Unsupervised Learning HW0

Due: Wed Sept 14, 2022 at 11:59pm

Welcome to Unsupervised Learning. The goal of this assignment is for you to recall core math and ML concepts to prepare for the course, and get familiarized with the homework submission system (Gradescope). Everyone enrolled or on the waitlist intending to enroll must submit this assignment on Gradescope by the due date.

Even though the purpose of this homework is to recall core concepts, the score received on this assignment will count towards your final grade in this course. You must show your work to receive full credit. You must cite all resources (including online material, books, articles, help taken from specific individuals, etc.) you used to complete your work.

This homework assignment is to be done individually. All homeworks (including this one) should be typesetted properly in pdf format. Handwritten solutions will not be accepted. You must include your name and UNI in your homework submission.

1 Properties about norms

A norm on $\mathbb{R}^d$ is a function $\|\cdot\| : \mathbb{R}^d \to \mathbb{R}$ which satisfies the following properties:

- Positivity: for any $x \in \mathbb{R}^d$, $\|x\| \geq 0$, with equality iff $x = 0$.
- Homogeneity: for any $x \in \mathbb{R}^d$ and $t \in \mathbb{R}$, $\|tx\| = |t| \cdot \|x\|$.
- Triangle inequality: for any $x, y \in \mathbb{R}^d$, $\|x + y\| \leq \|x\| + \|y\|$.

A useful family of norms are the l_p norms, defined as follows for $p \geq 1$:

$$\|x\|_p = \Big(\sum_{i=1}^{d} |x_i|^p\Big)^{1/p}.$$

These include the familiar l_1, l_2, and l_∞ norms. You may assume all of these satisfy the definition of norm.

(i) Show that for any $x \in \mathbb{R}^d$,

$$\|x\|_2 \leq \|x\|_1 \leq \|x\|_2 \cdot \sqrt{d}$$

and give example when each of these inequalities is tight.

(ii) Show that for any $x \in \mathbb{R}^d$ and any $p \geq 1$, $\|x\|_1 \geq \|x\|_p$.

(iii) Show that for any $x \in \mathbb{R}^d$, and any $1 \leq p \leq q$, $\|x\|_p \geq \|x\|_q$. (ie, the l_p norm of a vector is always larger than its l_q norm, for $p \leq q$.)

(iv) Show that for any $x \in \mathbb{R}^d$ and $1 \leq p \leq q$,

$$\|x\|_p \leq \|x\|_q \cdot d^{(1/p)-(1/q)}.$$

(you can use Holder's inequality, with states that $|x \cdot y| \leq \|x\|_a \|y\|_b$ for any vectors x, y and any $a, b \geq 1$ with $(1/a) + (1/b) = 1$.)

2 A simple property of the gradient

If $f : \mathbb{R}^d \to \mathbb{R}$ is a differentiable function, then one often hears that the gradient

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_d} \right)$$

points in the direction of steepest ascent of f provided that $\nabla f \neq 0$. In other words, this says for any unit vector u one has the bound

$$\frac{d}{dt} f(x + tu)\Big|_{t=0} \leq \frac{d}{dt} f(x + tv)\Big|_{t=0}$$

where $v = \nabla f(x)/\|\nabla f(x)\|$. Prove this inequality and show that it is strict unless $u = v$.

3 Basics of Dimension Reduction and Clustering

From any introductory Machine Learning course, you should be aware of the techniques: PCA and k-means. In this problem, we explore how these techniques can be improved via kernel methods.

(i) How can we benefit by running PCA and k-means clustering in a transformed feature space?

3.1 Kernelized PCA

Given a (mean centered) data $D \times n$ data matrix X, a typical way of doing PCA is to analyze the eigenvectors/values of the $D \times D$ covariance matrix XX^T. If $D \gg n$, then computing the covariance (and thus PCA is computationally prohibitive). A natural question to wonder is what if instead we compute the eigenvectors/values of the much smaller inner product $n \times n$ matrix $X^\mathsf{T}X$?

(ii) Let (λ_i, v_i) be the eigenvalue/vector pairs of XX^T, and (μ_i, u_i) be the eigenvalue/vector pairs of $X^\mathsf{T}X$. Show that one can write (λ_i, v_i) purely in terms of (μ_i, u_i) and possibly the original data, thus significantly improving the computational effort!

(iii) Say we want to project the given data matrix X into the $k < \min\{D, n\}$ dimensional PCA subspace. Using only (μ_i, u_i) and possibly the original data (cf. Part (i)), derive an expression for the k-dimensional PCA projection of the data matrix X.

(iv) Another advantage of computing PCA using the inner product matrix $X^\mathsf{T}X$ is that we can kernelize all calculations for finding the k-dimensional PCA projection of both the input data matrix as well as of new datapoints. Let $K(x_i, x_j)$ be a kernel function that efficiently computes the inner product $\langle \phi(x_i), \phi(x_j) \rangle$ in a (possibly nonlinear) feature space ϕ. Derive kernelized expressions for k-dimensional PCA projection in the feature space ϕ of a datapoint x_i from the input data matrix and for a new datapoint x.

3.2 Kernelized k-Means

(v) For fixed vectors $x_1, ..., x_n$ show that $f(c) = \sum_{i=1}^{n} \|x_i - c\|_2^2$ is minimized at $c = \frac{1}{n}\sum_{i=1}^{n} x_i$. This tells us that, for the k-means problem, it suffices to find a k-way partition of the data because the optimal centers are implicitly the means of each cluster.

(vi) For c computed above and an arbitrary x_0, show that $\|x_0 - c\|_2^2$ can be computed if we know the dot products between the points $x_0, x_1, ..., x_n$.

(vii) Propose an algorithm that performs Lloyd's k-means algorithm in the feature space that has a kernel function $K(.,.)$.

3.3 Advantages of Kernelization

(viii) Discuss the possible advantages of the kernelized versions of these methods in contrast to the non-kernelized version?

4 Deviations of the Chi-Squared Distribution

The χ^2-distribution with n degrees of freedom is the distribution of the random variable Y such that $Y = X_1^2 + ... + X_n^2$, where X_i are i.i.d. variables drawn from the standard normal distribution $N(0,1)$. Our goal is to study the *deviations* of the χ^2 distribution.

(i) For a fixed value of $t > 0$, show that $\mathbb{E}[e^{tY}] = (1-2t)^{-n/2}$.

(ii) Prove that for all $\epsilon \in (0,1)$ we have $\Pr[Y > (1+\epsilon)^2 n] < e^{-cn\epsilon^2}$ for some constant c independent of n and ϵ.

This type of a question ($\Pr[Y > a]$) is the study of deviations. That is, what is the chance that a realization of a random variable Y exceeds a particular value a. This is very useful for analyzing statistical models and random processes.

Hint: use the Chernoff bounding technique

- (for any $t > 0$): $\Pr[Y > a] = \Pr[e^{tY} > e^{ta}] \leq \dfrac{\mathbb{E}[e^{tY}]}{e^{ta}}$.
- Use calculus to find the optimal setting of t.

You can check yourself that $\Pr[Y < (1-\epsilon)^2 n] < e^{-cn\epsilon^2}$ following a similar proof.

5 The Graph Laplacian

Consider an undirected graph $G(V,E)$. Let f be a function from V to $\mathbb{R}$, or equivalently, a vector in the vector space $\mathbb{R}^V$. Let D be the **degree matrix** of G, i.e. $D_{vv} = \text{degree}(v)$ and $D_{uv} = 0$ for $u \neq v$. Let A be the **adjacency matrix** of G, i.e. $A_{uv} = 1$ if edge $uv \in E$ and 0 otherwise. The matrix $L = D - A$ is called the **Laplacian** of G.

(i) Prove that $f^T L f = \sum_{uv \in E} (f(u) - f(v))^2$.

(ii) Prove that if G has k connected components then L has exactly $n-k$ eigenvectors corresponding to non-zero eigenvalues.

6 ML review

State true or false. You must justify your answer with detailed explanations/derivations.

(i) A finite hypothesis class $\mathcal{H}$ can have a VC-dimension at most $\log_2 |\mathcal{H}|$.

(ii) Lasso regression is consistent. That is, as the number of training samples approach infinity, L_2 error of Lasso regression approaches that of the optimal L_2 regressor.

(iii) Consider a multi-class classifier $f : \mathbb{R}^d \to \{1, 2, \ldots, c\}$, that maps data from $\mathbb{R}^d$ to one of c different categories ($c \geq 2$). The classifier f is optimal (with respect to the 0-1 loss) if f is defined as

$$f(\vec{x}) := \operatorname*{arg\,max}_{y \in \{1,\ldots,c\}} \Big(\Pr[Y = y] \cdot \prod_{i=1}^{d} \Pr\left[X_i = \vec{x}_i \mid Y = y, X_1 = \vec{x}_1, \ldots, X_{i-1} = \vec{x}_{i-1}\right] \Big).$$

(iv) Let $X \in \mathbb{R}^d$ be a random vector with zero mean and finite second moments. For any direction indicated by a unit length vector $v \in \mathbb{R}^d$), the variance of X in the direction v is given by the expression

$$v^{\mathsf{T}} \, \mathbb{E}(XX^{\mathsf{T}})v$$

(v) Strictly convex functions $f : \mathbb{R}^d \to \mathbb{R}$ are bounded from below.