CS 189 Introduction to Machine Learning
Fall 2019 Jennifer Listgarten & Stella Yu

HW03

Due: 9/26 at 10 p.m.

0 Getting Started

Read through this page carefully. You may typeset your homework in latex or submit neatly handwritten/scanned solutions. Please start each question on a new page. Deliverables:

1. Submit a PDF of your writeup, **with an appendix for your code**, to assignment on Gradescope, "HW[X] Write-Up". If there are graphs, include those graphs in the correct sections. Do not simply reference your appendix.
2. If there is code, submit all code needed to reproduce your results, "HW[X] Code".
3. If there is a test set, submit your test set evaluation results, "HW[X] Test Set".
4. In all cases, replace "[X]" with the number of the assignment you are submitting.

1 Alternative Priors

Maximum A Posteriori (MAP) is a method to estimate some parameters θ from a probabilistic perspective. Let the data be X and the parameter be θ, MAP assumes that θ is a distribution and utilizes Bayesian methods to compute the posterior distribution $P(\theta|X)$ given prior knowledge of $P(\theta)$. MAP does not yield the full distribution of $P(\theta|X)$ due to intractability issues and instead infers a point estimate of θ by maximizing the posterior distribution:

$$\begin{aligned}\theta_{\text{MAP}} &= \arg\max_{\theta} P(\theta|X) = \arg\max_{\theta} P(X|\theta)P(\theta) \\ &= \arg\max_{\theta} \prod_{i=1}^{n} P(x_i|\theta)P(\theta) = \arg\max_{\theta} \sum_{i=1}^{n} \log P(x_i|\theta) + n \log P(\theta),\end{aligned}$$

where $x_i \in \mathbb{R}^d$ and sampled from the data distribution. In lecture, we discovered the the bijection between Ridge Regression and MAP when both the noise and prior follow a Gaussian distribution. In this problem, we will explore the use of alternative priors in MAP and see how these subtle changes affect our final solution θ^*.

(a) In Ridge Regression, we assume that $y = X\theta + Z$, where $X \in \mathbb{R}^{n\times d}$, $\theta \in \mathbb{R}^d$, and $Z \sim N(0, \sigma^2 I_n)$ follows the form of a standard Gaussian distribution. We also assume that the initial distribution of $P(\theta) \sim N(0, \sigma_h^2 I_d)$. Now, we will explore the more general case where

the prior is a distribution not centered at 0. Prove that, if the prior $P(\theta) \sim N(\mu_\theta, \sigma_h^2 I_d)$, the solution is:

$$\theta^* = \left(X^T X + \frac{\sigma^2}{\sigma_h^2} I\right)^{-1} \left(X^T y + \frac{\sigma^2}{\sigma_h^2} \mu_\theta\right)$$

(b) Consider $d = 2$ and the linear model $y = \theta_1^* x + \theta_2^* + z$, where z is the standard normal distribution $N(0, 1)$. Suppose $\theta_1^* = 2$ and $\theta_2^* = 1$ with $n = 10$ data points sampled from this linear model. For now, assume the prior $P(\theta) = N(0, \sigma_h^2 I_d)$ with σ_h initially at 1. **Write a python program that graphs contour plots in the weight space (θ_1 for x-axis, θ_2 for y-axis) for the prior $P(\theta)$, posterior $P(\theta|X)$, and the MLE distribution $P(X|\theta)$.** Ideally, three plots should be made for the prior, posterior, and MLE distribution respectively. What happens if you change $\sigma_h = 10$? Plot these results and explain why. (Starter code is provided in `problem1_starter.py`. Please set the random seed to 7 and total data points to 10 for consistency in grading.)

(c) Continuing from Part (b), if our prior is $P(\theta) = N(\mu_\theta, \sigma_h^2 I_2)$, what happens if we change $\mu_\theta = \begin{bmatrix} -5 \\ -5 \end{bmatrix}$? What about $\mu_\theta = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$? **Plot similar diagrams mentioned in Part(b).** From your results, how does the position of the prior affect the position θ_{MAP} ? At the same time, how does θ_{MLE} change with respect to the prior?

(d) The Laplace distribution parameterized by μ and σ has the following PDF:

$$L(x; \mu, \sigma_h) = \frac{1}{2\sigma_h} \exp\left(-\frac{|x - \mu|}{\sigma_h}\right).$$

Now, assume that our prior for each weight θ_i is i.i.d. and follows the distribution $P(\theta_i) \sim L(\mu_i, \sigma_h)$ for $i = 1, 2, ..., d$. Like before, $y = X\theta + Z$, with $Z \sim N(0, \sigma^2 I_n)$. Prove that optimizing MAP is equivalent to the optimization equation:

$$\arg\min_{\theta \in \mathbb{R}^d} ||y - X\theta||^2 + 2\frac{\sigma^2}{\sigma_h}|\theta - \mu|_1,$$

where $|\cdot|_1$ is the L1 norm. This equation will be useful in plotting Laplace MAP contours in the next sub-question.

(e) From Parts (b) and (d), assume that our prior for all θ_i follows $P(\theta_i) \sim L(0, 1)$. **With everything else held constant, fill in the starter code from Part (b) to calculate similar contour plots for prior, posterior, and MAP for weights θ_1 and θ_2.** Now try, $P(\theta_i) \sim L(0, 0.0001)$ and plot this too. What is happening to the weights in θ_{MAP}? And why?

(f) What if our prior θ is uniformly distributed in the range $(0, 1)$? How is the MAP solution related to the MLE solution in this case? (No need to plot)

2 Probabilistic Model of Linear Regression

Both ordinary least squares and ridge regression have interpretations from a probabilistic standpoint. In particular, assuming a generative model for our data and a particular noise distribution,

we will derive least squares and ridge regression as the maximum likelihood and maximum *a-posteriori* parameter estimates, respectively. This problem will walk you through a few steps to do that. (Along with some side digressions to make sure you get a better intuition for MLE and MAP estimation.)

(a) Assume that X and Y are both one-dimensional random variables, i.e. $X, Y \in \mathbb{R}$. Assume an affine model between X and Y: $Y = Xw_1 + w_0 + Z$, where $w_1, w_0 \in \mathbb{R}$, and $Z \sim N(0, 1)$ is a standard normal (Gaussian) random variable. Assume w_1, w_0 are fixed parameters (i.e., they are not random). **What is the conditional density (PDF) of Y given X?**

(b) Given n points of training data $\{(x_1, Y_1), (x_2, Y_2), \ldots, (x_n, Y_n)\}$ generated in an iid fashion by the probabilistic model in the previous part, **derive the maximum likelihood estimator for w_1, w_0 from this training data.**

(c) Now, consider a different generative model. Let $Y = Xw + Z$, where $Z \sim U[-0.5, 0.5]$ is a continuous random variable uniformly distributed between -0.5 and 0.5. Again assume that w is a fixed parameter. **What is the conditional distribution of Y given X?**

(d) Given n points of training data $\{(x_1, Y_1), (x_2, Y_2), \cdots, (x_n, Y_n)\}$ generated in an i.i.d. fashion in the setting of the part (c) **derive a maximum likelihood estimator of w.** Assume that $x_i > 0$ for all $i = 1, \ldots, n$. (Note that MLE for this case need not be unique; but you are required to report only one particular estimate.)

(e) (One-dimensional Ridge Regression) Now, let us return to the case of Gaussian noise. Given n points of training data $\{(x_1, Y_1), (x_2, Y_2), \cdots, (x_n, Y_n)\}$ generated according to $Y_i = x_i W + Z_i$, where $Z_i \sim N(0, 1)$ are iid standard normal random variables. Assume $W \sim N(0, \sigma^2)$ as a prior is also a normal random variable and is independent of both the Z_i's and the x_i's. **Use Bayes' Theorem to derive the posterior distribution of W given the training data. What is the mean of the posterior distribution of W given the data?**

Hint: Compute the posterior up-to proportionality and try to identify the distribution by completing the square.

(f) Consider n training data points $\{(\mathbf{x}_1, Y_1), (\mathbf{x}_2, Y_2), \cdots, (\mathbf{x}_n, Y_n)\}$ generated according to $Y_i = \mathbf{w}^\top \mathbf{x}_i + Z_i$ where $Y_i \in \mathbb{R}, \mathbf{w}, \mathbf{x}_i \in \mathbb{R}^d$ with $\mathbf{w}$ fixed, and $Z_i \sim N(0, 1)$ iid standard normal random variables. **Argue why the maximum likelihood estimator for w is the solution to a least squares problem.**

(g) (Multi-dimensional ridge regression) Consider the setup of the previous part: $Y_i = \mathbf{W}^\top \mathbf{x}_i + Z_i$, where $Y_i \in \mathbb{R}, \mathbf{W}, \mathbf{x}_i \in \mathbb{R}^d$, and $Z_i \sim N(0, 1)$ iid standard normal random variables. Now we treat $\mathbf{W}$ as a random vector and assume a prior knowledge about its distribution. In particular, we use the prior information that the random variables W_j are i.i.d. $\sim N(0, \sigma^2)$ for $j = 1, 2, \ldots, d$. **Derive the posterior distribution of W given all the $\mathbf{x}_i, Y_i$ pairs. What is the mean of the posterior distribution of the random vector W?**

Hint: Use hints from part (f) and the following identities: For $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}$ and $\mathbf{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}$ we have $\mathbf{X}^\top \mathbf{X} = \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top$ and $\mathbf{X}^T \mathbf{Y} = \sum_{i=1}^n \mathbf{x}_i Y_i$.

3 Simple Bias-Variance Tradeoff

Consider a random variable X, which has unknown mean μ and unknown variance σ^2. Given n iid realizations of training samples $X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n$ from the random variable, we wish to estimate the mean of X. We will call our estimate of X the random variable $\hat{X}$, which has mean $\hat{\mu}$. There are a few ways we can estimate μ given the realizations of the n samples:

1. Average the n samples: $\frac{x_1+x_2+\ldots+x_n}{n}$.
2. Average the n samples and one sample of 0: $\frac{x_1+x_2+\ldots+x_n}{n+1}$.
3. Average the n samples and n_0 samples of 0: $\frac{x_1+x_2+\ldots+x_n}{n+n_0}$.
4. Ignore the samples: just return 0.

In the parts of this question, we will measure the *bias* and *variance* of each of our estimators. The *bias* is defined as

$$\mathbb{E}[\hat{X} - \mu]$$

and the *variance* is defined as

$$\text{Var}[\hat{X}].$$

(a) **What is the bias of each of the four estimators above?**

(b) **What is the variance of each of the four estimators above?**

(c) Suppose we have constructed an estimator $\hat{X}$ from some samples of X. We now want to know how well $\hat{X}$ estimates a fresh (new) sample of X. Denote this fresh sample by X'. Note that X' is an i.i.d. copy of the random variable X. **Derive a general expression for the expected squared error $\mathbb{E}[(\hat{X} - X')^2]$ in terms of σ^2 and the bias and variance of the estimator $\hat{X}$. Similarly, derive an expression for the expected squared error $\mathbb{E}[(\hat{X} - \mu)^2]$. Compare the two expressions and comment on the differences between them, if any.**

(d) For the following parts, we will refer to expected total error as $\mathbb{E}[(\hat{X} - \mu)^2]$. It is a common mistake to assume that an unbiased estimator is always "best." Let's explore this a bit further. **Compute the expected squared error for each of the estimators above.**

(e) **Demonstrate that the four estimators are each just special cases of the third estimator, but with different instantiations of the hyperparameter n_0.**

(f) **What happens to bias as n_0 increases? What happens to variance as n_0 increases?**

(g) Say that $n_0 = \alpha n$. **Find the setting for α that would minimize the expected total error, assuming you secretly knew μ and σ.** Your answer will depend on σ, μ, and n.

(h) For this part, let's assume that we had some reason to believe that μ *should be small* (close to 0) and σ *should be large*. In this case, **what happens to the expression in the previous part?**

(i) In the previous part, we assumed there was reason to believe that μ *should be small*. Now let's assume that we have reason to believe that μ is not necessarily small, but *should be close to some fixed value* μ_0. **In terms of X and μ_0, how can we define a new random variable X' such that X' is expected to have a small mean? Compute the mean and variance of this new random variable.**

(j) Draw a connection between α in this problem and the regularization parameter λ in the ridge-regression version of least-squares. **What does this problem suggest about choosing a regularization coefficient and handling our data-sets so that regularization is most effective**? This is an open-ended question, so do not get too hung up on it.

4 Robotic Learning of Controls from Demonstrations and Images

Huey, a home robot, is learning to retrieve objects from a cupboard, as shown in Fig. 1. The goal is to push obstacle objects out of the way to expose a goal object. Huey's robot trainer, Anne, provides demonstrations via tele-operation. When tele-operating the robot, Anne can look at the images captured by the robot and provide controls to Huey remotely.

During a demonstration, Huey records the RGB images of the scene for each timestep, $x_0, x_1, ..., x_n$, where $x_i \in \mathbb{R}^{30\times30\times3}$ and the controls for his body, $u_0, u_1, \ldots, u_n$, where $u_i \in \mathbb{R}^3$. The controls correspond to making small changes in the 3D pose (i.e. translation and rotation) of his body. Examples of the data are shown in the figure.

Under an assumption (sometimes called the Markovian assumption) that all that matters for the current control is the current image, Huey can try to learn a linear *policy* π (where $\pi \in \mathbb{R}^{2700\times3}$) which linearly maps image states to controls (i.e. $\pi^\top x = u$). We will now explore how Huey can recover this policy using linear regression. Note: please use **numpy** and **numpy.linalg** to complete this assignment and convert the loaded data `x_train.p` and `x_test.p` into float.

(a) To get familiar with the structure of the data, **please visualize the 0th, 10th and 20th images in the training dataset. Also find out what's their corresponding control vectors.**

(b) Load the n training examples from `x_train.p` and compose the matrix X, where $X \in \mathbb{R}^{n\times2700}$. Note, you will need to flatten the images to reduce them to a single vector. The flattened image vector will be denoted by $\bar{x}$ (where $\bar{x} \in \mathbb{R}^{2700\times1}$). Next, load the n examples from `y_train.p` and compose the matrix U, where $U \in \mathbb{R}^{n\times3}$. Try to perform ordinary least squares to solve:

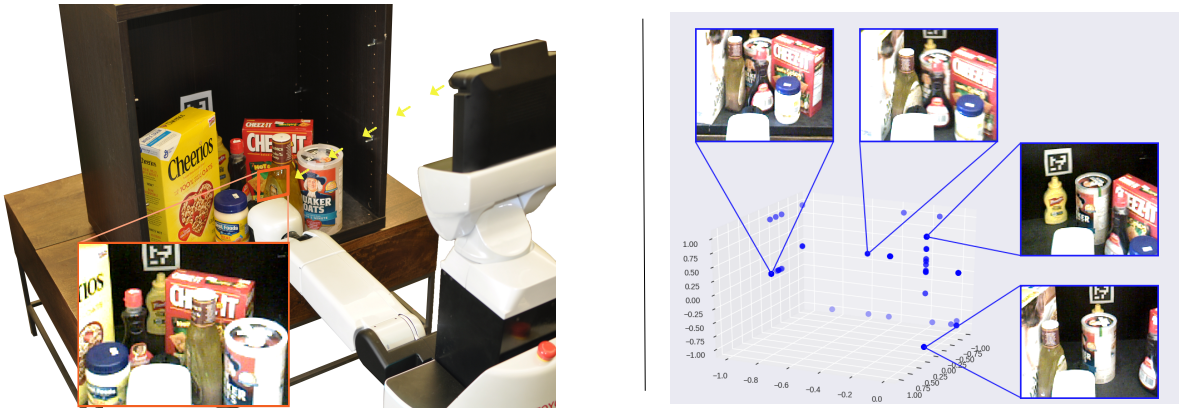

Figure 1: A) Huey trying to retrieve a mustard bottle. An example RGB image of the workspace taken from his head mounted camera is shown in the orange box. The angle of the view gives Huey and eye-in-hand perspective of the cupboard he is reaching into. B) A scatter plot of the 3D control vectors, or u labels. Notice that each coordinate of the label lies within the range of $[-1, 1]$ for the change in position. Example images, or states x, are shown for some of the corresponding control points. The correspondence is indicated by the blue lines.

$$\min_{\pi} ||X\pi - U||_F$$

to learn the *policy* $\pi \in \mathbb{R}^{2700\times 3}$. **Report what happens as you attempt to do this and explain why.**

(c) Now try to perform ridge regression:

$$\min_{\pi} ||X\pi - U||_2^2 + \lambda||\pi||_2^2$$

on the dataset for regularization values $\lambda = \{0.1, 1.0, 10, 100, 1000\}$. Measure the average squared Euclidean distance for the accuracy of the policy on the training data:

$$\frac{1}{n}\sum_{i=0}^{n-1} ||\bar{x}_i^T\pi - u_i||_2^2$$

Report the training error results for each value of λ.

(d) Next, we are going to try standardizing the states. For each pixel value in each data point, x, perform the following operation:

$$x \mapsto \frac{x}{255} \times 2 - 1.$$

Since we know the maximum pixel value is 255, this rescales the data to be between $[-1, 1]$. **Repeat the previous part and report the average squared training error for each value of** λ.

(e) Evaluate both *policies* (i.e. with and without standardization on the new validation data `x_test.p` and `y_test.p` for the different values of λ. **Report the average squared Euclidean loss** and **qualitatively explain how changing the values of λ affects the performance in terms of bias and variance**.

(f) To better understand how standardizing improved the loss function, we are going to evaluate the *condition number* κ of the optimization, which is defined as

$$\kappa = \frac{\sigma_{\max}(X^T X + \lambda I)}{\sigma_{\min}(X^T X + \lambda I)}$$

or the ratio of the maximum singular value to the minimum singular value of the relevant matrix. Roughly speaking, the condition number of the optimization process measures how stable the solution will be when some error exists in the observations. More precisely, given a linear system $Ax = b$, the condition number of the matrix A is the maximum ratio of the relative error in the solution x to the relative error of b.

For the regularization value of $\lambda = 100$, **report the condition number with the standardization technique applied and without**.