior knowledge can effect your estimates of batting averages.

Dataset

In this problem, we have generated data for 5000 fictional baseball players. The data is divided into 3 parts - 'pre_season.txt', 'mid_season.txt', and 'end_season.txt'. Each of these files has 3 columns: the id for the player (an integer), the number of at_bats for the player (an at-bat is an opportunity to get a hit), and the number of hits the player got during those at-bats. The data files can be loaded using the provided load_data function in hw1_baseball.py. The batting average for a player can be computed by dividing the number of hits by the number of at_bats.

Maximum Likelihood Estimator [3 Points]

Assume for the moment that you only have access to the data in 'mid_season.txt'. Midway through the season, you would like to estimate the end of season batting averages for all 5000 players. Write a function to compute the maximum likelihood estimate of the batting average for all 5000 players. Make sure to turn in your code.

The probability that we observe k hits out of n at bats can be modeled using the Bernoulli distribution as we did in class for coin flips. Then our maximum likelihood estimate can be determined using log-probability of our observations for a single player using:

$$\hat{p} = \underset{p}{\operatorname{argmax}}(k \log p + (n-k)\log(1-p))$$

Taking the derivative with respect to p and setting to 0:

$$\begin{aligned} \frac{k}{\hat{p}} - \frac{n-k}{1-\hat{p}} =& 0 \\ \hat{p} =& \frac{k}{n} \end{aligned}$$

We take $\hat{p}$ to be our estimate of the batting average for that player in the future. This can be implemented in code by simply dividing (element-wise) the vector containing the number of hits for each player by the vector containing the number of at-bats for each player. Using numpy, this looks like normal division. We also choose no data cases to be 0, following the convention from baseball.

```
def mle(at_bats, hits):
    divisor = numpy.where(at_bats != 0, at_bats, numpy.ones_like(at_bats))
    return hits / divisor
```

Maximum a Posteriori Estimator [3 Points]

Unsatisfied with the MLE estimates, you decide that you would like to use the pre-season statistics of the players as a prior on what their in-season batting averages will be. Write a function to compute the maximum a posteriori estimate of the batting average for all 5000 players. Briefly describe how you choose to incorporate prior information. Make sure to turn in your code.

We use the beta distribution as a prior for our data with parameters α (for successes) and β for failures. Then the posterior distribution is also a beta distribution, and when $k + \alpha > 1, (n - k) + \beta > 1$, the MAP estimate for this posterior is given by:

$$\hat{p} = \frac{k + \alpha - 1}{(k + \alpha) + (n - k) + \beta - 2}$$
$$\hat{p} = \frac{k + \alpha - 1}{n + \alpha + \beta - 2}$$

Note that if the $k+\alpha > 1, (n-k)+\beta > 1$ condition does not hold, this closed form expression is not the mode of the Beta distribution. If we were to directly use $\alpha \leftarrow \textit{pre\_season\_hits}$ and $\beta \leftarrow (\textit{pre\_season\_at\_bats} - \textit{pre\_season\_hits})$, then we will not meet these conditions for all players. We also note that if $n = 0, \alpha+\beta = 2$, we get a divide by 0 in this MAP estimate. We also note that returning a 0 batting average for $n = 0$ is equivalent to setting α to 1 and $\beta > 1$ in the prior. In the limit as β approaches 1, this becomes a uniform prior. The uniform prior for these cases is an acceptable answer, but probably not the best choice for players with no data, since we expect most of the mass of our prior distribution to be near .300 based on the data of all the players. A better choice is to set $\alpha = \beta$, or $\frac{\alpha-1}{\alpha+\beta-2} = 0.3$ in those cases. Below I give 4 different reference implementations, all of which would be acceptable. Other choices can also be fine.

```
def max_ap_1(at_bats, hits, pre_season_at_bats, pre_season_hits):
    # returns 0 when we have no data, like using a uniform prior in those cases
    # sets alpha = pre_season_hits + 1, beta = (pre_season_at_bats - pre_season_hits) + 1 in the prior
    divisor = at_bats + pre_season_at_bats
    divisor = numpy.where(divisor != 0, divisor, numpy.ones_like(divisor))
    return (hits + pre_season_hits) / divisor


def max_ap_2(at_bats, hits, pre_season_at_bats, pre_season_hits):
    # if we have enough data, then
    # sets alpha = pre_season_hits + 1, beta = (pre_season_at_bats - pre_season_hits) + 1 in the prior,
    # both are strictly greater than 1. prior is again near uniform
    beta = numpy.maximum(pre_season_at_bats - pre_season_hits + 1, numpy.ones_like(pre_season_at_bats) 
    alpha = numpy.maximum(pre_season_hits + 1, numpy.ones_like(pre_season_hits) + 0.01)
    return (hits + alpha - 1) / (at_bats + alpha + beta - 2)


def max_ap_3(at_bats, hits, pre_season_at_bats, pre_season_hits):
    # sets alpha = pre_season_hits + 1, beta = (pre_season_at_bats - pre_season_hits) + 1 in the prior,
    # both are strictly greater than 1. higher minimum values put more mass near 0.5 in the prior
    beta = numpy.maximum(pre_season_at_bats - pre_season_hits + 1, numpy.full_like(pre_season_at_bats, 
    alpha = numpy.maximum(pre_season_hits + 1, numpy.full_like(pre_season_hits, 1.2))
    return (hits + alpha - 1) / (at_bats + alpha + beta - 2)


def max_ap_4(at_bats, hits, pre_season_at_bats, pre_season_hits):
    # directly returns 0.3 when we don't have enough data for the posterior,
    # corresponds to setting alpha and beta such that the mode of the posterior is 0.3
    # when we have enough data for the posterior,
    # uses alpha = pre_season_hits + 1, beta = (pre_season_at_bats - pre_season_hits) + 1 in the prior
```

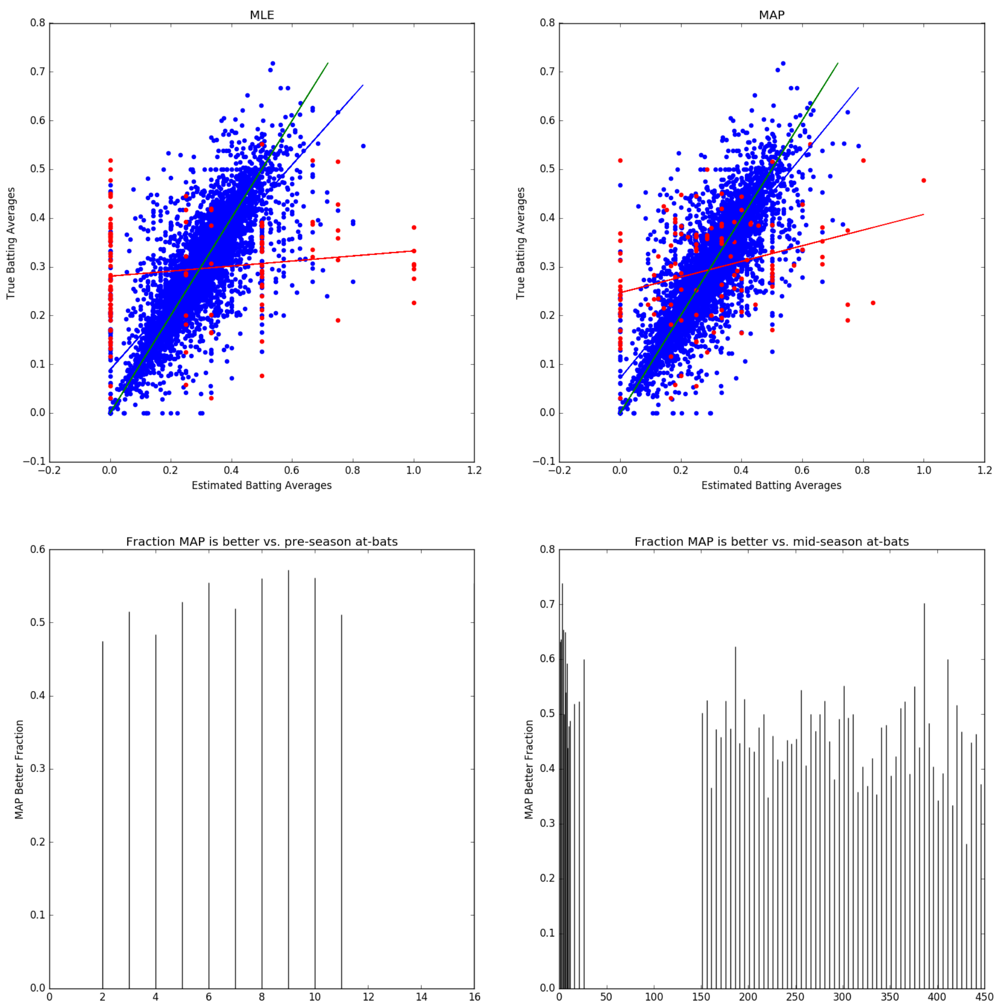
```
    indicator_sufficient = at_bats + pre_season_at_bats > 0
    divisor = numpy.where(indicator_sufficient, at_bats + pre_season_at_bats, numpy.ones_like(at_bats))

    return numpy.where(
        indicator_sufficient,
        (hits + pre_season_hits) / divisor,
        numpy.full_like(pre_season_at_bats, 0.3))
```

Visualize Your Estimates [4 Points]

Compute the actual batting averages from ‘end_season.txt’ (do not include statistics from the other files in these actual averages) and compare your estimates of the batting average to these estimates. Use the provided visualize function in hw1_baseball.py to visualize and compare your MLE and MAP estimators. Make sure to turn in your visualizations.

Here is an example output. Yours may look somewhat different:



- Does the MLE estimator appear to fail in certain cases? Why?
  When there is not enough data, the MLE estimates have very high variance, and they frequently poorly estimate the true parameters.

- Does the MAP estimator appear to fail in certain cases? Why?
  The MAP estimator can also give poor solutions when there is not enough data, depending on the choice of parameters and prior. We reduce the variance by introducing bias, but this bias can potentially also give bad solutions.
- What conclusions do you draw from this experiment?
  In this experiment the prior doesn't matter much except when there are very few at-bats for a player. Using the MAP estimator is a better choice in these cases and overall.

Note: The data files for this subproblem, and the following subproblem can be found here.

Logistic Regression on Movie Review Dataset : Danish [15 Points]

In this problem, you will explore logistic regression to classify movie reviews into two classes - positive & negative. The dataset to be used is IMDB Large Movie Review dataset (Maas et. al, ACL 2011). The datafiles are present in the link shared above.

Details about dataset

The dataset comprises of two folders : 'train' and 'test', and each of these in turn contain two subfolders - pos & neg. Each file in these subfolders is a unique review. In total, we have 25K training reviews (12.5K positive, and remaining 12.5K negative). The test folder too has 12.5K positive and 12.5K negative reviews. For our task, we will use bag of word representation.

Exercises

For this exercise, we will directly use LogisticRegression library from sklearn.linear_models (feel free to use an equivalent library in any language of your choice). We will experiment with different values of $C \in \{0.001, 0.01, 0.1, 1, 10, 100\}$. Here, C is the inverse of regularization constant. We will also closely study the learnt parameter/weight/coefficient vector.
Please note that the answers might differ slightly based on whether a student used an inbuilt CountVectorizer (and inbuilt stopwords library) or not. However, the trends and majority of the answers wouldn't change. The following answers and plots are based on the implementation that used the stopword file that was shared. As per this implementation the vocabulary size is 107132.

- **[4 Points]** Plot train and test accuracy for varying values of C. First plot should contain both train and test accuracy vs C with L2 regularizer (penalty) and the second plot should employ L1 regularizer (penalty). What do you observe in the two plots? Which value of C is optimum in these two cases?

  Solution: The optimum value of C is 0.1 for both cases (for L2 penalty classifier, 0.01 and 0.1 give very close test accuracies). For plots, refer Fig 1
- **[2 Points]** While using L2 regularizer, and different values of C, plot the L2 norm of weight vector vs C. What do you observe?
  Solution: For plots, refer Fig 2
- **[2 Points]** While using L1 regularizer, and different values of C, plot the L1 norm of weight vector vs C. What do you observe?
  Solution: For plots, refer Fig 2
- **[2 Points]** Study how sparsity (i.e percentage of zero elements in a vector) of the parameter vector changes with different values of C. In one plot, depict two curves - one for L1 regularizer and the other one for L2 regularizer. Jot down your observations.
  Solution: Refer to fig 3

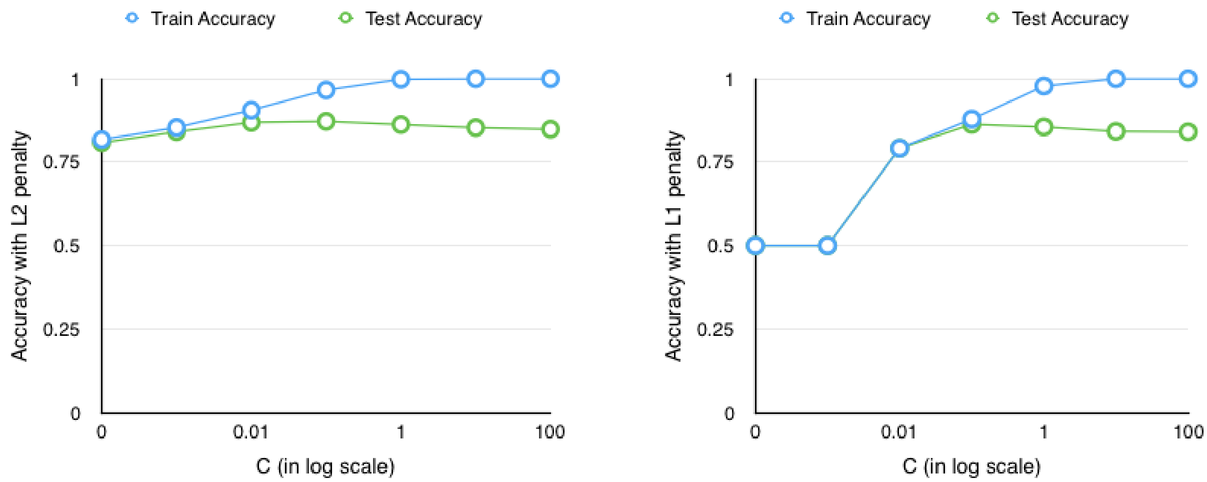


Figure 1: Train and Test accuracy vs C

Now we will try to visualize the basis of the classification! One way to do so is to look at the weight vector and analyze the top (least) K values.

- **[3 Points]** While using L2 regularizer, and the optimum value of C (with respect to test accuracy), which 5 words correspond to the largest weight indices in the learnt weight vector? Which 5 words correspond to the least weight indices in the learnt parameter vector?
  Solution: Top words corresponding to largest weights : excellent, perfect, favorite, superb, wonderful
  Top words corresponding to least weights : worst, waste, awful, disappointment, poorly

- **[2 Points]** While using L2 regularizer, and the optimum value of C (with respect to test accuracy), which review is predicted positive with highest probability? Similary, which review is predicted negative with highest certainty?
  Solution: Most positive review : test/pos/5358_9.txt Most negative review : test/neg/3478_1.txt

  The copy of the solution code can be found at http://pastebin.com/ca7LSba4.

Implementation Details

- Please ignore all the words in the file 'stopwords.txt'. The vocabulary must be constructed by splitting the raw text only with whitespace characters, and converting them into lowercase.
- Since the vocabulary size will be close to 100K, and there are 25K reviews in training and test set, the bag of word matrix will be somewhat large, hence its best to use sparse formats wherever necessary.
- You may choose to ignore the words that occur in the test set but never show up in the training set.

Note : For the entire programming exercise, please turn in your code in a single zipped folder that might contain multiple source code files.

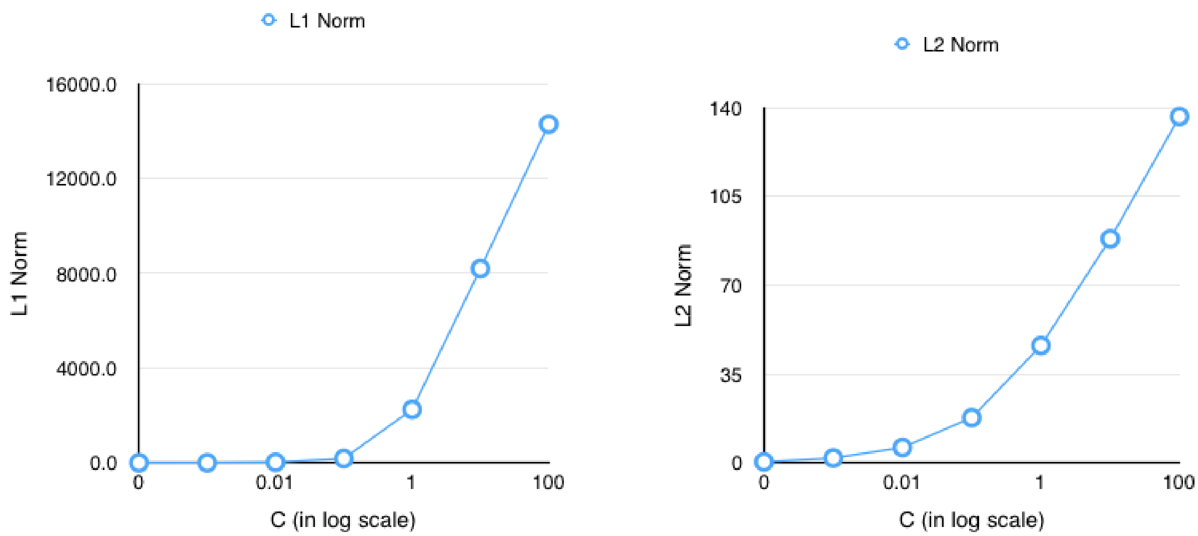


Figure 2: L1 and L2 norms vs C

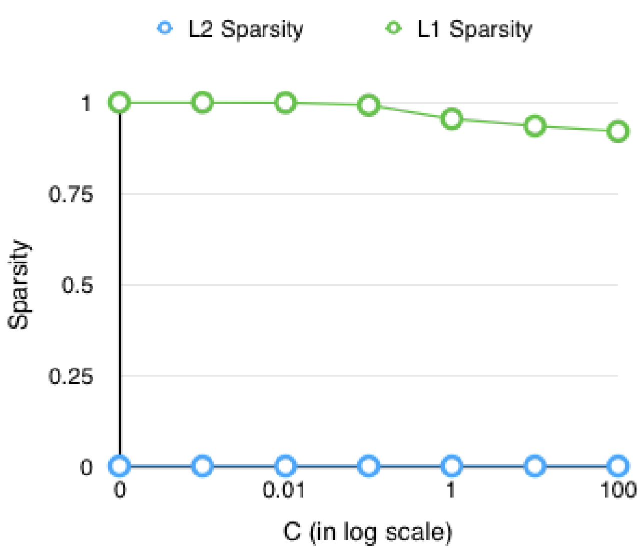


Figure 3: Sparsity vs C