CS 189:289A Introduction to Machine Learning
Fall 2021 Jennifer Listgarten and Jitendra Malik

HW1

Due 9/20/21 at 11:59pm

- Homework 1 consists of both written and coding questions.
- We prefer that you typeset your answers using LaTeX or other word processing software. If you haven't yet learned LaTeX, one of the crown jewels of computer science, now is a good time! Neatly handwritten and scanned solutions will also be accepted for the written questions.
- In all of the questions, **show your work**, not just the final answer.

Deliverables:

1. Submit a PDF of your homework to the Gradescope assignment entitled "HW1 Write-Up". **Please start each question on a new page.** If there are graphs, include those graphs in the correct sections. **Do not** put them in an appendix. We need each solution to be self-contained on pages of its own.
    - In your write-up, please state with whom you worked on the homework. This should be on its own page and should be the first page that you submit.
    - In your write-up, please copy the following statement and sign your signature next to it. (Mac Preview and FoxIt PDF Reader, among others, have tools to let you sign a PDF file.) We want to make it *extra* clear so that no one inadvertently cheats. *"I certify that all solutions are entirely in my own words and that I have not looked at another student's solutions. I have given credit to all external sources I consulted."*
    - **Replicate all your code in an appendix**. Begin code for each coding question in a fresh page. Do not put code from multiple questions in the same page. When you upload this PDF on Gradescope, *make sure* that you assign the relevant pages of your code from appendix to correct questions.

1 Multivariate Gaussians: A review

(a) Consider a two dimensional random variable $Z \in \mathbb{R}^2$. In order for the random variable to be jointly Gaussian, a necessary and sufficient condition is that

- Z_1 and Z_2 are each marginally Gaussian, and
- $Z_1|Z_2 = z$ is Gaussian, and $Z_2|Z_1 = z$ is Gaussian.

A second characterization of a jointly Gaussian RV $Z \in \mathbb{R}^2$ is that it can be written as $Z = AX$, where $X \in \mathbb{R}^2$ is a collection of i.i.d. standard normal RVs and $A \in \mathbb{R}^{2\times 2}$ is a matrix.

Note that the probability density function of a multivariate Gaussian RV with mean vector, μ, and covariance matrix, Σ, is:

$$f(\mathbf{z}) = \exp\left(-\frac{1}{2}(\mathbf{z}-\mu)^T\Sigma^{-1}(\mathbf{z}-\mu)\right) / \sqrt{(2\pi)^k|\Sigma|}$$

.

Let X_1 and X_2 be i.i.d. standard normal RVs. Let U denote a binary random variable uniformly distributed on $\{-1, 1\}$, independent of everything else. Use one of the two characterizations given above to determine whether the following RVs are jointly Gaussian, and calculate the covariance matrix (regardless of whether the RVs are jointly Gaussian).

- $Z_1 = X_1$ and $Z_2 = X_2$.
- $Z_1 = X_1$ and $Z_2 = X_1 + X_2$.
- $Z_1 = X_1$ and $Z_2 = -X_1$.
- $Z_1 = X_1$ and $Z_2 = UX_1$.

(b) Use the above example to show that two Gaussian random variables can be uncorrelated, but not independent. On the other hand, show that two uncorrelated, jointly Gaussian RVs are independent.

(c) With the setup above, let $Z = VX$, where $V \in \mathbb{R}^{2\times 2}$, and $Z, X \in \mathbb{R}^2$. What is the covariance matrix Σ_Z? Is this also true for a RV other than Gaussian?

(d) Use the above setup to show that $X_1 + X_2$ and $X_1 - X_2$ are independent. Give another example pair of linear combinations that are independent.

(e) Given a jointly Gaussian RV $Z \in \mathbb{R}^2$ with covariance matrix $\Sigma_Z = \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{12} & \Sigma_{22}\end{bmatrix}$, how would you derive the distribution of $Z_1|Z_2 = z$?

Hint: The following identity may be useful

$$\begin{bmatrix}a & b\\ b & c\end{bmatrix}^{-1} = \begin{bmatrix}1 & 0\\ -\frac{b}{c} & 1\end{bmatrix}\begin{bmatrix}\left(a-\frac{b^2}{c}\right)^{-1} & 0\\ 0 & \frac{1}{c}\end{bmatrix}\begin{bmatrix}1 & -\frac{b}{c}\\ 0 & 1\end{bmatrix}.$$

2 Linear Regression, Projections and Pseudoinverses

We are given $X \in \mathbb{R}^{n \times d}$ where $n > d$ and $\text{rank}(X) = d$. We are also given a vector $y \in \mathbb{R}^n$. Define the orthogonal projection of y onto range(X) as $P_X(y)$.

(a) Prove that $P_X(y) = \underset{w \in \text{range}(X)}{\arg\min} |y - w|^2$.

Side Note: In lecture, we learned how to find $\hat{\theta}$ that minimizes the least squares loss $L(\theta) = |y - X\theta|^2$. In other words, we tried to find θ such that $X\theta$ is the vector in the columnspace of X that is closest to our response vector y. Hence, $P_X(y) = X\theta$.

(b) An orthogonal projection is a linear transformation. Hence, we can define $P_X(y) = Py$ for some projection matrix P. Specifically, given $1 \leq d \leq n$, a matrix $P \in \mathbb{R}^{n \times n}$ is said to be a rank-d orthogonal projection matrix if $\text{rank}(P) = d$, $P = P^\top$ and $P^2 = P$. Prove that P is a rank-d projection matrix if and only if there exists a $U \in \mathbb{R}^{n \times d}$ such that $P = UU^\top$ and $U^\top U = I$

Hint Use the eigendecomposition of P.

(c) Prove that if P is a rank d projection matrix, then $\text{tr}(P) = d$.

(d) The Singular Value Decomposition theorem states that we can write any matrix X as

$$X = \sum_{i=1}^{\min\{n,d\}} \sigma_i u_i v_i^\top = \sum_{i:\sigma_i > 0} \sigma_i u_i v_i^\top$$

where $\sigma_i \geq 0$, and $\{u_i\}$ and $\{v_i\}$ are an orthonormal. Show that

(a) $\{v_i : \sigma_i > 0\}$ are an orthonormal basis for the row space of of X

(b) Similarly, $\{u_i : \sigma_i > 0\}$ are an orthonormal basis for the columnspace of X
Hint: consider $X^\top$.

(e) Prove that if $X \in \mathbb{R}^{n \times d}$ and $\text{rank}(X) = d$, then $X(X^\top X)^{-1}X^\top$ is a rank-d orthogonal projection matrix. What is the corresponding matrix U?

(f) Define the Moore-Penrose pseudoinverse to be the matrix:

$$X^\dagger = \sum_{i:\sigma_i > 0} \sigma_i^{-1} v_i u_i^\top,$$

To what operator does the matrix $X^\dagger X$ correspond? What is $X^\dagger X$ if $\text{rank}(X) = d$? If $\text{rank}(X) = d$ and $n = d$?

3 Convergence for Logistic Regression

In this problem, we will take steps toward proving that gradient descent converges to a unique minimizer of the logistic regression cost function, binary cross-entropy. In particular, we will consider the simplified case where we are minimizing this cost function for a single data point. For weights $w \in \mathbb{R}^d$, data $x \in \mathbb{R}^d$, and a label $y \in \{0, 1\}$, the logistic regression cost function is given by

$$J(w) = -y \log s(x \cdot w) - (1 - y) \log(1 - s(x \cdot w))$$

Where $s(\gamma) = 1/(1 + \exp(-\gamma))$ is the logistic function (also called the sigmoid). You may assume that $x \neq 0$.

(a) To start, write the gradient descent update function $G(w)$, which maps w to the result of a single gradient descent update with learning rate $\epsilon > 0$.

(b) Show that the cost function J has a unique minimizer w^* by proving that J is strictly convex. *Hint: A sufficient condition for a function to by strictly convex is that its Hessian is positive definite.*

(c) Next, show that for a sufficiently chosen ϵ, G is a *contraction*, which means that there is a constant $0 < \rho < 1$ such that, for any $w, w' \in \mathbb{R}^d$, $|G(w) - G(w')| < \rho |w - w'|$. *Hint: this is equivalent to showing that the gradient has bounded norm:* $||\nabla_w G(w)|| < \rho$

(d) Finally, calling $w^{(t)}$ the t-th iterate of gradient descent, show that $|w^* - w^{(t)}| < \rho^t |w^* - w^{(0)}|$, so that $\lim_{t \to \infty} |w^* - w^{(t)}| = 0$.

4 Geometry of Ridge Regression

You recently learned ridge regression and how it differs from ordinary least squares. In this question we will explore how ridge regression is related to solving a constrained least squares problem in terms of their parameters and solutions.

(a) Given a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and a vector $\mathbf{y} \in \mathbb{R}^n$, define the optimization problem

$$\begin{aligned} &\text{minimize } \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2. \\ &\text{subject to } \|\mathbf{w}\|_2^2 \le \beta^2. \end{aligned} \tag{1}$$

We can utilize Lagrange multipliers to incorporate the constraint into the objective function by adding a term which acts to "penalize" the thing we are constraining. Rewrite the constrained optimization problem into an unconstrained optimization problem.

(b) Recall that ridge regression is given by the unconstrained optimization problem

$$\min_{w} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \nu\|\mathbf{w}\|_2^2. \tag{2}$$

One way to interpret "ridge regression" is as the Lagrangian form of a constrained problem. Qualitatively, how would increasing β in our previous problem be reflected in the desired penalty ν of ridge regression (i.e. if our threshold β increases, what should we do to λ)?

(c) One reason why we might want to have small weights $\mathbf{w}$ has to do with the sensitivity of the predictor to its input. Let $\mathbf{x}$ be a d-dimensional list of features corresponding to a new test point. Our predictor is $\mathbf{w}^\top\mathbf{x}$. What is an upper bound on how much our prediction could change if we added noise $\boldsymbol{\epsilon} \in \mathbb{R}^d$ to a test point's features $\mathbf{x}$?

(d) Derive that the solution to ridge regression (2) is given by $\hat{\mathbf{w}}_\mathbf{r} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$. What happens when $\lambda \to \infty$? It is for this reason that sometimes regularization is referred to as "shrinkage."

(e) Note that in computing $\hat{\mathbf{w}}_\mathbf{r}$, we are trying to invert the matrix $\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}$ instead of the matrix $\mathbf{X}^\top\mathbf{X}$. If $\mathbf{X}^\top\mathbf{X}$ has eigenvalues $\sigma_1^2, \ldots, \sigma_d^2$, what are the eigenvalues of $\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}$? Comment on why adding the regularizer term $\lambda\mathbf{I}$ can improve the inversion operation numerically.

(f) Let the number of parameters $d = 3$ and the number of datapoints $n = 5$, and let the eigenvalues of $\mathbf{X}^\top\mathbf{X}$ be given by 1000, 1 and 0.001. We must now choose between two regularization parameters $\lambda_1 = 100$ and $\lambda_2 = 0.5$. Which do you think is a better choice for this problem and why?

(g) Another advantage of ridge regression can be seen for under-determined systems. Say we have the data drawn from a $d = 5$ parameter model, but only have $n = 4$ training samples of it, i.e. $\mathbf{X} \in \mathbb{R}^{4 \times 5}$. Now this is clearly an underdetermined system, since $n < d$. Show that ridge regression with $\lambda > 0$ results in a unique solution, whereas ordinary least squares has an infinite number of solutions.

Hint: To make this point, it may be helpful to consider $\mathbf{w} = \mathbf{w}_0 + \mathbf{w}^*$ where $\mathbf{w}_0$ is in the null space of $\mathbf{X}$ and $\mathbf{w}^*$ is a solution.

(h) For the previous part, what will the answer be if you take the limit $\lambda \to 0$ for ridge regression?

Hint: Think about the SVD of $\mathbf{X}$.

(i) Tikhonov regularization is a general term for ridge regression, where the implicit constraint set takes the form of an ellipsoid instead of a ball. In other words, we solve the optimization problem

$$\mathbf{w} = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\Gamma\mathbf{w}\|_2^2$$

for some full rank matrix $\Gamma \in \mathbb{R}^{d\times d}$. Derive a closed form solution to this problem.

5 Blair and their giant peaches

Make sure to include the code you write for this problem in an appendix

Blair is a mage testing how long they can fly a collection of giant peaches. They have n training peaches – with masses given by $x_1, x_2, \dots x_n$ – and flies these peaches once to collect training data. The experimental flight time of peach i is given by y_i. They believe that the flight time is well approximated by a polynomial function of the mass

$$y_i \approx w_0 + w_1 x_i + w_2 x_i^2 \cdots + w_D x_i^D$$

where their goal is to fit a polynomial of degree D to this data. Include all text responses and plots in your write-up. You can use `Numpy`, `Matplotlib`, and the function `scipy.io.loadmat`.

(a) Show how Blair's problem can be formulated as a linear regression problem.

(b) You are given data of the masses $\{x_i\}_{i=1}^n$ and flying times $\{y_i\}_{i=1}^n$ in the "x_train" and "y_train" keys of the file `1D_poly.mat` with the masses centered and normalized to lie in the range $[-1, 1]$. Write code to perform the least squares fit. You may not use `numpy.linalg.lstsq`, but you can use other linear algebra functions, such as `numpy.linalg.solve`[1]. Letting f_D denote the fitted polynomial, plot the average training error $R(D) = \frac{1}{n}\sum_{i=1}^n (y_i - f_D(x_i))^2$ against D in the range $D \in \{0, 1, 2, 3, \dots, n-1\}$.

(c) How does the average training error behave as a function of D, and why? What happens if you try to fit a polynomial of degree n with a standard matrix inversion method?

(d) Blair has taken CS189 so they decide that they need to run another experiment before deciding that their prediction is true. They run another fresh experiment of flight times using the same peaches, to obtain the data with key "y_fresh" in 1D_POLY.MAT. Denoting the fresh flight time of peach i by $\tilde{y}_i$, plot the average error $\tilde{R}(D) = \frac{1}{n}\sum_{i=1}^n (\tilde{y}_i - f_D(x_i))^2$ for the same values of D as in part (b) using the polynomial approximations f_D also from the previous part. How does this plot differ from the plot in (b) and why?

(e) How do you propose using the two plots from parts (b) and (d) to "select" the right polynomial model for Blair?

(f) Blair has a new hypothesis – the flying time is actually a function of the mass, smoothness, size, and sweetness of the peach, and some multivariate polynomial function of all of these parameters. A D-multivariate polynomial function looks like

$$f_D(\mathbf{x}) = \sum_j \alpha_j \prod_i x_i^{p_{ji}},$$

where $\forall j : \sum_i p_{ji} \leq D$. Here α_j is the scale constant for jth term and p_{ji} is the exponent of x_i in jth term. The data in `polynomial_regression_samples.mat` (100000×5) with columns corresponding to the 5 attributes of the peach. Use 4-fold cross-validation to decide which of

[1] Outside of this class you should use `numpy.linalg.lstsq`.

$D \in \{0, 1, 2, 3, 4, 5\}$ is the best fit for the data provided. Specifically, the best fit is defined as the choice of parameters that has the lowest mean squared error averaged across the four validation folds. For this part, compute the polynomial coefficients via ridge regression with penalty $\lambda = 0.1$, instead of ordinary least squares. You may not use `numpy.linalg.lstsq`, but you can use other linear algebra functions, such as `numpy.linalg.solve`. You should implement the 4-fold cross-validation using only `numpy` (i.e. do not use `sci-kit learn` or another machine learning library). The `assemble_feature` function provided in `hw1_prob5.py` can be used to construct the data matrix representing the D-multivariate polynomial basis functions from the 5 raw features.

Hint: To implement ridge regression using numpy.linalg.solve, consider the matrix equation you get by setting the gradient of the ridge regression loss function to zero.

(g) Now redo the previous part, but use 4-fold cross-validation on all combinations of $D \in \{1, 2, 3, 4, 5\}$ and $\lambda \in \{0.05, 0.1, 0.15, 0.2\}$ - this is referred to as a grid search. Find the best D and λ that best explains the data using ridge regression. Print the average training/validation error per sample for all D and λ.