Andrew Li-Yang Liu (ALL2209)
Nakul Verma
Unsupervised Learning
Columbia University
September 14, 2022

UML Homework 0

1. Properties about norms

1i:

Let $x \in \mathbb{R}^d$ be $x = (x_1, \dots, x_d)$. Then:

$$\left|\left|x\right|\right|_1^2 = (|x_1| + \cdots + |x_d|)^2 = x_1^2 + \cdots + x_d^2 + 2\sum_{i \neq j} |x_i||x_j|$$

We note that the sum over cross terms $2\sum_{i \neq j} |x_i||x_j|$ is non-negative, so we have:

$$\left|\left|x\right|\right|_1^2 \geq x_1^2 + \cdots + x_d^2 = \left|\left|x\right|\right|_2^2$$

Now let $a = (1,1,1 \dots,1)$ and $b = (|x_1|, \dots |x_d|)$. Note that $a^T b = |x_1| + \cdots + |x_d| = \left|\left|x\right|\right|_1$

We note that by Cauchy-Schwarz inequality:

$$|a^T b| \leq \left|\left|a\right|\right|_2 \left|\left|b\right|\right|_2$$

Hence:

$$\left|\left|x\right|\right|_1 \leq \sqrt{d}\sqrt{x_1^2 + \cdots + x_d^2} = \sqrt{d}\left|\left|x\right|\right|_2$$

Hence, we've shown that:

$$\left|\left|x\right|\right|_2 \leq \left|\left|x\right|\right|_1 \leq \sqrt{d}\left|\left|x\right|\right|_2$$

For an example when the inequality is tight, just use the case where $x = (0,0,0, \dots,0)$.

1ii:

This result arises trivially from 1iii, so simply see the proof for 1iii.

In particular, once we prove 1iii, simply take p from part iii to be equal to 1, and q from part iii to be p from part ii. This proves part ii.

1iii:

Consider the quantity:

$$\frac{d}{dp}\ln\left(||x||_p\right) = \frac{d}{dp}\ln\left((|x_1|^p + \cdots + |x_d|^p)^{\frac{1}{p}}\right) = \frac{d}{dp}\frac{1}{p}\ln(|x_1|^p + \cdots + |x_d|^p)$$
$$= -\frac{1}{p^2}\ln(|x_1|^p + \cdots + |x_d|^p) + \frac{1}{p}\left(\frac{|x_1|^p \ln(|x_1|) + \cdots + |x_d|^p \ln(|x_d|)}{|x_1|^p + \cdots + |x_d|^p}\right)$$
$$= \frac{1}{p}\left(\frac{\sum_{i=1}^d |x_i|^p \ln(|x_i|)}{||x||_p^{\,p}} - \ln\left(||x||_p\right)\right)$$

But $\frac{d}{dp}\ln\left(||x||_p\right) = \frac{1}{||x||_p}\left(\frac{d}{dp}||x||_p\right)$

So:

$$\frac{d}{dp}||x||_p = \frac{||x||_p}{p}\left(\frac{\sum_{i=1}^d |x_i|^p \ln(|x_i|)}{||x||_p^{\,p}} - \ln\left(||x||_p\right)\right)$$

We want to prove this is negative. We note that, for any 1<=i<=d:

$$\left(\sum_{k=1}^d |x_k|^p\right)^{\frac{1}{p}} \geq |x_i|$$

So:

$$\frac{1}{p}\ln\left(\sum_{k=1}^d |x_k|^p\right) \geq \ln(|x_i|)$$
$$\frac{1}{p}\ln\left(\sum_{k=1}^d |x_k|^p\right)\sum_{i=1}^d |x_i|^p \geq \sum_{i=1}^d |x_i|^p \ln(|x_i|)$$
$$\frac{1}{p}\ln\left(||x||_p^p\right) \geq \frac{\sum_{i=1}^d |x_i|^p \ln(|x_i|)}{||x||_p^p}$$
$$\ln\left(||x||_p\right) \geq \frac{\sum_{i=1}^d |x_i|^p \ln(|x_i|)}{||x||_p^p}$$

Hence, we must have:

$$\frac{d}{dp}||x||_p \leq 0$$

So we see that $||x||_p$ decreases with p. Hence, given any $x \in \mathbb{R}^d$ and any $1 \leq p \leq q$, we have $||x||_p \geq ||x||_q$.

iv.
We note that:

$$\left|\left|x\right|\right|_p^p = \sum_{i=1}^{d} |x_i|^p = \sum_{i=1}^{d} |x_i|^p \cdot 1$$

Applying Holder's inequality, we get:

$$\left|\left|x\right|\right|_p^p \leq \left(\sum_{i=1}^{d} (|x_i|^p)^{\frac{q}{p}}\right)^{\frac{p}{q}} \left(\sum_{i=1}^{d} 1^{\frac{q}{q-p}}\right)^{1-\frac{p}{q}}$$

Where we note that we used $r = \frac{p}{q} \geq 1$. Hence:

$$\left|\left|x\right|\right|_p^p \leq \left(\sum_{i=1}^{d} |x_i|^q\right)^{\frac{p}{q}} d^{1-\frac{p}{q}}$$

$$\left|\left|x\right|\right|_p \leq \left(\sum_{i=1}^{d} |x_i|^q\right)^{\frac{1}{q}} d^{\frac{1}{p}-\frac{1}{q}} = \left|\left|x\right|\right|_q d^{\frac{1}{p}-\frac{1}{q}}$$

As expected.

2. A Simple Property of the Gradient:

$$\frac{d}{dt} f(x+tu)\Big|_{t=0} = \nabla f(x+tu) \cdot u|_{t=0} = \nabla f(x) \cdot u = D_u f(x)$$

Where D_u denotes the directional derivative in the u direction. To maximize this quantity, we note that:

$$\nabla f(x) \cdot u = \left|\left|\nabla f(x)\right|\right| \left|\left|u\right|\right| \cos\theta = \left|\left|\nabla f(x)\right|\right| \cos\theta$$

Where θ is the angle between $\nabla f(x)$ and u. This quantity is maximized when we have $\theta = 0$, i.e. when $u = \frac{\nabla f(x)}{||\nabla f(x)||}$.

Hence, if we take $v = \frac{\nabla f(x)}{||\nabla f(x)||}$, then:

$$D_u f(x) \leq D_v f(x)$$

Therefore, by the first equation:

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

Once again, we get equality when the left-hand side is maximized, i.e. when $\theta = 0$, or $u = \frac{\nabla f(x)}{||\nabla f(x)||} = v$, as desired.

3. Basics of Dimension Reduction and Clustering

3i:

In the original space, it is often impossible to linearly separate the data. Transforming the data into a latent feature space may make it easier to linearly separate them, perform PCA, and k-means clustering.

3.1 Kernelized PCA
ii:
We note that:

$$XX^T v_i = \lambda_i v_i$$

And:

$$X^T X u_i = \mu_i u_i$$

Hence:

$$X(X^T X u_i) = X(\mu_i u_i)$$
$$XX^T(Xu_i) = \mu_i(Xu_i)$$

So we can have:

$$v_i = Xu_i$$
$$\lambda_i = \mu_i$$

iii:
We have from before $v_i = Xu_i$. So choose the top k eigenvectors $v_1, \dots, v_k$. We thus have:

$$V = (v_1, \dots, v_k) = XU$$

Where $U = (u_1, \dots, u_k)$. Then, the PCA projection $\tilde{X}$ would be:

$$\tilde{X} = V^T X = U^T X^T X = (X^T XU)^T = (\mu U)^T$$

Where $\mu = diag(\mu_1, \dots, \mu_k)$.

iv:
To calculate PCA in feature space ϕ, i.e. the PCA projection of $\phi(x)$, we perform the projection:

$$V^T \phi(x) = U^T \phi(X)^T \phi(x)$$

Where we note that $\phi(X) = (\phi(x_1) \dots \phi(x_n))$.

Hence, we can kernelize this computation by using the kernel K:

$$V^T\phi(x) = U^T\begin{pmatrix} K(x_1, x) \\ \vdots \\ K(x_n, x) \end{pmatrix}$$

3.2 Kernelized k-means:
v:
For fixed x1,…,xn, consider:

$$f(c) = \sum_{i=1}^{n} \left|\left|x_i - c\right|\right|_2^2$$

Minimize with respect to c by taking derivative with respect to c and setting equal zero:

$$\partial_c f = \sum_{i=1}^{n} -2(x_i - c) = 0$$

Hence:

$$\sum_{i=1}^{n} x_i = nc$$

$$c = \frac{1}{n}\sum_{i=1}^{n} x_i$$

As desired.

vi:
For c computed above and an arbitrary x0, we have:

$$\begin{aligned} \left|\left|x_0 - c\right|\right|_2^2 &= \left|\left|x_0\right|\right|_2^2 + \left|\left|c\right|\right|_2^2 - 2\langle x_0, c\rangle \\ &= \langle x_0, x_0\rangle + \frac{1}{n^2}\sum_{i,j=1}^{n}\langle x_i, x_j\rangle - \frac{2}{n}\sum_{i=1}^{n}\langle x_0, x_i\rangle \end{aligned}$$

Hence, we see that we can compute this quantity if we know the dot products between x0, x1,…,xn, as desired.

vii: Kernelized Lloyd's k-means algorithm
I split this answer into two parts, one "naïve" algorithm that is a straightforward extension of Lloyd's algorithm and a modification that allows it to be implemented in practice.

Naïve implementation

Randomly initialize clusters $c_1, \ldots, c_k$.

Loop until clusters converge:

For each i=1,…,n:

For each j=1,…,k:

Compute $d_{ij} = \left|\left|\phi(x_i) - c_j\right|\right|_2^2 = K(x_i, x_i) + \frac{1}{n^2}\sum_{l,m=1}^{n} K(x_l, x_m) - \frac{2}{n}\sum_{l=1}^{n} K(x_i, x_l)$

Assign data i to cluster j (x_i, C_j) with smallest distance d_{ij}.

Compute new clusters $c_j = \frac{1}{n_j}\sum_{l \in C_j} \phi(x_l)$ for j=1,…,k.

Obviously, since the feature space may be infinite dimensional, computing the cluster centers may be prohibitively expensive. Hence, in practice we can use a modification of this algorithm:

Practical implementation

Randomly initialize cluster assignments $(x_1, C_1), \dots, (x_n, C_n)$.

Loop until cluster assignments converge:

For each i=1,…,n:

For each j=1,…,k:

Compute $d_{ij} = \left|\left|\phi(x_i) - c_j\right|\right|_2^2 = K(x_i, x_i) + \frac{1}{n^2}\sum_{l,m=1}^{n} K(x_l, x_m) - \frac{2}{n}\sum_{l=1}^{n} K(x_i, x_l)$

Assign data i to cluster j (x_i, C_j) with smallest distance d_{ij}.

3.3 Advantages of Kernelization

viii: Kernelization allows us to compute distances in the transformed space as dot products of vectors in the original space. Since the transformed space generally has much larger dimension than the original space (the feature space is potentially infinite-dimensional), kernelization thus substantially reduces computation time.

4. Deviations of the Chi-squared distribution

4i:

$$E[e^{tY}] = E\left[e^{t(X_1^2 + \cdots + X_n^2)}\right] = E\left[e^{tX_1^2}\right] \dots E\left[e^{tX_n^2}\right] = E\left[e^{tX_1^2}\right]^n$$

The last two steps were due to the fact that the X's are iid.

Let's compute the expectation $E\left[e^{tX_1^2}\right]$:

$$E\left[e^{tX_1^2}\right] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{tx_1^2} e^{-\frac{1}{2}x_1^2} dx_1 = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\left(\frac{1}{2} - t\right)x_1^2} dx_1 = \frac{1}{\sqrt{2\pi}} \sqrt{\frac{\pi}{\frac{1}{2} - t}} = (1 - 2t)^{-\frac{1}{2}}$$

Where in the second last step we used for formula for integral of a Gaussian.

Hence:

$$E[e^{tY}] = E\left[e^{tX_1^2}\right]^n = (1-2t)^{-\frac{n}{2}}$$

As desired.

4ii:
Let a be:

$$a = (1+\epsilon)^2 n$$

By Chernoff bounding technique, we have:

$$P[Y > a] = P[e^{tY} > e^{ta}] \leq \frac{E[e^{tY}]}{e^{ta}} = (1-2t)^{-\frac{n}{2}} e^{-ta}$$

Next, minimize this upper bound w.r.t t by taking derivative w.r.t t:

$$\frac{d}{dt}\left[(1-2t)^{-\frac{n}{2}} e^{-ta}\right] = n(1-2t)^{-\frac{n}{2}-1} e^{-ta} - a(1-2t)^{-\frac{n}{2}} e^{-ta} = 0$$

$$\left(\frac{n}{1-2t} - a\right) = 0$$
$$t = \frac{1}{2}\left(1 - \frac{n}{a}\right)$$

Plugging this value of t into the upper bound:

$$P[Y > a] \leq (1-2t)^{-\frac{n}{2}} e^{-ta} = \left(\frac{n}{a}\right)^{-\frac{n}{2}} e^{-\frac{1}{2}(a-n)} = (1+\epsilon)^n e^{-\frac{1}{2}(\epsilon^2+2\epsilon)n} = (1+\epsilon)^n e^{-n\epsilon} e^{-\frac{1}{2}n\epsilon^2}$$

We note that for $n > 0$, we have $(1+\epsilon)^n e^{-n\epsilon} < 1$. This is because $e^{\epsilon} = 1 + \epsilon + \frac{\epsilon^2}{2} + \cdots > 1 + \epsilon$, so $e^{n\epsilon} > (1+\epsilon)^n$, so $(1+\epsilon)^n e^{-n\epsilon} < 1$. Hence:

$$P[Y > a] \leq (1+\epsilon)^n e^{-n\epsilon} e^{-\frac{1}{2}n\epsilon^2} \leq e^{-\frac{1}{2}n\epsilon^2}$$

As desired ($c = \frac{1}{2}$).

5. The Graph Laplacian

i: Say there are n nodes.

$$f^T L f = \sum_{i,j=1}^{n} f_i L_{ij} f_j = \sum_{i,j=1}^{n} f_i (D_{ij} - A_{ij}) f_j$$
$$= \sum_{i,j=1}^{n} f_i D_{ij} f_j - \sum_{i,j=1}^{n} f_i A_{ij} f_j$$
$$= \sum_{i=1}^{n} f_i^2 degree(i) - \sum_{i,j=1}^{n} f_i A_{ij} f_j$$

Note that $degree(i) = \sum_{j=1}^{n} A_{ij}$, so:

$$f^T L f = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} A_{ij} \right) f_i^2 - \sum_{i,j=1}^{n} f_i A_{ij} f_j$$
$$= \frac{1}{2} \left\{ \sum_{i=1}^{n} \left(\sum_{j=1}^{n} A_{ij} \right) f_i^2 - 2 \sum_{i,j=1}^{n} f_i A_{ij} f_j + \sum_{j=1}^{n} \left(\sum_{i=1}^{n} A_{ij} \right) f_j^2 \right\}$$

Where, in the second step, we use the fact that $\sum_{i=1}^{n} \left(\sum_{j=1}^{n} A_{ij} \right) f_i^2 = \sum_{j=1}^{n} \left(\sum_{i=1}^{n} A_{ij} \right) f_j^2$ due to the fact that the adjacency matrix is symmetric. Hence:

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} A_{ij} \left(f_i - f_j \right)^2$$
$$= \sum_{ij \in E} \left(f_i - f_j \right)^2$$

As desired. Note that we got rid of the factor of ½ because the double sum double counts each edge.

ii:

Since we have k connected components, we can partition the graph into k disjoint subgraphs, which we call $G_1, \dots, G_k$. Now, consider the vectors:

$$v_i(j) = \begin{cases} \dfrac{1}{\sqrt{|G_i|}} & , \text{if } j \in G_i \\ 0, & \text{otherwise} \end{cases}$$

Then, since the graphs are disjoint, we have that for $i \neq j$, the vectors v_i and v_j are nonzero on different entries, and thus are orthogonal: $\langle v_i, v_j \rangle = 0$. Moreover, the norm of each vector is:

$$||v_i|| = \sqrt{\sum_{j=1}^{n} \left(v_i(j) \right)^2} = \sqrt{\sum_{j \in G_i} \frac{1}{|G_i|}} = \sqrt{\frac{|G_i|}{|G_i|}} = 1$$

So the vectors are orthonormal. Next, we show that they are eigenvectors of L, with corresponding eigenvalue 0:

$$(Lv)_i = \sum_j L_{ij} v_j = \sum_j D_{ij} v_j - \sum_j A_{ij} v_j = \deg(i)\, v_i - \sum_j A_{ij} v_j = \sum_j A_{ij} v_i - \sum_j A_{ij} v_j$$
$$= \sum_j A_{ij}(v_i - v_j) = \sum_{j \in N(i)} (v_i - v_j)$$

Where $N(i)$ denotes the neighborhood of node i. Since vectors of nodes in the same neighborhood are necessarily vectors of nodes in the same disjoint subgraph, which means said vectors are equal, we thus have:

$$Lv_i = 0$$

For i=1,…,k.

(Apologies for the abuse of indices notation here)

In other words, the v_i's are a set of orthonormal eigenvectors of L with corresponding eigenvalue 0. This means that L has at least k eigenvectors with eigenvalue 0.

Now we want to demonstrate that L has at most k eigenvectors with eigenvalue 0. Suppose there was an additional eigenvector v^* that has corresponding eigenvalue 0. From the result derived in 5.i, we see that:

$$v^{*T} L v^* = \sum_{ij \in E} (v^*(i) - v^*(j))^2 = 0$$

This means that each of the terms in the sum are zero, i.e. $v^*(i) = v^*(j)$ for $(i, j) \in E$. Hence, we know that v^* is non-zero and has a constant for all indices belonging to the same subgraph. Therefore, when computing the inner product, $\langle v^*, v_i \rangle \neq 0 \; \forall i = 1, \ldots, k$.

But eigenvectors of a symmetric matrix are necessarily orthogonal (and a Laplacian matrix is symmetric), therefore v^* cannot be an eigenvector, a contradiction. Hence, there are at most k eigenvectors with eigenvalue 0.

Since there are simultaneously at least k and at most k eigenvectors with corresponding eigenvalue 0, there are exactly k eigenvectors of L with eigenvalue 0. Since L has rank n, this means there are $n - k$ nonzero eigenvalues, as desired.

6. ML Review:

i.

True. H has the VC-dimension of d if d is the largest number of points $(x_1, \dots, x_d) \subset X$ such that for all labelings of $(x_1, \dots, x_d)$, there exists some $f \in H$ that achieves that labeling. Say H is finite.

For a given d, in general we need 2^d classifiers to achieve all 2^d labelings. Since there are only $|H|$ classifiers in the hypothesis class, we must have:

$$2^d \leq |H|$$

Hence:

$$d \leq \log_2 |H|$$

ii.
False! The optimal L2 regressor can, in general, be non-linear, whereas the Lasso regressor is strictly linear (even as n->infinity), so Lasso regression is in general not consistent.

iii:
False. The given $f(x)$ is the Naïve Bayes approximation to the Bayes classifier, which assumes each coordinate is conditionally independent of each other (and thus does not capture potential interdependencies of different features). In general, Naïve Bayes is not the optimal classifier.

iv:
True. For any random vector $X \in \mathbb{R}^d$, the projection in the v direction is: $v^T X$. The variance in this direction is thus:

$$\begin{aligned} &Var(v^T X) = E[(v^T X)^2] - E[v^T X]^2 \\ &= E[(v^T X)^T (v^T X)] - (v^T E[X])^2 \\ &= v^T E[X X^T] v \end{aligned}$$

v:

False. Consider the function:

$$f(x) = x + e^x$$

This function happens to also be continuous, so to test strict convexity we just take the second derivative:

$$f''(x) = e^x > 0 \quad \forall x$$

However, clearly:

$$\lim_{x \to -\infty} f(x) = -\infty$$

In other words, f has no lower bound.