CS 189:289A Introduction to Machine Learning
Fall 2021 Jennifer Listgarten and Jitendra Malik HW0

Due 09/08/21

- Homework 0 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.
- **Start early. This is a long assignment. Most of the material is prerequisite material not covered in lecture; you are responsible for finding resources to understand it.**

Deliverables:

1. Submit a PDF of your homework to the Gradescope assignment entitled "HW0 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 Probability Potpourri

1. Recall the covariance of two random variables X and Y is defined as $\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$. For a multivariate random variable Z (i.e., each index of Z is a random variable), we define the covariance matrix Σ such that $\Sigma_{ij} = \text{Cov}(Z_i, Z_j)$. Concisely, $\Sigma = \mathbb{E}[(Z - \mu)(Z - \mu)^\top]$, where μ is the mean value of the random column vector Z. Prove that the covariance matrix is always positive semidefinite (PSD).
   Hint: Use linearity of expectation.

2. The probability that an archer hits her target when it is windy is 0.4; when it is not windy, her probability of hitting the target is 0.7. On any shot, the probability of a gust of wind is 0.3. Find the probability that

   (i) on a given shot there is a gust of wind and she hits her target.

   (ii) she hits the target with her first shot.

   (iii) she hits the target exactly once in two shots.

   (iv) there was no gust of wind on an occasion when she missed.

3. An archery target is made of 3 concentric circles of radii $1/\sqrt{3}$, 1 and $\sqrt{3}$ feet. Arrows striking within the inner circle are awarded 4 points, arrows within the middle ring are awarded 3 points, and arrows within the outer ring are awarded 2 points. Shots outside the target are awarded 0 points.

   Consider a random variable X, the distance of the strike from the center (in feet), and let the probability density function of X be

   $$f(x) = \begin{cases} \frac{2}{\pi(1+x^2)} & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

   What is the expected value of the score of a single strike?

4. Let $X \sim \text{Pois}(\lambda)$, $Y \sim \text{Pois}(\mu)$. given that $X \perp\!\!\!\perp Y$, derive an expression for $\mathbb{P}(X \mid X + Y = n)$. What well-known probability distribution is this? What are its parameters?

Solution:

1. For $v \in \mathbb{R}^n$, $v^\top \mathbb{E}[(X-\mu)(X-\mu)^\top]v = \Sigma_{i=1}^n \Sigma_{j=1}^n \mathbb{E}[(X_i-\mu_i)(X_j-\mu_j)]v_i v_j = \mathbb{E}[v^\top (X-\mu)(X-\mu)^\top v] = \mathbb{E}[((X-\mu)^\top v)^2] \geq 0$. Note the second identity comes from linearity of expectation.

2. Denote with H the event that she hits her target, and with W the event that there is a gust of wind. Then we know that: $P(H \mid W) = 0.4$, $P(H \mid W^c) = 0.7$ and $P(W) = 0.3$.

   (i) $P(H \cap W) = P(H \mid W)P(W) = 0.12$

   (ii) $P(H) = P(H \mid W)P(W) + P(H \mid W^c)P(W^c) = 0.61$

   (iii) This probability is equal to $\binom{2}{1} P(H)P(H^c) = 0.4758$

(iv) $P(W^c \mid H^c) = \frac{P(H^c|W^c)P(W^c)}{P(H^c)} = 0.538$

3. The expected value is

$$\begin{aligned}
&\int_0^{1/\sqrt{3}} 4\frac{2}{\pi(1+x^2)}\,\mathrm{d}x + \int_{1/\sqrt{3}}^{1} 3\frac{2}{\pi(1+x^2)}\,\mathrm{d}x + \int_1^{\sqrt{3}} 2\frac{2}{\pi(1+x^2)}\,\mathrm{d}x \\
= \;& \frac{2}{\pi}\left[4\left(\arctan\frac{1}{\sqrt{3}} - \arctan 0\right) + 3\left(\arctan 1 - \arctan\frac{1}{\sqrt{3}}\right) + 2\left(\arctan\sqrt{3} - \arctan 1\right)\right] \\
= \;& \frac{13}{6}
\end{aligned}$$

4. To derive this conditional distribution, we can write

$$P(X = k|X + Y = n) = \frac{P(X = k \cap X + Y = n)}{P(X + Y = n)}$$

using the definition of conditional probability. The event $X = k \cap X + Y = n$ can equivalently be expressed as $X = k \cap Y = n - k$ and we can express this using that $X \perp\!\!\!\perp Y$, i.e.,

$$\begin{aligned}
P(X = k \cap Y = n - k) &= \frac{e^{-\lambda}\lambda^k}{k!}\frac{e^{-\mu}\mu^{n-k}}{(n-k)!} \\
&= \frac{1}{n!}e^{-(\lambda+\mu)}\binom{n}{k}\lambda^k\mu^{n-k}
\end{aligned} \tag{1}$$

Now, we note that we can use the law of total probability with the above to get an expression for the denominator

$$\begin{aligned}
P(X + Y = n) &= \sum_{k=0}^{n} P(X = k \cap Y = n - k) \\
&= \sum_{k=0}^{n} \frac{1}{n!}e^{-(\lambda+\mu)}\binom{n}{k}\lambda^k\mu^{n-k} \\
&= \frac{1}{n!}e^{-(\lambda+\mu)}\sum_{k=0}^{n}\binom{n}{k}\lambda^k\mu^{n-k} \\
&= \frac{1}{n!}e^{-(\lambda+\mu)}(\lambda+\mu)^n
\end{aligned} \tag{2}$$

where the last equality comes from binomial expansion. Lastly, we plug these in to get

$$P(X = k|X + Y = n) = \binom{n}{k}\frac{\lambda^k\mu^{n-k}}{(\lambda+\mu)^n}$$

This is exactly the PMF for a binomial distribution with parameters n and $p = \frac{\lambda}{\lambda+\mu}$.

2 Properties of Gaussians

1. Prove that $\mathbb{E}[e^{\lambda X}] = e^{\sigma^2\lambda^2/2}$, where $\lambda \in \mathbb{R}$ is a fixed constant, and $X \sim N(0, \sigma^2)$. As a function of λ, $\mathbb{E}[e^{\lambda X}]$ is also known as the *moment-generating function*.

2. For $X \sim N(0, \sigma^2)$ and $t > 0$ prove that $\mathbb{P}(X \geq t) \leq \exp\left(-t^2/2\sigma^2\right)$, then show that $\mathbb{P}(|X| \geq t) \leq 2\exp\left(-t^2/2\sigma^2\right)$. *Hint*: Consider using Markov's inequality in combination with the result of the previous part.

3. Let $X_1, \ldots, X_n \sim N(0, \sigma^2)$ be iid (independent and identically distributed). Can you prove a similar concentration result for the average of n Gaussians: $\mathbb{P}(\frac{1}{n}\sum_{i=1}^n X_i \geq t)$? What happens as $n \to \infty$?
   Hint: Without proof, use the fact that linear combinations of iid Gaussian-distributed variables are also Gaussian-distributed. Be warned that summing two Gaussian variables does **not** mean that you can sum their probability density functions.

4. Give an example of two Gaussian-distributed random variables X and Y, such that there exists a linear combination $\alpha X + \beta Y$, for some $\alpha, \beta \in \mathbb{R}$, which is *not* Gaussian-distributed. Note that examples of the kind $X \sim N(0, 1)$, $Y = -X$ and their linear combination $X + Y = 0$ *will not* be valid solutions; we will consider constant random variables as Gaussians with variance equal to 0.

5. Take two orthogonal vectors $u, v \in \mathbb{R}^n$, $u \perp v$, and let $X = (X_1, \ldots, X_n)$ be a vector of n iid standard Gaussians, $X_i \sim N(0, 1), \forall i \in [n]$. Let $u_x = \langle u, X\rangle$ and $v_x = \langle v, X\rangle$. Are u_x and v_x independent?
   Hint: First try to see if they are correlated; you may use the fact that jointly normal random variables are independent iff they are uncorrelated.

6. Prove that $\mathbb{E}\left[\max_{1\leq i\leq n} |X_i|\right] \leq C\sqrt{\log(2n)}\sigma$ for some constant $C \in \mathbb{R}$, where $X_1, \ldots, X_n \sim N(0, \sigma^2)$ are iid. (Interestingly, a similar lower bound holds: $\mathbb{E}\left[\max_{1\leq i\leq n} |X_i|\right] \geq C'\sqrt{\log(2n)}\sigma$ for some C'; but you don't need to prove the lower bound).
   Hint: Use Jensen's inequality: $f(\mathbb{E}[Y]) \leq \mathbb{E}[f(Y)]$ for any convex function f.

Solution:

1.
$$\mathbb{E}[e^{\lambda X}] = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} e^{\lambda x}e^{-x^2/2\sigma^2}dx = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{\lambda\sigma z}e^{-z^2/2}dz$$
$$= e^{\sigma^2\lambda^2/2}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-(z-\lambda\sigma)^2/2}dz = e^{\sigma^2\lambda^2/2}.$$

2. For any $\lambda > 0$, we have:
$$\mathbb{P}(X \geq t) = \mathbb{P}(e^{\lambda X} \geq e^{\lambda t}) \leq e^{-\lambda t}\mathbb{E}[e^{\lambda X}] = e^{-\lambda t}e^{\sigma^2\lambda^2/2},$$

where the inequality applies Markov's inequality. Setting $\lambda = t/\sigma^2$ gives the claim. For the second part, we can proceed with the extended Markov inequality:

$$\mathbb{P}(|X| \geq t) \leq e^{-\lambda t}\mathbb{E}[e^{\lambda|X|}] \leq e^{-\lambda t}\left(\mathbb{E}[e^{\lambda X}] + \mathbb{E}[e^{-\lambda X}]\right) = 2e^{-\lambda t}e^{\sigma^2\lambda^2/2}.$$

Setting $\lambda = t/\sigma^2$ again gives the claim. For this part, a symmetry argument relying on the fact that $\mathbb{P}(X \geq t) = \mathbb{P}(X \leq -t)$ would also suffice.

3. From the hint we know that $\frac{1}{n}\sum_{i=1}^n X_i$ follows a Gaussian distribution, so we only need to determine its mean and variance. Its mean is clearly 0. Its variance, on the other hand, can be computed as follows:

$$\left(\frac{1}{n}\sum_{i=1}^n X_i\right) = \frac{1}{n^2}\, n(X_i) = \frac{\sigma^2}{n},$$

where we use the fact that the variance of a sum of uncorrelated variables separates into a sum of their variances. Now we can apply the concentration result of the previous part to conclude:

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n X_i \geq t\right) \leq \exp\left(-nt^2/2\sigma^2\right).$$

As $n \to \infty$, the probability of the average $\frac{1}{n}\sum_{i=1}^n X_i$ being away from 0 vanishes; this result is a special case of the Weak Law of Large Numbers.

4. Let $X \sim N(0,1)$ and $Y = \xi X$, where ξ takes values in $\{-1, 1\}$ with equal probability. The sum $X + Y$ is not Gaussian, even though both X and Y are Gaussian.

5. We use the fact that Gaussian random variables are independent if and only if they are uncorrelated (again under some regularity which is satisfied). Therefore, we only need to compute the correlation of u_x and v_x:

$$\mathbb{E}[u_x v_x] = \mathbb{E}\left[\left(\sum_{i=1}^n u_i X_i\right)\left(\sum_{i=1}^n v_i X_i\right)\right] = \sum_{i=1}^n u_i v_i \mathbb{E}[X_i^2] = \langle u, v\rangle = 0.$$

Therefore, u_x and v_x are independent. Notice that this is a somewhat paradoxical conclusion, given that both u_x and v_x were computed using the same Gaussian vector X.

6. Let $\lambda > 0$. By Jensen's inequality, we have:

$$\begin{aligned}\lambda\mathbb{E}\left[\max_{1\leq i\leq n}|X_i|\right] &\leq \log\mathbb{E}[e^{\lambda\max_i|X_i|}] \leq \log\sum_{i=1}^n \mathbb{E}[e^{\lambda|X_i|}] \leq \log\sum_{i=1}^n\left(\mathbb{E}[e^{\lambda X_i}] + \mathbb{E}[e^{-\lambda X_i}]\right)\\ &= \log\sum_{i=1}^n 2e^{\sigma^2\lambda^2/2} = \log 2ne^{\sigma^2\lambda^2/2} = \log(2n) + \frac{1}{2}\sigma^2\lambda^2.\end{aligned}$$

Set $\lambda = \frac{\sqrt{\log(2n)}}{\sigma}$:

$$\mathbb{E}\left[\max_{1\leq i\leq n}|X_i|\right] \leq \sigma\sqrt{\log(2n)} + \frac{\sigma}{2}\sqrt{\log(2n)} = \frac{3}{2}\sigma\sqrt{\log(2n)}.$$

3 Linear Algebra Review

1. Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix. Prove equivalence between these three different definitions of positive semidefiniteness (PSD).

   (a) For all $x \in \mathbb{R}^n$, $x^\top A x \geq 0$.

   (b) All the eigenvalues of A are nonnegative.

   (c) There exists a matrix $U \in \mathbb{R}^{n \times n}$ such that $A = UU^\top$.

   Mathematically, we write positive semidefiniteness as $A \succeq 0$.

2. Now that we're equipped with different definitions of positive semidefiniteness, use them to prove the following properties of PSD matrices.

   (a) If A and B are PSD, then $2A + 3B$ is PSD.

   (b) If A is PSD, all diagonal entries of A are nonnegative: $A_{ii} \geq 0, \forall i \in [n]$.

   (c) If A is PSD, the sum of all entries of A is nonnegative: $\sum_{j=1}^{n} \sum_{i=1}^{n} A_{ij} \geq 0$.

   (d) If A and B are PSD, then $\mathrm{Tr}(AB) \geq 0$, where $\mathrm{Tr}(M)$ denotes the *trace* of M.

   (e) If A and B are PSD, then $\mathrm{Tr}(AB) = 0$ if and only if $AB = 0$.

3. Let $A \in \mathbb{R}^{n \times n}$ be a symmetric, PSD matrix. Write $\|A\|_F$ as a function of the eigenvalues of A. *Hint:* Recall that $\|A\|_F = \sqrt{\mathrm{Tr}(A^\top A)}$. If you haven't seen this before, you should try to prove it. However, you can accept this as a given fact for this homework assignment.

Solution:

1. (a) $\Rightarrow$ (b): Let λ be an eigenvalue of A with corresponding eigenvector v. Then:

$$v^\top A v = \lambda v^\top v = \lambda \|v\|^2.$$

   By part (a), we know that $\lambda \|v\|^2 \geq 0$, so $\lambda \geq 0$.
   (b) $\Rightarrow$ (c): Consider the eigendecomposition of A, $A = V \Lambda V^\top$, where Λ is a diagonal matrix with entries equal to the eigenvalues of A, $\lambda_1, \ldots, \lambda_n$. Define $U := V\sqrt{\Lambda}$, where $\sqrt{\Lambda}$ is diagonal with entries equal to $\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n}$; notice that this choice is justified because, by assumption, the eigenvalues are nonnegative. Clearly, $A = UU^\top$.
   (c) $\Rightarrow$ (a): Let $x \in \mathbb{R}^n$, then:

$$x^\top A x = x^\top U U^\top x = (U^\top x)^\top (U^\top x) = \|U^\top x\|^2 \geq 0.$$

2. (a) $x^\top (2A + 3B) x = 2x^\top A x + 3x^\top B x \geq 0$.

   (b) Fix $i \in [n]$. Take $x = e_i$ in the first definition of PSD, where e_i is a canonical vector, i.e. it has zeros everywhere but at coordinate i, where it is equal to 1. Then $e_i^\top A e_i = A_{ii} \geq 0$.

   (c) Take $x = \mathbf{1}$ to be the all-ones vector in the first definition of PSD. Then $\mathbf{1}^\top A \mathbf{1} = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} \geq 0$.

(d) By the third definition of PSD, let $A = UU^\top$ and $B = VV^\top$. Then:

$$\mathrm{Tr}(AB) = \mathrm{Tr}\Big(UU^\top VV^\top\Big) = \mathrm{Tr}\Big(U^\top VV^\top U\Big) = \mathrm{Tr}\Big(U^\top V(U^\top V)^\top\Big) \geq 0,$$

which follows because $M := U^\top V(U^\top V)^\top$ is PSD by the third definition, and $\mathrm{Tr}(M) \geq 0$ by part (b).

(e) If $AB = 0$, then clearly $\mathrm{Tr}(AB) = 0$. To prove the other direction, by the third definition of PSD, let $A = UU^\top$ and $B = VV^\top$, for some U and V. Then:

$$\mathrm{Tr}(AB) = \mathrm{Tr}\Big(UU^\top VV^\top\Big) = \mathrm{Tr}\Big(V^\top UU^\top V\Big) = \mathrm{Tr}\Big((U^\top V)^\top U^\top V\Big),$$

Since $M := (U^\top V)^\top U^\top V$ is PSD, $\mathrm{Tr}(M) = \sum_i \lambda_i(M) = 0$ only if $\lambda_i(M) = 0$ for all $i \in [n]$. From the eigendecomposition of M, it follows that $M = 0$, and moreover this implies $U^\top V = 0$. With this, we have $AB = U(U^\top V)V^\top = U(0)V^\top = 0$.

3. We know the frobenius norm of A can be written as

$$\|A\|_F = \sqrt{tr(A^\top A)}$$

Now, we use that A is PSD, so

$$A^\top A = U\Lambda U^\top U\Lambda U^\top = U\Lambda^2 U^\top$$

By the cyclic property of trace,

$$tr(U\Lambda^2 U^\top) = tr(\Lambda^2 U^\top U) = tr(\Lambda^2)$$

We know Λ is a diagonal matrix containing the eigenvalues of A, i.e., $tr(\Lambda^2) = \sum \lambda^2$ therefore,

$$\|A\|_F = \sqrt{\sum \lambda^2}$$

4 Gradients and Norms

1. Define the ℓ_p-norm as $\|x\|_p = \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$, where $x \in \mathbb{R}^n$.
   Prove that $\frac{1}{\sqrt{n}}\|x\|_2 \le \|x\|_\infty \le \|x\|_1 \le \sqrt{n}\|x\|_2$. The Cauchy–Schwarz inequality is helpful here.

2. (a) Let $\alpha = \sum_{i=1}^n y_i \ln \beta_i$ for $y, \beta \in \mathbb{R}^n$. What are the partial derivatives $\frac{\partial \alpha}{\partial \beta_i}$?
   (b) Let $\beta = \sinh \gamma$ for $\gamma \in \mathbb{R}^n$ (treat the sinh as an element-wise operation; i.e. $\beta_i = \sinh \gamma_i$). What are the partial derivatives $\frac{\partial \beta_i}{\partial \gamma_j}$?
   (c) Let $\gamma = A\rho + b$ for $b \in \mathbb{R}^n$, $\rho \in \mathbb{R}^m$, $A \in \mathbb{R}^{n\times m}$. What are the the partial derivatives $\frac{\partial \gamma_i}{\partial \rho_j}$?
   (d) Let $f(x) = \sum_{i=1}^n y_i \ln(\sinh(Ax+b)_i)$; $A \in \mathbb{R}^{n\times m}$, $y \in \mathbb{R}^n$, $b \in \mathbb{R}^n$ are given. What are the partial derivatives $\frac{\partial f}{\partial x_j}$?
   Hint: Use the chain rule.

3. Consider a linear decision function $f(x) = w \cdot x + \alpha$ and the hyperplane decision boundary $H = \{x : w \cdot x = -\alpha\}$. Prove that if w is a unit vector, then the *signed distance* (the ℓ_2-norm distance with an appropriate sign) from x to the closest point on H is $w \cdot x + \alpha$.

4. Consider the function which maps a vector to its maximum entry, $x \mapsto \max_i x_i$. While this function is non-smooth, a common trick in machine learning is to use a smooth approximation, *LogSumExp*, defined as follows.

$$\text{LSE} : \mathbb{R}^n \to \mathbb{R}, \text{LSE}(x) = \ln\left(\sum_{i=1} e^{x_i}\right).$$

   One of the nice properties of this function is that it is convex, which can be proved by showing its Hessian matrix is positive semidefinite. To that end, compute its gradient and Hessian. You do not need to prove that the Hessian is PSD.

5. Let $X \in \mathbb{R}^{n\times d}$ be a data matrix, consisting of n samples, each of which has d features, and let $y \in \mathbb{R}^n$ be a vector of outcomes. We wish to find the *best linear approximation*, i.e. we want to find the θ that minimizes the loss $L(\theta) = \|y - X\theta\|_2^2$. Assuming X has full column rank, compute $\theta^* = \text{argmin}_\theta L(\theta)$ in terms of X and y.

Solution:

1. First, we show: $\frac{1}{\sqrt{n}}\|x\|_2 \le \|x\|_\infty$

$$\frac{1}{\sqrt{n}}\|x\|_2 = \frac{1}{\sqrt{n}}\sqrt{\sum x_i^2} = \sqrt{\frac{1}{n}\sum x_i^2} \le \sqrt{\frac{1}{n}\sum \max x_i^2} = \sqrt{\max x_i^2} = \max \sqrt{x_i^2} = \max \|x_i\| = \|x\|_\infty$$

Next: $\|x\|_\infty \leq \|x\|_1$

$$\|x\|_\infty = \max_i |x_i| \leq \sum_1^n |x_i| = \|x\|_1$$

Lastly: $\|x\|_1 \leq \sqrt{n}\|x\|_2$
From the Cauchy-Schwarz theorem, $| < x, y > |^2 \leq \|x\|_2^2 \|y\|_2^2$. Let $y = \text{sgn}(x)$. Then we have

$$| < x, \text{sgn}(x) > |^2 \leq \|x\|_2^2 \|\text{sgn}(x)\|_2^2 \Leftrightarrow (\sum_i |x_i|)^2 \leq (\sum_i {x_i}^2)(\sum_i \text{sgn}(x_i)^2)$$

Since $\sum_i \text{sgn}(x_i)^2 \leq \sum_i 1 = n$ we get

$$\|x\|_1^2 \leq n \cdot \|x\|_2^2 \Leftrightarrow \|x\|_1 \leq \sqrt{n} \cdot \|x\|_2$$

Thus, we have shown $\frac{1}{\sqrt{n}}\|x\|_2 \leq \|x\|_\infty \leq \|x\|_1 \leq \sqrt{n}\|x\|_2$

2. (a) $\frac{\partial \alpha}{\partial \beta_i} = \sum_{j=1}^n \frac{\partial (y_j \ln \beta_j)}{\partial \beta_i} = \frac{y_i}{\beta_i}$

   (b) $\frac{\partial \beta_i}{\gamma_j} = \begin{cases} 0 & i \neq j \\ \cosh(\gamma_j) & i = j \end{cases}$

   (c) $\frac{\partial \gamma_i}{\partial \rho_j} = A_{ij}$

   (d) Using the previous parts, we can apply the chain rule as $\frac{\partial f}{\partial x_j} = \Sigma_{k=1}^n \Sigma_{l=1}^n \frac{\partial f}{\partial \beta_k} \frac{\partial \beta_k}{\partial \gamma_l} \frac{\partial \gamma_l}{\partial x_j}$. This can be simplified using the result from (b) to see the partial derivative $\frac{\partial \beta_a}{\partial \gamma_b}$ is zero unless $k = l$. This yields $\Sigma_{k=1}^n \frac{\partial f_i}{\partial \beta_k} \frac{\partial \beta_k}{\partial \gamma_k} \frac{\partial \gamma_k}{\partial x_j}$. Then we can expand and substitute in to get $\Sigma_{k=1}^n \frac{y_k}{\sinh(Ax+b)_k} \cosh((Ax+b)_k) A_{kj} = A_j^T (y \circ \frac{\cosh(Ax+b)}{\sinh(Ax+b)}) = A_j^T (y \circ \coth(Ax+b))$.

3. Let $\bar{x}$ be the closest point on H to x; the distance from x to $\bar{x}$ is given by $\|x - \bar{x}\|_2$. We know that from the projection theorem, $x - \bar{x}$ is orthogonal to H, and the direction orthogonal to H is given exactly by w as that is the definition of H. Therefore, we can say that $x - \bar{x} = \kappa w$ for some κ. We note that w is unit norm, so it now suffices to find κ to define the sought distance. We also know that $\bar{x}$ lies on H and so must satisfy $w \cdot \bar{x} = -\alpha$. We substitute $\bar{x} = x - \kappa w$ into that equation to get

   $$w \cdot (x - \kappa w) = -\alpha$$

   we distribute using the fact that κ is a scaling factor

   $$w \cdot x - \kappa w \cdot w = -\alpha$$

   Now, we again use that w is unit norm that gives us

   $$w \cdot x + \alpha = \kappa$$

   So we have shown the signed distance is $w \cdot x + \alpha$.

4. We can first compute the gradient by finding each of the partials by applying the chain rule. Notice that the gradient is actually the softmax function, which we will see soon in the class.

$$\begin{aligned}\frac{\partial}{\partial x_k}\operatorname{LSE}(x) &= \frac{1}{\sum_{i=1}^n e^{x_i}} \cdot \frac{\partial}{\partial x_k}\sum_{i=1}^n e^{x_i} \\ &= \frac{e^{x_k}}{\sum_{i=1}^n e^{x_i}}\end{aligned}$$

$$\therefore \nabla_x \operatorname{LSE}(x) = \frac{1}{\sum_{i=1}^n e^{x_i}}\begin{bmatrix} e^{x_1} \\ e^{x_2} \\ \vdots \\ e^{x_n}\end{bmatrix}$$

Now we compute each entry of the Hessian. To that end, notice that the diagonal and off-diagonal entries will have different values.

$$\begin{aligned}\frac{\partial}{\partial x_l}\frac{\partial}{\partial x_k}\operatorname{LSE}(x) &= \frac{\partial}{\partial x_l}\frac{e^{x_k}}{\sum_{i=1}^n e^{x_i}} \\ &= \frac{1}{\left(\sum_{i=1}^n e^{x_i}\right)^2}\left[\sum_{i=1}^n e^{x_i}\cdot\frac{\partial}{\partial x_l}e^{x_k} - e^{x_k}\cdot\frac{\partial}{\partial x_l}\sum_{i=1}^n e^{x_i}\right] \\ &= \frac{1}{\left(\sum_{i=1}^n e^{x_i}\right)^2}\left[\sum_{i=1}^n e^{x_i}\cdot\frac{\partial}{\partial x_l}e^{x_k} - e^{x_k}\cdot e^{x_l}\right] \\ &= \frac{1}{\sum_{i=1}^n e^{x_i}}\cdot\frac{\partial}{\partial x_l}e^{x_k} - \frac{e^{x_k}}{\sum_{i=1}^n e^{x_i}}\cdot\frac{e^{x_l}}{\sum_{i=1}^n e^{x_i}}\end{aligned}$$

To simplify this, notice that that the derivative of e^{x_k} with respect to x_l is 0 when $l \neq k$, since it is a constant. Hence the first term is non-zero only along the diagonal. Therefore:

$$\left[\nabla_x^2 \operatorname{LSE}(x)\right]_{kl} = \begin{cases}\frac{e^{x_k}}{\sum_{i=1}^n e^{x_i}} - \frac{e^{x_k}}{\sum_{i=1}^n e^{x_i}}\cdot\frac{e^{x_k}}{\sum_{i=1}^n e^{x_i}} & : k = l \\ -\frac{e^{x_k}}{\sum_{i=1}^n e^{x_i}}\cdot\frac{e^{x_l}}{\sum_{i=1}^n e^{x_i}} & : k \neq l\end{cases}.$$

If we let $z = \nabla_x \operatorname{LSE}(x)$, we can compactly express the hessian as the difference between a diagonal matrix and outer product below.

$$\nabla_x^2 \operatorname{LSE}(x) = \operatorname{diag}(z) - zz^\top$$

5. The loss is convex, so we find the optimizer θ^* by finding a stationary point of $L(\theta)$. This gives:

$$\nabla_\theta L(\theta) = -2X^\top(y - X\theta) = 0,$$

or in other words $X^\top y = X^\top X\theta$. Since X has full column rank, $X^\top X$ is invertible, and so $\theta^* = (X^\top X)^{-1}X^\top y$.

5 Isocontours of Normal Distributions

Let $f(\mu, \Sigma)$ be the probability density function of a normally distributed random variable in $\mathbb{R}^2$. Write code to plot the isocontours of the following functions, each on its own separate figure. You are free to use Matplotlib, NumPy, and SciPy.

(a) $f(\mu, \Sigma)$, where $\mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$.

(b) $f(\mu, \Sigma)$, where $\mu = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}$.

(c) $f(\mu_1, \Sigma_1) - f(\mu_2, \Sigma_2)$, where $\mu_1 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$ and $\Sigma_1 = \Sigma_2 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$.

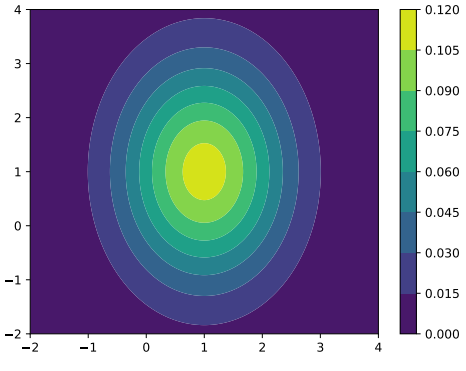
(d) $f(\mu_1, \Sigma_1) - f(\mu_2, \Sigma_2)$, where $\mu_1 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$, $\Sigma_1 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$ and $\Sigma_2 = \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}$.

(e) $f(\mu_1, \Sigma_1) - f(\mu_2, \Sigma_2)$, where $\mu_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$, $\Sigma_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$ and $\Sigma_2 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$.

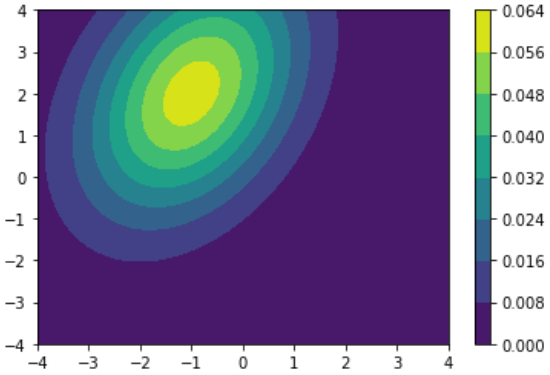
Solution:

```
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats

def plot_contours():
    fig = plt.figure(figsize=(10,10))
    ax0 = fig.add_subplot(111)
    ax0.contour(rv.pdf(pos).reshape(500,500))
    plt.show()

# Part a

# Generate grid of points at which to evaluate pdf
x = np.linspace(-2, 4, 500)
y = np.linspace(-2, 4, 500)
X,Y = np.meshgrid(x, y)
pos = np.array([Y, X]).T
rv = scipy.stats.multivariate_normal([1, 1], [[1, 0], [0, 2]])
Z = rv.pdf(pos)

plt.contourf(X, Y, Z)
plt.colorbar()
plt.show()

# Part b
x = np.linspace(-4, 4, 500)
y = np.linspace(-4, 4, 500)
X,Y = np.meshgrid(x, y)
pos = np.array([Y, X]).T
rv = scipy.stats.multivariate_normal([-1, 2], [[2, 1], [1, 4]])
Z = rv.pdf(pos)

plt.contourf(X, Y, Z)
plt.colorbar()
plt.show()
```

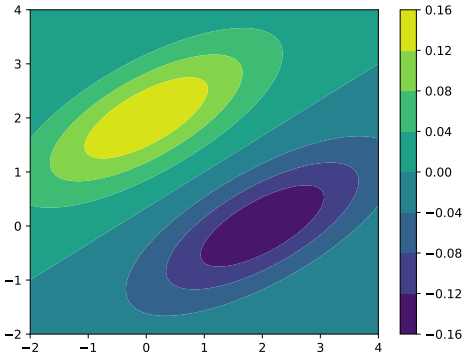
```
# Part c
x = np.linspace(-2, 4, 500)
y = np.linspace(-2, 4, 500)
X,Y = np.meshgrid(x, y)
pos = np.array([Y, X]).T
rv1 = scipy.stats.multivariate_normal([0, 2], [[2, 1], [1, 1]])
rv2 = scipy.stats.multivariate_normal([2, 0], [[2, 1], [1, 1]])
Z = rv1.pdf(pos) - rv2.pdf(pos)

plt.contourf(X, Y, Z)
plt.colorbar()
plt.show()

# Part d
x = np.linspace(-2, 4, 500)
y = np.linspace(-2, 4, 500)
X,Y = np.meshgrid(x, y)
pos = np.array([Y, X]).T
rv1 = scipy.stats.multivariate_normal([0, 2], [[2, 1], [1, 1]])
rv2 = scipy.stats.multivariate_normal([2, 0], [[2, 1], [1, 4]])
Z = rv1.pdf(pos) - rv2.pdf(pos)

plt.contourf(X, Y, Z)
plt.colorbar()
plt.show()

# Part e
x = np.linspace(-3, 3, 500)
y = np.linspace(-3, 3, 500)
X,Y = np.meshgrid(x, y)
pos = np.array([Y, X]).T
rv1 = scipy.stats.multivariate_normal([1, 1], [[2, 0], [0, 1]])
rv2 = scipy.stats.multivariate_normal([-1, -1], [[2, 1], [1, 2]])
Z = rv1.pdf(pos) - rv2.pdf(pos)

plt.contourf(X, Y, Z)
plt.colorbar()
plt.show()
```



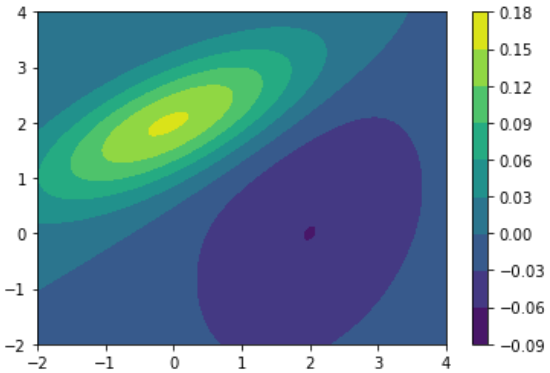
(a)



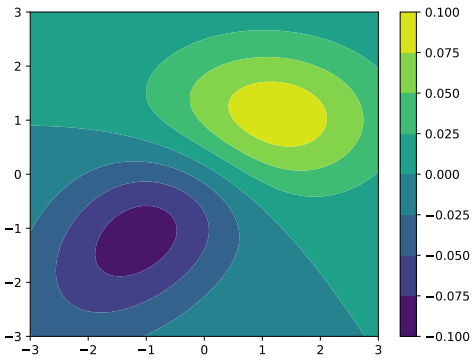
(b)



(c)



(d)



(e)

6 Eigenvectors of the Gaussian Covariance Matrix

Consider two one-dimensional random variables $X_1 \sim \mathcal{N}(3, 9)$ and $X_2 \sim \frac{1}{2}X_1 + \mathcal{N}(4, 4)$, where $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2. Write a program that draws $n = 100$ random two-dimensional sample points from (X_1, X_2) such that the ith value sampled from X_2 is calculated based on the ith value sampled from X_1. In your code, make sure to choose and set a fixed random number seed for whatever random number generator you use, so your simulation is reproducible, and document your choice of random number seed and random number generator in your write-up. For each of the following parts, include the corresponding output of your program.

(a) Compute the mean (in $\mathbb{R}^2$) of the sample.

(b) Compute the 2×2 covariance matrix of the sample.

(c) Compute the eigenvectors and eigenvalues of this covariance matrix.

(d) On a two-dimensional grid with a horizonal axis for X_1 with range $[-15, 15]$ and a vertical axis for X_2 with range $[-15, 15]$, plot

  (i) all $n = 100$ data points, and

  (ii) arrows representing both covariance eigenvectors. The eigenvector arrows should originate at the mean and have magnitudes equal to their corresponding eigenvalues.

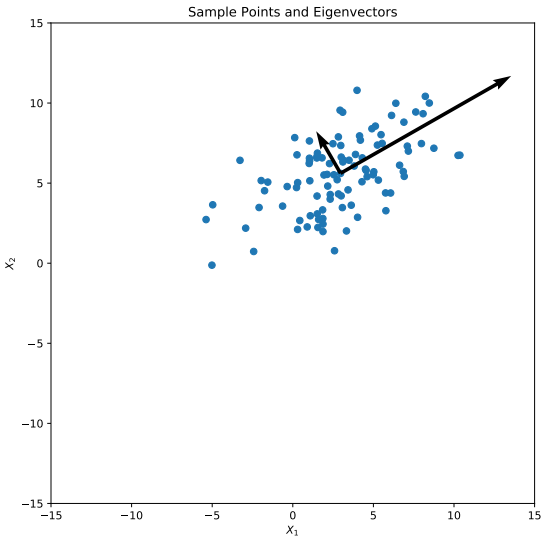
(e) Let $U = [v_1 \quad v_2]$ be a 2×2 matrix whose columns are the eigenvectors of the covariance matrix, where v_1 is the eigenvector with the larger eigenvalue. We use $U^\top$ as a rotation matrix to rotate each sample point from the (X_1, X_2) coordinate system to a coordinate system aligned with the eigenvectors. (As $U^\top = U^{-1}$, the matrix U reverses this rotation, moving back from the eigenvector coordinate system to the original coordinate system). *Center* your sample points by subtracting the mean μ from each point; then rotate each point by $U^\top$, giving $x_{\text{rotated}} = U^\top(x - \mu)$. Plot these rotated points on a new two dimensional-grid, again with both axes having range $[-15, 15]$.

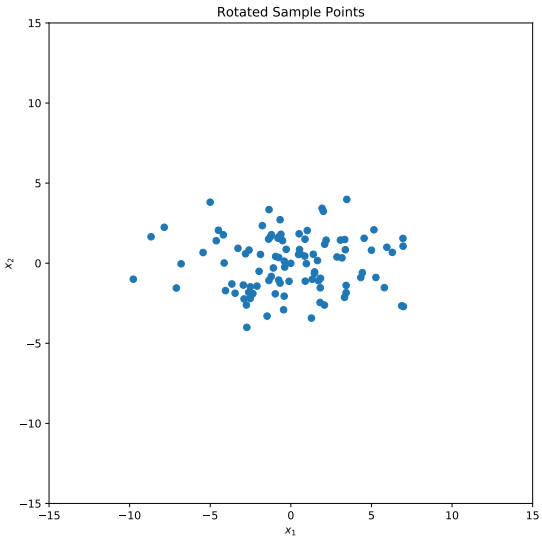
In your plots, **clearly label the axes and include a title**. Moreover, **make sure the horizontal and vertical axis have the same scale!** The aspect ratio should be one. You are free to use Matplotlib and NumPy.

Solution:

```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(9)

X = np.random.normal(loc=3, scale=3, size=100)
Y = np.random.normal(loc=4, scale=2, size=100)
sample = np.array([np.array((x, 0.5 * x + y)) for (x, y) in zip(X, Y)])

# Part a (compute the sample mean)
sample_mean = np.mean(sample, axis=0)
print('Sample Mean = {0}'.format(sample_mean))
```

```
#Sample Mean = [2.96143749 5.61268062]

# Part b (compute the sample covariance matrix)
sample_cov = np.cov(sample.T)
print('Sample Covariance')
print(sample_cov)

#Sample Covariance
#[[9.93191037 3.96365428]
# [3.96365428 5.30782634]]

# Part c (compute the eigenvalues and eigenvectors)
eigen_values, eigen_vectors = np.linalg.eig(sample_cov)
print('Eigenvalues = {0}'.format(eigen_values))
print('Eigenvectors (columns)')
print(eigen_vectors)

#Eigenvalues = [12.20856027  3.03117644]
#Eigenvectors (columns)
# [[ 0.86713795 -0.49806804]
#  [ 0.49806804  0.86713795]]


# Part d (plot data and eigenvectors scaled by eigenvalues)
plt.figure(figsize=(8, 8))
plt.scatter(sample[:, 0], sample[:, 1])
plt.xlim(-15, 15)
plt.ylim(-15, 15)
plt.xlabel(r"$X_1$")
plt.ylabel(r"$X_2$")
plt.title("Sample Points and Eigenvectors")
vec_X = [sample_mean[0], sample_mean[0]]
vec_Y = [sample_mean[1], sample_mean[1]]
vec_U = [eigen_vectors[0][0] * eigen_values[0], eigen_vectors[0][1] * eigen_values[1]]
vec_V = [eigen_vectors[1][0] * eigen_values[0], eigen_vectors[1][1] * eigen_values[1]]
plt.quiver(vec_X, vec_Y, vec_U, vec_V, angles="xy", scale_units="xy", scale=1)
plt.show()

# Part e (plot rotated data in coorinate system defined by eigenvectors)
rotated = np.dot(eigen_vectors.T, (sample - sample_mean).T).T
plt.figure(figsize=(8, 8))
plt.scatter(rotated[:, 0], rotated[:, 1])
plt.xlim(-15, 15)
plt.ylim(-15, 15)
plt.xlabel(r"$x_1$")
plt.ylabel(r"$x_2$")
plt.title("Rotated Sample Points")
plt.show()
```


Sample Points and Eigenvectors
15
10
5
0
−5
−10
−15
−15
−10
−5
0
5
10
15
X_1
X_2


Rotated Sample Points
15
10
5
0
−5
−10
−15
−15
−10
−5
0
5
10
15
x_1
x_2