CS 189/289A Introduction to Machine Learning
Fall 2024 Jennifer Listgarten, Saeed Saremi

HW4

Due 10/25/24 11:59 pm PT

- Homework 4 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 "HW 4 Written". **Please start each question on a new page.** If there are graphs, include those graphs in the correct sections. **Do not** just stick the graphs in the appendix. We need each solution to be self-contained on pages of its own.
    - **Replicate all of your code in an appendix**. Begin code for each coding question on 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 the appendix to correct questions.
2. Submit all the code needed to reproduce your results to the Gradescope assignment entitled "HW 4 Code". Yes, you must submit your code twice: in your PDF write-up following the directions as described above so the readers can easily read it, and once in the format described below for ease of reproducibility.
    - You must set **random seeds** for all random utils to ensure reproducibility.
    - Do **NOT** submit any data files that we provided.
    - Please also include a short file named README listing your name, student ID, and instructions on how to reproduce your results.
    - Please take care that your code doesn't take up inordinate amounts of time or memory.

1 Derivation of PCA

Assume we are given n training data points $(\mathbf{x}_i, y_i)$. We collect the target values into $\mathbf{y} \in \mathbb{R}^n$, and the inputs into the matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ where the rows are the $d-$dimensional feature vectors $\mathbf{x}_i^\top$ corresponding to each training point. Furthermore, assume that the data has been centered such that $\frac{1}{n}\sum_{i=1}^n \mathbf{x_i} = \mathbf{0}$, $n > d$ and $\mathbf{X}$ has rank d. The covariance matrix is given by

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$$

When $\bar{\mathbf{x}} = 0$ (i.e., we have subtracted the mean in our samples), we obtain $\Sigma = \frac{1}{n}\mathbf{X}^\top\mathbf{X}$. We will assume this to be the case for this problem.

(a) Maximum Projected Variance: We would like the vector $\mathbf{w}$ such that projecting your data onto $\mathbf{w}$ will retain the maximum amount of information, i.e., variance. We can formulate the optimization problem as

$$\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}_i^\top\mathbf{w}\right)^2 = \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n}\mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w}\,. \tag{1}$$

Show that the maximizer for this problem is equal to the eigenvector $\mathbf{v}_1$ that corresponds to the largest eigenvalue λ_1 of Σ. Also show that the optimal value of this problem is equal to λ_1.

Hint: Use the spectral decomposition of Σ and consider reformulating the optimization problem using a new variable.

Solution:

We start by invoking the spectral decomposition of $\Sigma = \mathbf{V}\Lambda\mathbf{V}^\top$, which is a symmetric positive semi-definite matrix.

$$\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n}\mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w} = \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \mathbf{w}^\top\mathbf{V}\Lambda\mathbf{V}^\top\mathbf{w} = \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} (\mathbf{V}^\top\mathbf{w})^\top\Lambda\mathbf{V}^\top\mathbf{w} \tag{2}$$

Define a new variable $\mathbf{z} = \mathbf{V}^\top\mathbf{w}$, and maximize over this variable. Note that because $\mathbf{V}$ is invertible, there is a one to one mapping between $\mathbf{w}$ and $\mathbf{z}$. Also note that the constraint is the same because the length of the vector $\mathbf{w}$ does not change when multiplied by an orthogonal matrix.

$$\max_{\mathbf{z}:\|\mathbf{z}\|_2=1} \mathbf{z}^\top\Lambda\mathbf{z} = \max_{\mathbf{z}:\|\mathbf{z}\|_2=1} \sum_{i=1}^{d}\lambda_i z_i^2$$

From this new formulation, we can see that we can maximize this by "throwing all of our eggs into one basket"; that is, setting $z_i^* = 1$ if i is the index of the largest eigenvalue, and $z_i^* = 0$ otherwise. Note that, under our constraint that the norm of z must be 1, this maximizes our value: if we were to reduce the value of z_i that corresponds to the largest eigenvalue and assign that value to a different z_j with a smaller or equal eigenvalue, we would get a value that is strictly less than or equal to setting z_i to 1. In other words, $\mathbf{z}$ is a one hot vector. Thus,

$$\mathbf{z}^* = \mathbf{V}^\top\mathbf{w}^* \implies \mathbf{w}^* = \mathbf{V}\mathbf{z}^* = \mathbf{v}_1$$

where $\mathbf{v}_1$ is the principal eigenvector and corresponds to λ_1. Plugging this into the objective function, we see that the optimal value is λ_1.

(b) Let us call the solution of the above part $\mathbf{w}^{(1)}$. Next, we will use a *greedy procedure* to find the ith component of PCA by doing the following optimization

$$\begin{aligned} \text{maximize} \quad & \tfrac{1}{n}(\mathbf{w}^{(i)})^\top \mathbf{X}^\top \mathbf{X}\mathbf{w}^{(i)} \\ \text{subject to} \quad & \|\mathbf{w}^{(i)}\|_2 = 1 \\ & (\mathbf{w}^{(i)})^\top \mathbf{w}^{(j)} = 0 \quad \forall j < i, \end{aligned} \tag{3}$$

where the $\mathbf{w}^{(j)}$ vectors for all $j < i$ are defined recursively using the same maximization procedure above. Show, using your work in the previous part, that the maximizer for this problem is equal to the eigenvector $\mathbf{v}_i$ that corresponds to the ith eigenvalue λ_i of Σ. Also show that optimal value of this problem is equal to λ_i.

Solution: Again use the spectral decomposition of $\Sigma = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, and let $\mathbf{z}^{(i)} = \mathbf{V}^T\mathbf{w}^{(i)}$. We can use the same strategy as from the previous part to write the optimization problem as

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{\ell=1}^{d} \lambda_\ell (z_\ell^{(i)})^2 \\ \text{subject to} \quad & \|\mathbf{z}^{(i)}\|_2 = 1 \\ & (\mathbf{z}^{(i)})^\top \mathbf{z}^{(j)} = 0 \quad \forall j < i, \end{aligned} \tag{4}$$

Note that the last constraint comes from the fact that

$$(\mathbf{w}^{(i)})^\top \mathbf{w}^{(j)} = 0 \implies (\mathbf{V}z^{(i)})^T(\mathbf{V}\mathbf{z}^{(j)}) = 0 \implies (\mathbf{z}^{(i)})^T\mathbf{V}^T\mathbf{V}\mathbf{z}^{(j)} = 0 \implies (\mathbf{z}^{(i)})^T\mathbf{z}^{(j)} = 0.$$

We see that we can maximize this by throwing all of our eggs into one basket, as explained in the previous part, and setting $z_\ell^{(i)} = 1$ if ℓ is the index of the ith largest eigenvalue and others to 0. Because $\mathbf{z}^{(i)}$ is a one-hot vector corresponding to the ith largest eigenvalue, it would also be orthogonal to $\mathbf{z}^{(j)}$ for all $j < i$. Plugging this into the objective function, we see that the optimal value is λ_i.

2 PCA and Least Squares

Consider the ridge regression estimator,

$$\widehat{w}_{\text{ridge}} := \arg\min_{w \in \mathbb{R}^d} \|Xw - y\|_2^2 + \lambda \|w\|_2^2,$$

where $X \in \mathbb{R}^{n \times d}$ and $y \in \mathbb{R}^n$. Suppose that X has already been centered and has singular value decomposition $X = U\Sigma V^\top = \sum_{i=1}^d \sigma_i u_i v_i^\top$, where $U \in \mathbb{R}^{n \times d}$, $\Sigma \in \mathbb{R}^{d \times d}$, and $V \in \mathbb{R}^{d \times d}$, and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d \geq 0$ are the diagonal components of Σ.

(a) Show that

$$\widehat{w}_{\text{ridge}} = \sum_{i=1}^d \rho_\lambda(\sigma_i) v_i u_i^\top y$$

for some function $\rho_\lambda(\sigma)$ that you will determine. What is $\rho_\lambda(\sigma)$ for $\widehat{w}_{\text{ridge}}$? What is $\rho_\lambda(\sigma)$ for $\widehat{w}_{\text{OLS}} = \arg\min_w \|Xw - y\|_2^2$?

Solution: The ridge regression solution is

$$\begin{aligned}
\widehat{w}_{\text{ridge}} &= (X^\top X + \lambda I)^{-1} X^\top y \\
&= (V\Sigma U^\top U \Sigma V^\top + \lambda I)^{-1} V \Sigma U^\top y \\
&= (V\Sigma^2 V^\top + \lambda I)^{-1} V\Sigma U^\top y \\
&= (V\Sigma^2 V^\top + \lambda I)^{-1} \sum_{i=1}^d \sigma_i v_i u_i^\top y \\
&= \sum_{i=1}^d \sigma_i (V\Sigma^2 V^\top + \lambda I)^{-1} v_i u_i^\top y.
\end{aligned}$$

We now seek to simplify $(V\Sigma^2 V^T + \lambda I)^{-1} v_i$. We first note that

$$\begin{aligned}
(V\Sigma^2 V^T + \lambda I)^{-1} &= [V(\Sigma^2 + \lambda I)V^T]^{-1} \\
&= V(\Sigma^2 + \lambda I)^{-1} V^T
\end{aligned}$$

Applying the above inverse to v_i, we get

$$\begin{aligned}
V(\Sigma^2 + \lambda I)^{-1} V^T v_i &= V(\Sigma^2 + \lambda I)^{-1} e_i \\
&= V\left(\frac{1}{\sigma_i^2 + \lambda} e_i\right) \\
&= \frac{1}{\sigma_i^2 + \lambda} v_i
\end{aligned}$$

Using this result, we get our answer for $\widehat{w}_{\text{ridge}}$:

$$\widehat{w}_{\text{ridge}} = \sum_{i=1}^d \frac{\sigma_i}{\sigma_i^2 + \lambda} v_i u_i^\top y.$$

Hence $\rho_\lambda(\sigma) = \frac{\sigma_i}{\sigma_i^2+\lambda}$.

(Note that for the special case where $\lambda = 0$, i.e., ordinary least squares, $\rho_\lambda(\sigma) = \frac{1}{\sigma_i}$.)

(b) The ordinary least squares regression problem on the reduced k-dimensional PCA feature space (PCA-OLS) can be written

$$\hat{w}_{\text{PCA}} = \arg\min_{w\in\mathbb{R}^k} \|XV_k w - y\|^2,$$

where $V_k \in \mathbb{R}^{d\times k}$ is a matrix whose columns are the first k right singular vectors of X. This expression embeds the raw feature vectors onto the top k principal components by the transformation $V_k^\top x_i$. Assume the PCA dimension is less than the rank of the data matrix, $k \le r$, which implies that the matrix of PCA embedded data matrix XV_k has full rank.

(i) Write down the expression for the optimizer $\hat{w}_{\text{PCA}} \in \mathbb{R}^k$ in terms of U, y and the singular values of $\mathbf{X}$.

Hint: Just as V_k is a "shortened" version of V, you may want to use shortened versions of U and Σ. Knowing that $V^\top V = I$, what is the value of $V_k^\top V$?

Solution: The solution of the least-squares linear regression problem on the new matrix XV_k is

$$\begin{aligned}
\hat{w}_{\text{PCA}} &= \left((XV_k)^\top (XV_k)\right)^{-1} (XV_k)^\top y \\
&= (V_k^\top V\Sigma U^\top U\Sigma V^\top V_k)^{-1} V_k^\top V\Sigma U^\top y \\
&= (I_{k\times d}\Sigma^2 I_{d\times k})^{-1} I_{k\times d}\Sigma U^\top y \\
&= \Sigma_k^{-2}\Sigma_k U_k^\top y \\
&= \Sigma_k^{-1} U_k^\top y.
\end{aligned}$$

(ii) Note that the $\hat{w}_{\text{PCA}} \in \mathbb{R}^k$ you computed above is the vector of features applied to matrix XV_k. The actual features applied to X is the vector $V_k\hat{w}_{\text{PCA}} \in \mathbb{R}^d$. Rewrite $V_k\hat{w}_{\text{PCA}}$ in summation form similar to that in part (a).

Solution:

$$\begin{aligned}
V_k\hat{w}_{\text{PCA}} &= V_k\Sigma_k^{-1}U_k^\top y \\
&= \sum_{i=1}^{k} \frac{1}{\sigma_i} v_i u_i^\top y
\end{aligned}$$

(c) Compare the functions of σ you derived and what value of λ leads to $\hat{w}_{\text{OLS}}$ vs. $\hat{w}_{\text{ridge}}$ vs. $\hat{w}_{\text{PCA}}$. How do ridge regression and PCA-OLS deal with overfitting?

Hint: Penalizing certain types of singular values σ_i is a way to deal with overfitting.

Solution:

(a) If $\lambda > 0$, the function corresponds to ridge regression.

(b) If $\lambda = 0$, ridge regression degenerates to ordinary least squares.

(c) If $\lambda = 0$ for the first k components and $\lambda = \infty$ for the rest, the function corresponds to PCA-OLS (PCA regression).

For ridge regression, small singular values σ_i are penalized more than large ones. Similarly for PCA-OLS, large singular values are kept intact, while small ones (after certain number k) are completely removed. This means that the ridge regression can be thought of as a "smooth version" of PCA regression.

Overfitting happens when the singular values are very small and hence the weights in w become very large. Both ridge regression and PCA-OLS solve that problem by making sure that directions with small singular values do not affect the value of w that much.

3 Random Feature Embeddings

In this question, we revisit the task of dimensionality reduction. Dimensionality reduction is useful for several purposes, including visualization, storage, faster computation, etc. We can formalize dimensionality reduction as an embedding function, or *embedding*, $\psi : \mathbb{R}^d \to \mathbb{R}^k$, which maps data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ with d-dimensional features to reduced data points $\psi(\mathbf{x}_1), \ldots, \psi(\mathbf{x}_n)$ with k-dimensional features.

For the reduced data to remain useful, it may be necessary for the reductions to preserve some properties of the original data. Often, geometric properties like distance and inner products are important for machine learning tasks. And as a result, we may want to perform dimensionality reduction while ensuring that we approximately maintain the pairwise distances and inner products.

While you have already seen many properties of PCA so far, in this question we investigate whether random feature embeddings are a good alternative for dimensionality reduction. A few advantages of random feature embeddings over PCA can be: (1) PCA is expensive when the underlying dimension is high and the number of principal components is also large (however note that there are several very fast algorithms dedicated to doing PCA), (2) PCA requires you to have access to the feature matrix for performing computations. The second requirement of PCA is a bottleneck when you want to take only a low dimensional measurement of a very high dimensional data, e.g., in FMRI and in compressed sensing. In such cases, one needs to design an embedding scheme before seeing the data. We now turn to a concrete setting to study a few properties of PCA and random feature embeddings.

Suppose you are given n points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in $\mathbb{R}^d$.

Notation: The symbol $[n]$ stands for the set $\{1, \ldots, n\}$.

(a) Now consider an arbitrary embedding $\psi : \mathbb{R}^d \mapsto \mathbb{R}^k$ which preserves all pairwise distances and norms up-to a multiplicative factor for all points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in the data set, that is,

$$(1-\epsilon)\|\mathbf{x}_i\|^2 \le \|\psi(\mathbf{x}_i)\|^2 \le (1+\epsilon)\|\mathbf{x}_i\|^2 \qquad \text{for all } i \in [n], \quad \text{and} \tag{5}$$

$$(1-\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \le \|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j)\|^2 \le (1+\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \qquad \text{for all } i, j \in [n], \tag{6}$$

where $0 < \epsilon \ll 1$ is a small scalar. Further assume that $\|\mathbf{x}_i\| \le 1$ for all $i \in [n]$. **Show that the embedding ψ satisfying equations (6) and (5) preserves *each pairwise inner product*:**

$$|\psi(\mathbf{x}_i)^\top \psi(\mathbf{x}_j) - (\mathbf{x}_i^\top \mathbf{x}_j)| \le C\epsilon, \qquad \text{for all } i, j \in [n], \tag{7}$$

for some constant C. Thus, we find that if an embedding approximately preserves distances and norms up to a small multiplicative factor, and the points have bounded norms, then inner products are also approximately preserved upto an additive factor.

Hint: Break up the problem into showing that $\psi(\mathbf{x}_i)^\top \psi(\mathbf{x}_j) - (\mathbf{x}_i^\top \mathbf{x}_j) \ge -C\epsilon$, and $\psi(\mathbf{x}_i)^\top \psi(\mathbf{x}_j) - (\mathbf{x}_i^\top \mathbf{x}_j) \le C\epsilon$. The constant $C = 3$ should work, though you can use a larger constant if you need. You may also want to use the Cauchy-Schwarz inequality.

Solution: We will show that

$$-3\epsilon \le (\psi(\mathbf{x}_i)^\top \psi(\mathbf{x}_j) - \mathbf{x}_i^\top \mathbf{x}_j) \le 3\epsilon.$$

Taking absolute value, we obtain the claimed result for $C = 3$.

Lower bound: Expanding the second inequality from equation (6), we obtain

$$\left\|\psi(\mathbf{x}_i)\right\|^2 + \left\|\psi(\mathbf{x}_j)\right\|^2 - 2\psi(\mathbf{x}_i)^\top\psi(\mathbf{x}_j) \leq (1+\epsilon)\|\mathbf{x}_i\|^2 + (1+\epsilon)\left\|\mathbf{x}_j\right\|^2 - 2(1+\epsilon)\mathbf{x}_i^\top\mathbf{x}_j.$$

We use the lower bounds from equation (5):

$$\begin{aligned}\left\|\psi(\mathbf{x}_i)\right\|^2 &\geq (1-\epsilon)\|\mathbf{x}_i\|^2\\ \left\|\psi(\mathbf{x}_j)\right\|^2 &\geq (1-\epsilon)\left\|\mathbf{x}_j\right\|^2.\end{aligned}$$

And hence we have

$$(1-\epsilon)\|\mathbf{x}_i\|^2 + (1-\epsilon)\left\|\mathbf{x}_j\right\|^2 - 2\psi(\mathbf{x}_i)^\top\psi(\mathbf{x}_j) \leq (1+\epsilon)\|\mathbf{x}_i\|^2 + (1+\epsilon)\left\|\mathbf{x}_j\right\|^2 - 2(1+\epsilon)\mathbf{x}_i^\top\mathbf{x}_j.$$

Moving terms around we obtain

$$\begin{aligned}2(\mathbf{x}_i^\top\mathbf{x}_j - \psi(\mathbf{x}_i)^\top\psi(\mathbf{x}_j)) &\leq 2\epsilon(\|\mathbf{x}_i\|^2 + \left\|\mathbf{x}_j\right\|^2 - \mathbf{x}_i^\top\mathbf{x}_j)\\ &\leq 2\epsilon(\|\mathbf{x}_i\|^2 + \left\|\mathbf{x}_j\right\|^2 + \|\mathbf{x}_i\|\left\|\mathbf{x}_j\right\|)\\ &\leq 6\epsilon,\end{aligned}$$

since $\|\mathbf{x}_i\| \leq 1$. Thus we have

$$\begin{aligned}(\mathbf{x}_i^\top\mathbf{x}_j - \psi(\mathbf{x}_i)^\top\psi(\mathbf{x}_j)) &\leq 3\epsilon\\ (\psi(\mathbf{x}_i)^\top\psi(\mathbf{x}_j) - \mathbf{x}_i^\top\mathbf{x}_j) &\geq -3\epsilon\end{aligned}$$

Upper bound: Expanding the first inequality from equation (6), we obtain

$$\left\|\psi(\mathbf{x}_i)\right\|^2 + \left\|\psi(\mathbf{x}_j)\right\|^2 - 2\psi(\mathbf{x}_i)^\top\psi(\mathbf{x}_j) \geq (1-\epsilon)\|\mathbf{x}_i\|^2 + (1-\epsilon)\left\|\mathbf{x}_j\right\|^2 - 2(1-\epsilon)\mathbf{x}_i^\top\mathbf{x}_j.$$

We use the upper bounds from equation (5):

$$\begin{aligned}\left\|\psi(\mathbf{x}_i)\right\|^2 &\leq (1+\epsilon)\|\mathbf{x}_i\|^2\\ \left\|\psi(\mathbf{x}_j)\right\|^2 &\leq (1+\epsilon)\left\|\mathbf{x}_j\right\|^2.\end{aligned}$$

And hence we have

$$(1+\epsilon)\|\mathbf{x}_i\|^2 + (1+\epsilon)\left\|\mathbf{x}_j\right\|^2 - 2\psi(\mathbf{x}_i)^\top\psi(\mathbf{x}_j) \geq (1-\epsilon)\|\mathbf{x}_i\|^2 + (1-\epsilon)\left\|\mathbf{x}_j\right\|^2 - 2(1-\epsilon)\mathbf{x}_i^\top\mathbf{x}_j.$$

Moving terms around we obtain

$$\begin{aligned}2(\mathbf{x}_i^\top\mathbf{x}_j - \psi(\mathbf{x}_i)^\top\psi(\mathbf{x}_j)) &\geq -2\epsilon(\|\mathbf{x}_i\|^2 + \left\|\mathbf{x}_j\right\|^2 - \mathbf{x}_i^\top\mathbf{x}_j)\\ &\geq -2\epsilon(\|\mathbf{x}_i\|^2 + \left\|\mathbf{x}_j\right\|^2 + \|\mathbf{x}_i\|\left\|\mathbf{x}_j\right\|)\\ &\geq -6\epsilon,\end{aligned}$$

since $\|\mathbf{x}_i\| \leq 1$, and so

$$(\psi(\mathbf{x}_i)^\top\psi(\mathbf{x}_j) - \mathbf{x}_i^\top\mathbf{x}_j) \leq 3\epsilon$$

(b) Now we consider the *random feature embedding* using a Gaussian matrix. In next few parts, we work towards proving that if the dimension of embedding is moderately big, then with high probability, the random embedding preserves norms and pairwise distances approximately as described in equations (6) and (5).

Consider the random matrix $\mathbf{J} \in \mathbb{R}^{k\times d}$ with each of its entries being i.i.d. $\mathcal{N}(0,1)$ and consider the map $\psi_{\mathbf{J}} : \mathbb{R}^d \mapsto \mathbb{R}^k$ such that $\psi_{\mathbf{J}}(\mathbf{x}) = \frac{1}{\sqrt{k}}\mathbf{J}\mathbf{x}$. **Show that for any *fixed non-zero vector* u, the random variable $\frac{\|\psi_{\mathbf{J}}(\mathbf{u})\|^2}{\|\mathbf{u}\|^2}$ can be written as**

$$\frac{1}{k}\sum_{i=1}^{k} Z_i^2$$

where Z_i's are i.i.d. $\mathcal{N}(0,1)$ random variables.

Solution: Let $\mathbf{J} = \begin{bmatrix} J_1^\top \\ \vdots \\ J_k^\top \end{bmatrix}$. Then, we have

$$\frac{\|\psi_{\mathbf{J}}(\mathbf{u})\|^2}{\|\mathbf{u}\|^2} = \frac{1}{k}\sum_{i=1}^{k}\frac{(J_i^\top \mathbf{u})^2}{\|\mathbf{u}\|^2}.$$

Note that

$$J_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \mathbf{I}_d).$$

$J_i^T u$ can be simplified as $\sum_{j=1}^{d} J_{id}u_d$ where each $J_{id}u_d$ term represents a Gaussian with mean 0 and variance u_d^2. Since the sum of independent Gaussians is also Gaussian with mean equal to the sum of the means and variance equal to the sum of the variances, we get

$$(J_i^\top \mathbf{u}) \sim \mathcal{N}(0, \|\mathbf{u}\|^2),$$

and consequently, we have

$$Z_i = \frac{J_i^\top \mathbf{u}}{\|\mathbf{u}\|} \overset{i.i.d.}{\sim} \mathcal{N}(0,1).$$

As a result, we conclude that

$$\frac{\|\psi_{\mathbf{J}}(\mathbf{u})\|^2}{\|\mathbf{u}\|^2} = \frac{1}{k}\sum_{i=1}^{k}\frac{(J_i^\top \mathbf{u})^2}{\|\mathbf{u}\|^2}. = \frac{1}{k}\sum_{i=1}^{k} Z_i^2$$

where Z_i's are i.i.d. $\mathcal{N}(0,1)$ random variables.

(c) For any fixed pair of indices $i \neq j$, define the events

$$A_{ij} := \left\{ \frac{\left\|\psi_{\mathbf{J}}(\mathbf{x}_i) - \psi_{\mathbf{J}}(\mathbf{x}_j)\right\|^2}{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2} \in (1-\epsilon, 1+\epsilon) \right\}.$$

which corresponds to the event that the embedding $\psi_{\mathbf{J}}$ approximately preserves the angles between $\mathbf{x}_i$ and $\mathbf{x}_j$. In this part, we show that A_{ij} occurs with high probability.

To do this, you will use the fact that for independent random variables $Z_i \sim \mathcal{N}(0,1)$, we have the following probability bound

$$\mathbb{P}\left[\frac{1}{k}\sum_{i=1}^{k} Z_i^2 \notin (1-t, 1+t)\right] \leq 2e^{-kt^2/8}, \qquad \text{for all } t \in (0,1).$$

Note that this bound suggests that $\sum_{i=1}^{k} Z_i^2 \approx k = \sum_{i=1}^{k} \mathbb{E}[Z_i^2]$ with high probability. In other words, sum of squares of Gaussian random variables concentrates around its mean with high probability. **Using this bound and the previous subproblem, show that**

$$\mathbb{P}\left[A_{ij}^c\right] \leq 2e^{-k\epsilon^2/8},$$

where A_{ij}^c denotes the complement of the event A_{ij}.

Solution:

$$\begin{aligned}\mathbb{P}\left[A_{ij}^c\right] &= \mathbb{P}\left[\frac{\left\|\psi_{\mathbf{J}}(\mathbf{x}_i) - \psi_{\mathbf{J}}(\mathbf{x}_j)\right\|^2}{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2} \notin (1-\epsilon, 1+\epsilon)\right] \\ &\overset{\text{part (b)}}{=} \mathbb{P}\left[\frac{1}{k}\sum_{i=1}^{k} Z_i^2 \notin (1-\epsilon, 1+\epsilon)\right] \\ &\overset{\text{prob. bnd}}{\leq} 2e^{-k\epsilon^2/8}.\end{aligned}$$

(d) Using the previous problem, now **show that if** $k \geq \frac{16}{\epsilon^2}\log\left(\frac{n}{\delta}\right)$**, then**

$$\mathbb{P}\left[\text{for all } i, j \in [n], i \neq j, \frac{\left\|\psi_{\mathbf{J}}(\mathbf{x}_i) - \psi_{\mathbf{J}}(\mathbf{x}_j)\right\|^2}{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2} \in (1-\epsilon, 1+\epsilon)\right] \geq 1 - \delta.$$

That is show that for k large enough, with high probability the random feature embedding $\psi_{\mathbf{J}}$ approximately preserves the pairwise distances. Using this result, we can conclude that random feature embedding serves as a good tool for dimensionality reduction if we project to enough number of dimensions. This result is popularly known as the *Johnson-Lindenstrauss Lemma*.

Hint 1: Let

$$\mathcal{A} := \left\{\text{for all } i, j \in [n], i \neq j, \frac{\left\|\psi_{\mathbf{J}}(\mathbf{x}_i) - \psi_{\mathbf{J}}(\mathbf{x}_j)\right\|^2}{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2} \in (1-\epsilon, 1+\epsilon)\right\}$$

denote the event whose probability we would like to lower bound. Express the complement $\mathcal{A}^c$ in terms of the events A_{ij}^c, and try to apply a union bound to these events.

Solution: Recall the event from the hint:

$$\mathcal{A} := \left\{ \text{for all } i, j \in [n], i \neq j, \frac{\left\|\psi_{\mathbf{J}}(\mathbf{x}_i) - \psi_{\mathbf{J}}(\mathbf{x}_j)\right\|^2}{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2} \in (1-\epsilon, 1+\epsilon) \right\}$$

is equivalent to the intersection of all events A_{ij}, since all events A_{ij} must occur for $\mathcal{A}$ to occur. That is

$$\mathcal{A} = \bigcap_{i\in[n], j\in n, i\neq j} A_{ij}.$$

Note that there are $\binom{n}{2}$ events. Now as given in the hint, we have

$$\begin{aligned}
\mathbb{P}[\mathcal{A}] &= \mathbb{P}\left[\bigcap_{i,j\in[n],i\neq j} A_{ij}\right] \\
&= 1 - \mathbb{P}\left[\left(\bigcap_{i,j\in[n],i\neq j} A_{ij}\right)^c\right] \\
&= 1 - \mathbb{P}\left[\bigcup_{i,j\in[n],i\neq j} A_{ij}^c\right] \\
&\geq 1 - \sum_{i\in[n], j\in[n], i\neq j} \mathbb{P}\left[A_{ij}^c\right].
\end{aligned}$$

By the previous part of the problem,

$$\mathbb{P}\left[A_{ij}^c\right] \leq 2e^{-k\epsilon^2/8}.$$

Thus, we have that

$$\begin{aligned}
\mathbb{P}[\mathcal{A}] &\geq 1 - \sum_{i\in[n], j\in[n], i\neq j} \mathbb{P}\left[A_{ij}^c\right] \\
&\geq 1 - \sum_{i\in[n], j\in[n], i\neq j} 2e^{-k\epsilon^2/8} \\
&\geq 1 - \binom{n}{2} 2e^{-k\epsilon^2/8} \\
&\geq 1 - n(n-1)e^{-k\epsilon^2/8} \\
&\geq 1 - n^2 e^{-k\epsilon^2/8}.
\end{aligned}$$

If $k \geq \frac{16}{\epsilon^2}\log\left(\frac{n}{\delta}\right)$, we have that

$$\begin{aligned}
n^2 e^{-k\epsilon^2/8} &\leq n^2 e^{-2\log\left(\frac{n}{\delta}\right)} \\
&= n^2 * \frac{\delta^2}{n^2}
\end{aligned}$$

$$
\begin{aligned}
&= \delta^2 \\
&\leq \delta
\end{aligned}
$$

since $\delta \in (0, 1)$. Thus, $\mathbb{P}[\mathcal{A}] \geq 1 - n^2 e^{-k\epsilon^2/8} \geq 1 - \delta$.

4 Interpreting Neural Nets Using T-SNE

For this question, please go through the Google Colab Notebook ***here*** to complete the code.

In lecture, you have learned about how t-SNE is a method for nonlinear dimensionality reduction. This is particularly useful for analyzing many real-world datasets in which the data can be categorized according to underlying labels. In this question, you will examine the effect that a neural network has on the t-SNE of such a dataset.

(a) We will work with the CIFAR-10 dataset for this problem, in which the image data is categorized into 10 classes. Flatten the images and take the t-SNE of the training dataset. Plot the t-SNE embeddings and color-code each data point according to its class. Explain what you observe.

Solution: Solution code can be found ***here***.

All the data is grouped together in a bit of a jumble. This is because without any featurization of the data, it is difficult to interpret simply the flattened images. We will see that the featurization will induce meaningful distances and relationships between the data points in the respective high-dimensional spaces.

(b) Now, we have provided a trained neural network for you to analyze. Save it to your Google Drive so that you can access it from the Colab notebook. The model consists of several convolutional layers and a few linear layers. Calculate its accuracy on the test data.

Solution: The accuracy should be above 70%.

(c) Instead of taking the t-SNE of the training dataset directly, we will take the t-SNE of the *features* of the neural network when the training dataset is given as input. Using the "hook" functions provided in the notebook, save the outputs of the third convolutional layer of the network for each input data point. Take the t-SNE of these outputs and color-code each point according to its class. Explain what you observe.

Solution: We begin to see some "clustering" effects of the data according to the class labels, but not entirely.

(d) Do the same as the above except for both the first and second linear layers of the network. Overall, what does it look like the network is doing to the data? How might this change depending on the network's accuracy?

Solution: At the last layers of the network, we can most clearly see how the network is learning representations of the data points that allow them to be separated into classes. This makes sense because the NN is trained to classify the images. We can generally expect more of a "clustering" effect in the t-SNE in the later layers of the network than in the previous, and additionally if the network is more accurate than not.

5 Astronomer's conundrum

As machine learning invades everything in the world, you find that you can use machine learning to classify celestial bodies. Conveniently, you lose all your precious data and only have the rates at which these celestial bodies lose their mass via stellar wind.

There are three types of celestial bodies that you want to classify: dwarfs, giants, and black holes. Dwarfs slowly lose their mass; giants rapidly lose their mass; black holes, on the other hand, gain mass by absorbing stuff (i.e., they lose mass at negative rates).

Before we continue, let's familiarize ourselves with a kind of probability distribution called the *exponential distribution*. The probability density function of an exponential distribution with parameter λ has the following form:

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & x < 0. \end{cases}$$

Note the pdf decreases monotonically on $[0, +\infty)$.

The rate at which a dwarf or a giant *loses* its mass is exponentially distributed with parameters λ_d and λ_g, respectively. The rate at which a black hole *gains* mass is also exponentially distributed, with parameter λ_b.

(a) Suppose that we estimate the rate of *loss* for dwarfs and giants as $\lambda_d = 4$ and $\lambda_g = 3$ respectively. Moreover, we estimate the rate of *gain* for black holes as $\lambda_b = 5$ (i.e. the rate of loss for black holes is -5). We also know that 60% of all the celestial bodies are dwarfs, 30% are giants, and 10% are black holes. Determine the optimal Bayes classifier that assigns a new data point to one of these three classes based on its rate of loss x. Assume we use a 0-1 loss.

Solution: The decision boundary between black holes and the others is $x = 0$. We can see this by examining the posterior for black holes:

$$P(\text{black hole} \mid x) \propto P(x \mid \text{black hole})P(\text{black hole}) = \begin{cases} 0.1 * 5e^{5x} & x \leq 0, \\ 0 & x > 0. \end{cases}$$

For giants and dwarfs, the posterior is 0 when $x \leq 0$. Since a Bayes classifier maximizes posterior probability under the 0-1 loss, when $x \leq 0$, it will be classified as a black hole. When $x \geq 0$, the posteriors for the giant and dwarf will be higher.

The boundary between dwarfs and giants can be calculated by solving for the location at which the posterior probabilities are the same; that is

$$P(\text{giant} \mid x) = P(\text{dwarf} \mid x)$$
$$P(x \mid \text{giant})P(\text{giant}) = P(x \mid \text{dwarf})P(\text{dwarf})$$
$$3e^{-3x} \cdot 0.3 = 4e^{-4x} \cdot 0.6$$

which yields $x = \ln(8/3)$.

Therefore, our decision rule is: if $x < 0$, classify as black holes; if $0 \leq x < \ln(8/3)$, classify as dwarfs; if $x \geq \ln(8/3)$, classify as giants.

(b) Following the previous question, find the risk of your Bayes classifier. Feel free to use WolframAlpha or some other software for the integration.

Solution: The formula for risk is

$$\int_{-\infty}^{\infty} \left[\sum_{j=1}^{K} L(f(X), j) P(X \mid Y = j) P(Y = j) \right] \mathrm{d}x$$

where $f(X)$ is the decision rule and $K = 3$ classes. So, for a 0-1 loss, we have

$$\begin{aligned} &\int_X \left[\mathbf{1}\{0 \leq x < \ln(8/3)\} P(x \mid \text{giant}) P(\text{giant}) + \mathbf{1}\{x \geq \ln(8/3)\} P(x \mid \text{dwarf}) P(\text{dwarf}) \right] \mathrm{d}x \\ &= \int_0^{\ln(8/3)} 0.3 \cdot 3e^{-3x} \, \mathrm{d}x + \int_{\ln(8/3)}^{+\infty} 0.6 \cdot 4e^{-4x} \, \mathrm{d}x \\ &\approx 0.2841 + 0.0119 \\ &= 0.296 \end{aligned}$$

There is no error incurred for the black holes since the decision rule always correctly classifies them.

6 Risk Minimization with Doubt

Suppose we have a classification problem with classes labeled $1, \dots, c$ and an additional "doubt" category labeled $c+1$. Let $f : \mathbb{R}^d \to \{1, \dots, c+1\}$ be a decision rule. Define the loss function

$$L(f(\mathbf{x}), y) = \begin{cases} 0 & \text{if } f(\mathbf{x}) = y \quad f(\mathbf{x}) \in \{1, \dots, c\}, \\ \lambda_c & \text{if } f(\mathbf{x}) \neq y \quad f(\mathbf{x}) \in \{1, \dots, c\}, \\ \lambda_d & \text{if } f(\mathbf{x}) = c+1 \end{cases} \tag{8}$$

where $\lambda_c \geq 0$ is the loss incurred for making a misclassification and $\lambda_d \geq 0$ is the loss incurred for choosing doubt. In words this means the following:

- When you are correct, you should incur no loss.
- When you are incorrect, you should incur some penalty λ_c for making the wrong choice.
- When you are unsure about what to choose, you might want to select a category corresponding to "doubt" and you should incur a penalty λ_d.

In lecture, you saw a definition of risk over the expectation of data points. We can also define the risk of classifying a new individual data point $\mathbf{x}$ as class $f(\mathbf{x}) \in \{1, 2, \dots, c+1\}$:

$$R(f(\mathbf{x}) \mid \mathbf{x}) = \sum_{i=1}^{c} L(f(\mathbf{x}), i)\, P(Y = i \mid \mathbf{x}).$$

(a) First, we will simplify the risk function using our specific loss function separately for when $f(\mathbf{x})$ is or is not the doubt category.

i. Prove that $R(f(\mathbf{x}) = i \mid \mathbf{x}) = \lambda_c\,(1 - P(Y = i \mid \mathbf{x}))$ when i is not the doubt category (i.e. $i \neq c+1$).

Solution:

$$R(f(\mathbf{x}) = i|\mathbf{x}) = \sum_{j=1}^{c} L(f(\mathbf{x}) = i, y = j) P(Y = j|\mathbf{x}) \tag{9}$$

$$= 0 \cdot P(Y = i|\mathbf{x}) + \lambda_c \sum_{j=1, j\neq i} P(Y = j|\mathbf{x}) \tag{10}$$

$$= \lambda_c\,(1 - P(Y = i|\mathbf{x})) \tag{11}$$

ii. Prove that $R(f(\mathbf{x}) = c+1 \mid \mathbf{x}) = \lambda_d$.

Solution:

$$R(f(\mathbf{x}) = c+1 \mid \mathbf{x}) = \sum_{j=1}^{c} L(f(\mathbf{x}) = c+1, y = j) P(Y = j \mid \mathbf{x}) \tag{12}$$

$$= \lambda_d \sum_{j=1} P(Y = j \mid \mathbf{x}) \tag{13}$$

$$= \lambda_d \tag{14}$$

because $\sum_{j=1} P(Y = j|\mathbf{x})$ should sum to 1 since its a proper probability distribution.

(b) Show that the following policy $f_{opt}(x)$ obtains the minimum risk:

- (**R1**) Find the non-doubt class i such that $P(Y = i \mid \mathbf{x}) \geq P(Y = j \mid \mathbf{x})$ for all j, meaning you pick the class with the highest probability given x.
- (**R2**) Choose class i if $P(Y = i \mid \mathbf{x}) \geq 1 - \frac{\lambda_d}{\lambda_c}$
- (**R3**) Choose doubt otherwise.

Hint: In order to prove that $f_{opt}(x)$ minimizes risk, consider proof techniques that show that $f_{opt}(x)$ "stays ahead" of all other policies that *don't* follow these rules. For example, you could take a proof-by-contradiction approach: assume there exists some other policy, say $f'(x)$, that minimizes risk more than $f_{opt}(x)$. What are the scenarios where the predictions made by $f_{opt}(x)$ and $f'(x)$ might differ? In these scenarios, and based on the rules above that $f_{opt}(x)$ follows, why would $f'(x)$ not be able to beat $f_{opt}(x)$ in risk minimization?

Solution: Let $f_{opt} : \mathbb{R}^d \to \{1, \ldots, c+1\}$ be the decision rule which implements (**R1**)–(**R3**). We want to show that in expectation the rule f_{opt} is at least as good as an arbitrary rule f. Let $\mathbf{x} \in \mathbb{R}^d$ be a data point, which we want to classify. Let's examine all the possible scenarios where $f_{opt}(\mathbf{x})$ and another arbitrary rule $f(\mathbf{x})$ might differ:

1. Let $f_{opt}(\mathbf{x}) = i$ where $i \neq c + 1$.
   - Case 1a: $f(\mathbf{x}) = k$ where $k \neq i$. Then we get with (**R1**) that

$$\begin{aligned} R(f_{opt}(\mathbf{x}) = i \mid \mathbf{x}) &= \lambda_c \left(1 - P(Y = i \mid \mathbf{x})\right) \\ &\leq \lambda_c \left(1 - P(Y = k \mid \mathbf{x})\right) = R(f(\mathbf{x}) = k \mid \mathbf{x}). \end{aligned}$$

   - Case 1b: $f(\mathbf{x}) = c + 1$. Then we get with (**R2**) that

$$\begin{aligned} R(f_{opt}(\mathbf{x}) = i \mid \mathbf{x}) &= \lambda_c \left(1 - P(Y = i \mid \mathbf{x})\right) \\ &\leq \lambda_c \left(1 - \left(1 - \frac{\lambda_d}{\lambda_c}\right)\right) = \lambda_d = R(f(\mathbf{x}) = c + 1 \mid \mathbf{x}). \end{aligned}$$

2. Let $f_{opt}(\mathbf{x}) = c + 1$ and $f(\mathbf{x}) = k$ where $k \neq c + 1$. Then

$$R(f(\mathbf{x}) = k \mid \mathbf{x}) = \lambda_c(1 - P(Y = k \mid \mathbf{x}))$$
$$R(f_{opt}(\mathbf{x}) = c + 1 \mid \mathbf{x}) = \lambda_d$$

   We are in case (**R3**) which means that

$$\max_{j \in \{1,\ldots,c\}} P(Y = j \mid \mathbf{x}) < 1 - \frac{\lambda_d}{\lambda_c}.$$

Hence, $P(Y = k \mid \mathbf{x}) < 1 - \lambda_d/\lambda_c$, which means

$$\begin{aligned} R(f(\mathbf{x}) = k \mid \mathbf{x}) &> \lambda_c\left(1 - \left(1 - \frac{\lambda_d}{\lambda_c}\right)\right) \\ &= \lambda_d = R(f_{opt}(\mathbf{x})) \end{aligned}$$

In every case we proved that the rule f_{opt} is at least as good as the arbitrary rule f, which proves that f_{opt} is an optimal rule.

(c) How would you modify your optimum decision rule if $\lambda_d = 0$? What happens if $\lambda_d > \lambda_c$? Explain why this is or is not consistent with what one would expect intuitively.

Solution: If $\lambda_d = 0$, then the rule explained in R1 will hold iff there exists an $i \in \{1, \ldots, c\}$ such that $P(f_{opt}(\mathbf{x}) = i \mid \mathbf{x}) = 1$, since this is the only circumstance in which we satisfy R2. So we will either classify x in class i if we are 100% sure about this, or else we will choose doubt. Of course this is completely consistent with our intuition, because choosing doubt does not have any penalty at all, since $\lambda_d = 0$.

If $\lambda_d > \lambda_c$, then we will always classify x in the class $i \in \{1, \ldots, c\}$ which gives the highest probability of correct classification. Once again this makes sense, since the cost of choosing doubt is higher than classifying $\mathbf{x}$ in any of the classes, hence our best option is to classify x in the class which gives the highest probability for a correct classification.

7 Honor Code

1. **List all collaborators. If you worked alone, then you must explicitly state so.**

2. **Declare and sign the following statement**:

   "I certify that all solutions in this document are entirely my own and that I have not looked at anyone else's solution. I have given credit to all external sources I consulted."

   Signature : ______________________________

   While discussions are encouraged, *everything* in your solution must be your (and only your) creation. Furthermore, all external material (i.e., *anything* outside lectures and assigned readings, including figures and pictures) should be cited properly. We wish to remind you that the consequences of academic misconduct are *particularly severe*!