Homework 0: Statistical Foundations of Reinforcement Learning (Spring 2021)

University of California, Santa Barbara

Assigned on Mar 29, 2020 (Monday)

Due at 11:59 pm on Apr 5, 2020 (Monday)

Notes:

- Be sure to read "Policy on Academic Integrity" on the course syllabus.
- There are ***[85 points]*** in total in this homework.
- You need to submit your homework via Gradescope.
- Contract the instructor if you spot typos. Any updates or correction will be posted on the course Announcements page and piazza, so check there occasionally.

---

1 Policies *[0 points]*

Please read these policies. **Please answer the three questions below and include your answers marked in a "problem 1" in your solution set.** Homeworks which do not include these answers will not be graded.

Gradescope submission: When submitting your HW, please tag your pages correctly as is requested in gradescope. Untagged homeworks will not be graded, until the tagging is fixed.

Readings: Read the notes and required material.

Submission format: Submit your report as a *single* pdf file. Please typeset your writing using LaTeX, or compile scanned pages of hand-written solutions. Hand-written solutions must be legible for credit.

Collaboration: It is acceptable for you to discuss problems with other students; it is not acceptable for students to look at another students written answers. Each student must understand, write, and hand in their own answers.

Acknowledgments: If you find out solutions in published material, on the web, or from other textbooks, this must be acknowledged. If you find proofs in existing papers, it is ok to use these for guidance; you must acknowledge this, and you should always first make an attempt at the answer on your own. You need to understand all the written steps that you write.

1.1 List of Collaborators

List the names of all people you have collaborated with and for which question(s).

1.2 List of Acknowledgements

If you find an assignment's answer or use a another source for help, acknowledge for which question and provide an appropriate citation (there is no penalty, provided you include the acknowledgement). If not, then write "none".

1.3 Certify that you have read the instructions

Write "I have read these policies" to certify this.

1.4 Certify that you have read the Syllabus on the course website

Write "I have reviewed the syllabus of the course." to certify this.

2 Refreshers on optimization and probability. *[35 points]*

Some standard notations in optimization: $\max, \min, \arg\max, \arg\min$ are commonly used when working with optimizations programs. The subscripts of these operators denote the "argument / variable" that you are optimizing over. For example, in $\min_{\theta\in\Theta} f(\theta)$, f is the objective function or criterion function, θ is the argument or variable that you are optimizing over. Θ is the domain that you can choose θ from (sometimes abbreviated when it is the whole space or clear from context). For example, $\min_{\theta\in\Theta} f(\theta)$ returns the minimum objective function value; $\arg\min_\theta f(\theta)$ returns the argument θ^* that achieves the minimum function value.

(a) (Continuous optimization) Let $x_1, \ldots, x_n$ be real values. Consider a quadratic function $f(\theta) = \sum_{i=1}^n w_i(x_i - \theta)^2$. Assume $w_i > 0$. Derive the optimal solution θ^* that minimizes $f(\theta)$ - denoted by $\theta^* = \arg\min_\theta f(\theta)$? What happens if some w_i is negative?

(b) (counting and combinatorics) If $2n$ kids are randomly divided into two equal-sized subgroups, find the probability that the two tallest kids will be: (i) in the same subgroup; (ii) in different subgroups.

(c) (Bayes rule) In answering a question on a multiple choice test, a candidate either knows the answer with probability p ($0 \le p < 1$) or does not know the answer with probability $1 - p$. If she knows the answer, she puts down the correct answer with probability 0.99, whereas if she guesses, the probability of his putting down the correct result is $1/k$ (k choices to the answer). Find the conditional probability that the candidate knew the answer to a question, given that she has made the correct answer.

(d) (Likelihood and maximum likelihood) Consider a biased coin with the probability of turning up head $0 < p < 1$. We flip the coin 10 times and get a sequence of outcomes: $\{T, H, H, T, T, H, H, H, T, H\}$. We know that the likelihood (joint probability of the data as a function of the parameter p) of observing this sequence

$$L(p) := (1-p)\cdot p\cdot p\cdot(1-p)\cdot(1-p)\cdot p\cdot p\cdot p\cdot(1-p)\cdot p = p^6(1-p)^4$$

Calculate the value of p that maximizes the likelihood $L(p)$.

(Hint: you may wish to consider maximizing $\log L(p)$ instead. Why does it preserve the $\arg\max$ — the p^* that maximizes $L(p)$?)

(e) (Calculus, gradients) Let $w \in \mathbb{R}^d$ be a column vector. Let $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ be column vectors of the same dimension d and $y_1, \ldots, y_n \in \mathbb{R}$ be scalars. Let $\lambda \in \mathbb{R}$ be a non-negative scalar value. Consider function $F(w) = \sum_{i=1}^n (x_i^T w - y_i)^2 + \lambda \|w\|^2$. Calculate the gradient of F. Recall that the gradient with respect to a vector of variables is the vector of partial derivatives with respect to each variable w_i:

$$\nabla F(w) = \left[\frac{\partial F(w)}{\partial w_1}, \ldots, \frac{\partial F(w)}{\partial w_d}\right]^T \in \mathbb{R}^d.$$

(Hint: It is easier to work out the partial derivatives one at a time, then concatenate the partial derivatives into a gradient by applying the definition above. The same hint also applies to other instances where you need to calculate the gradient.)

(f) (Chain-rule and softmax function) Let $x_1, \ldots, x_n$ be real values. Let

$$f(x_1, \ldots, x_n) = \log \sum_{i=1}^n \exp(x_i).$$

This is called the softmax function or the log-sum-exp function[1] Calculate the gradient of f w.r.t. vector $x = (x_1, \ldots, x_n)^T$.

(g) (Simple mathematical proof) For the soft-max function in part (f). Prove that

$$\max_i x_i \leq f(x_1, \ldots, x_n) \leq \max_i (x_i) + \log n.$$

(Hint: A good practice when writing proof is to write a sequence of inequalities and explain every line in the following form:

"$f(x_1, \ldots, x_n) = \log \sum_{i=1}^n \exp(x_i) \leq ... \leq ... \leq \max_i (x_i) + \log n$. The first inequality is by definition, the second inequality is because ..., the third inequality is because ... ")

3 Refresher on time complexity and Python / Numpy / Scipy *[15 points]*

Please use Python3. The easiest interactive programming interface is perhaps jupyter notebook.

(a) In numpy, generate two matrices A and B with size 5 by 4 and size 4 by 3 respectively. In matrix A, make sure that the values are $1, 2, 3, \ldots, 19, 20$, from left-to-right then top to bottom. In matrix B, make sure that the values are $1, 2, 3, \ldots, 11, 12$ from top to bottom, then left to right. Write a script that print matrix A, matrix B and the matrix product of

[1] Note that this softmax function is a scalar function $\mathbb{R}^n \to \mathbb{R}$ and it is different from the "softmax" transformation typically used in machine learning to convert any score vector to a probability vector, i.e., a vector valued function from $\mathbb{R}^n \to \mathbb{R}^n$. The latter is a misnomer but has a wide-spread misuse to the point that it becomes a convention. We will be working with the other function too. To not confuse ourselves, we call this the soft-argmax function. See more details in Q6.)

AB. Attach your code and the printed output to your report (i.e., print out the Jupyter notebook).

Hint: you can use numpy.dot for matrix multiplication.

(b) (Sparse matrix-vector multiplication) What is the worst case time complexity (in Big O notation) of multiplying a matrix A of dimension $\mathbb{R}^{n\times n}$ with a dense vector $v \in \mathbb{R}^n$?

What is the time complexity if matrix A is sparse, denote the number of non-zero elements by $\text{nnz}(A)$?

Write a python function that takes a nonnegative integer n and outputs a sparse matrix A of size $(n-1)\times n$, such that for any $x \in \mathbb{R}^n$, $Ax = [x_1 - x_2, ..., x_{n-1} - x_n]^T$. Calls this function and print the resulting A for $n = 5$ using `print(A.toarray())`

(Make sure you use the sparse matrix library in python: scipy.sparse.)

(c) Create a 2 by 2 matrix A with entries being $\begin{bmatrix} 0.7 & 0.4 \\ 0.3 & 0.6 \end{bmatrix}$ and a vector $x = [1, 0]^T \in \mathbb{R}^2/$ Write a for loop in python that apply matrix A to x for k times. Print out the values of $Ax, A^2x, ..., A^kx$ for $k = 25$. What do you observe?

4 L_p norms *[20 points]*

For a vector $x \in \mathbb{R}^d$, the L_p norm is defined as:

$$\|x\|_p = (|x_1|^p + |x_2|^p + \dots |x_d|^p)^{1/p}$$

We will be frequently making use of $p = 1, 2, \infty$ norms.

(a) ***[5 points]*** Give a simple formula for the L_∞ norm, which is defined as $\|x\|_\infty = \lim_{p\to\infty} \|x\|_p$.

(b) ***[5 points]*** Verify that $\|x\|_\infty$ is a norm on vector space, i.e. for vectors a and b and scalar c, it satisfies the following three properties: $\|a + b\|_\infty \le \|a\|_\infty + \|b\|_\infty$; $\|ca\|_\infty = |c|\|a\|_\infty$; if $\|a\|_\infty = 0$ then a is the all 0's vector.

(c) ***[5 points]*** Show that:

$$\|x\|_\infty \le \|x\|_2 \le \|x\|_1$$

(d) ***[5 points]*** Verify the following special case of Holder's inequality for vectors a and b:

$$a \cdot b \le \|a\|_1 \|b\|_\infty$$

5 Statistics and inequalities*[20 points]*

(a) ***[5 points]*** (Convergence of the mean of i.i.d. samples) Let $X_1, ..., X_n$ be i.i.d. Bernoulli random variables (they take 1 with probability p and 0 with probability $1 - p$), calculate the mean and variance of $\frac{1}{n}\sum_i X_i$.

(b) ***[5 points]*** (Law of Total Expectation) Professor Yu-Xiang Wang plays basketball. Let us assume each shot Prof. Wang fires at the basket is an independent random variable, where he shoots three pointers with a 1/4 hit rate, and mid-range jumpers with a 1/3 hit rate. He shoots 10 balls in the first half of game (with a probability 0.4, a shot is a three-pointer attempt) and 20 balls in the second half of the game (with a probability 0.2 each shot is a three-pointer attempt).

Calculate the expected number of points Prof. Wang will get in this game using the **law of total expectation**.

(c) ***[5 points]*** (Law of Total Variance) For the same problem as in (b), calculate *the variance* of Prof. Wang's total number of points in this game by applying the **Law of Total Variance**.

For part (b) and (c), you may use Python / Jupyter notebook for the computation, but make sure you include how you are applying the law of total expectation / law of total variance.

(d) ***[5 points]*** (Hoeffding's inequality) Hoeffding's inequality says that if $Y_1, ..., Y_n$ are *independent* random variables satisfying that $a \leq Y_i \leq b$, then for all $t > 0$

$$\mathbb{P}\left[\sum_{i=1}^{n}(Y_i - \mathbb{E}[Y_i]) > t\right] \leq e^{-\frac{2t^2}{n(b-a)^2}}.$$

Now consider the i.i.d. Bernoulli random variables from Problem 5(a) Show (using Hoeffding's inequality) that for any $0 < \delta \leq 1$, with probability at least $1 - \delta$,

$$\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}[X_1]\right| \leq \sqrt{\frac{\log(2/\delta)}{2n}}.$$