COMS 4774 Unsupervised Learning Fall 2024
Problem Set #0

Brian Chen - bc2924@columbia.edu

August 31, 2024

Problem 1

Part (i)

The l_p norm is defined as

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

First we show the first half of the inequality.

$$\begin{aligned}
\|x\|_1 &= \sum_{i=1}^{d} |x_i| \\
&= \Big(\Big(\sum_{i=1}^{d} |x_i|\Big)^2\Big)^{1/2} \\
&= \Big(\sum_{i=1}^{d} |x_i|^2 + \text{ a bunch of nonnegative cross terms }\Big)^{1/2} \\
&\leq \Big(\sum_{i=1}^{d} |x_i|^2\Big)^{1/2} = \|x\|_2
\end{aligned}$$

This inequality is tight when x is a zero vector as all of the cross terms become zero.

Now we show the second half of the inequality. Noticing that

$$\sqrt{d} = \underbrace{\sqrt{1+1+\ldots}}_{d \text{ ones}} = \|\mathbf{1}\|$$

where $\mathbf{1}$ is the ones vector in $\mathbb{R}^d$, we see that

$$\|x\|_2 \cdot \sqrt{d} = \|x\|_2 \|\mathbf{1}\|_2$$

$$\text{(Cauchy-Schwarz)} \geq |x \cdot \mathbf{1}|$$

$$\geq \left| \sum_{i=1}^{d} \max(x_i) \right|$$

$$\geq \sum_{i=1}^{d} |x_i| = \|x\|_1$$

The inequality is tight again when x is a zeros vector.

Part (ii)

Proceeding identically from above, notice that

$$\|x\|_1 \overset{?}{\leq} \|x\|_p$$

$$\sum_{i=1}^{d} |x_i| \overset{?}{\geq} \Big(\sum_{i=1}^{d} |x_i|^p \Big)^{1/p}$$

$$\Big(\sum_{i=1}^{d} |x_i| \Big)^p \overset{?}{\geq} \sum_{i=1}^{d} |x_i|^p$$

$$\sum_{i=1}^{d} |x_i|^p + \text{nonnegative cross terms} \geq \sum_{i=1}^{d} |x_i|^p$$

Since the terms inside the brackets are nonnegative due to the absolute value, we see that the LHS is indeed greater than or equal to the RHS, which proves the problem statement that for any $x \in \mathbb{R}^d$ and $p \geq 1$, $\|x\|_1 \geq \|x\|_p$.

Part (iii)

For any $1 \leq p \leq q$, we have

$$\Big(\sum_{i=1}^{d} |x_i|^p \Big)^{1/p} \overset{?}{\geq} \Big(\sum_{i=1}^{d} |x_i|^q \Big)^{1/q}$$

$$\Big(\sum_{i=1}^{d} |x_i|^p\Big)^{q/p} \overset{?}{\geq} \sum_{i=1}^{d} |x_i|^q$$

$$\geq \sum_{i=1}^{d} |x_i|^p$$

Since $q \geq p$, $q/p \geq 1$. Thus, the LHS is raised to the power of something greater than or equal to 1. Since the terms on the inside are all nonnegative, raising to the power of something > 1 trivially can only increase the term. Therefore, $\|x\|_p \geq \|x\|_q$ for $p \leq q$.

Part (iv)

Proof. By the Hölder inequality,

$$\sum_{i=1}^{d} |x_i| \leq \|x\|_a \cdot d^{1/b}$$

since $\|\mathbf{1}\|_b = (\sum_{i=1}^{d} 1^b)^{1/b}$. Now choose $a = \frac{q}{p}$, $b = \frac{q}{q-p}$, which satisfies the Holder condition of $\frac{1}{a} + \frac{1}{b} = 1$. We get

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

$$\Big(\sum_{i=1}^{d} |x_i|^p\Big)^{\frac{1}{p}} \leq \sum_{i=1}^{d} \Big(|x|^q\Big)^{\frac{1}{q}} \cdot d^{\frac{q-p}{pq}}$$

$$\|x\|_p \leq \|x\|_q \cdot d^{\frac{1}{p}-\frac{1}{q}}$$

□

Problem 2

Let v be a unit vector in $\mathbb{R}^d$. Then the directional derivative of f in the direction of v is[1]

$$\nabla_v f(x) = \lim_{h \to 0} \frac{f(x + vh) - f(x)}{h}$$

Recall that we can express $f(x)$ alternatively using its Taylor expansion,

$$f(x + vh) = f(x) + \nabla f(x) vh + \dots$$

where the remaining terms go as higher powers of h. The directional derivative is thus

$$\begin{aligned} \nabla_v f(x) &= \lim_{h \to 0} \frac{(f(x) + \nabla f(x) vh + \dots) - f(x)}{h} \\ &= \|\nabla f(x) \cdot v\| \\ \text{(Cauchy-Schwarz)} &\leq \|\nabla f(x)\| \|v\| \end{aligned}$$

Since we see that

$$\nabla_v f(x) \leq \|\nabla f(x)\| \|v\|$$

the directional derivative is maximized when $v = \frac{\nabla_v f(x)}{\|\nabla f(x)\|}$ due to the inequality. This clearly also implies that the inequality

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

is strict unless when $u = v$ since the directional derivative is maximized at the above setting of v, and any other value of u will return a smaller value of $\frac{d}{dt} f(x + tu)_{t=0}$.

[1] `https://en.wikipedia.org/wiki/Directional_derivative`

Problem 3

Part (i)

The kernel trick can be employed when our data is nonlinear but we are interested in linear features; for instance, a dataset in the shape of concentric circles does not have interesting principle components, but in a higher dimensional space we may be able to use PCA to discover some structure.

Similarly, data may not exhibit clusters discoverable by k-means in some low dimensional space (e.g. a set of concentric rings in $\mathbb{R}^2$), but by raising the data into a higher dimensional space we could be able to discover these distinct clusters.

Part (ii)

By definition, the eigenvector/eigenvalue pairs (μ_i, u_i) for $X^\mathsf{T}X$ satisfy

$$X^\mathsf{T}Xu_i = \mu_i u_i$$

If we left multiply this expression by the data matrix X, we get

$$XX^\mathsf{T}Xu_i = X\mu_i u_i$$

$$\Rightarrow XX^\mathsf{T}Xu_i = \mu_i Xu_i$$

where we can commute the μ_i term out since it is a scalar. From here we can immediately see that this expression is in the form of an eigenvalue equation; specifically, let $Xu_i \equiv v_i$. Then

$$XX^\mathsf{T}v_i = \mu_i v_i$$

Thus, the eigenvalue and eigenvector pairs of XX^T are related to (μ_i, u_i) by

$$\boxed{\lambda_i = \mu_i; v_i = Xu_i}$$

Part (iii)

From COMS4771, we know that the objective for general-k PCA is

$$\operatorname*{arg\,min}_{Q\in\mathbf{R}^{d\times k}, Q^\mathsf{T}Q=1} \frac{1}{n}\sum_{i=1}^{n} \left\|\vec{x_i} - QQ^\mathsf{T}\vec{x_i}\right\|^2 = \operatorname*{arg\,min}_{Q\in\mathbf{R}^{d\times k}, Q^\mathsf{T}Q=1} \frac{1}{n}\sum_{i=1}^{n} \left\|X - QQ^\mathsf{T}X\right\|_F^2$$

$$\propto \operatorname*{arg\,max}_{Q\in\mathbf{R}^{d\times k}, Q^\mathsf{T}Q=I} \operatorname{tr}\left(Q^\mathsf{T}(\frac{1}{n}XX^\mathsf{T})Q\right)$$

XX^T is symmetric and positive semidefinite, so we can use spectral decomposition to rewrite it.

$$\frac{1}{n}XX^\mathsf{T} = W\Lambda W^{-1} = \sum_i \lambda_i v_i v_i^\mathsf{T}$$

where $\{v_1, \ldots, v_i\}$ are the orthonormal eigenvectors of $\frac{1}{n}XX^\mathsf{T}$ and $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$ are the associated eigenvalues, where we put the largest eigenvalue on top by convention.

We see that

$$\begin{aligned} Q^\mathsf{T}[\sum_i \lambda_i v_i v_i^\mathsf{T}]Q &= \sum_i \lambda_i Q^\mathsf{T} v_i v_i^\mathsf{T} Q \\ &= \sum_i \lambda_i [Q^\mathsf{T} v]^\mathsf{T} [Q^\mathsf{T} v] \\ &= \sum_i \lambda_i (Qv_i)^2 \end{aligned}$$

To maximize this, we want to find

$$\max_{Qv_i} \sum_i \lambda (Qv_i)^2 \qquad \text{s.t.} \sum_i (Qv_i) = k$$

which has optimal solution of $Qv_1 = \cdots = Qv_k = 1, Qv_{k+1} = \cdots = Qv_n = 0$, or in other words the top k eigenvectors of $\frac{1}{n}XX^\mathsf{T}$.

Part (iv)

To derive kernelized PCA[2], we need to recompute the PCA equation except with the higher-dimensional embeddings $\phi(x_i)$. Assuming we have a kernel function $K(x_i, x_j)$ that can calculate the inner product $\langle \phi(x_i), \phi(x_j) \rangle$

We need to evaluate the covariance of the lifted data,

$$C = \frac{1}{n} \sum_{i=1}^n \phi(x_i)\phi(x_i)^\mathsf{T}$$

First we need to find the eigenvectors of of C. Let v be an eigenvector of C. Then we have

$$Cv = \lambda v$$

$$\frac{1}{n} \sum_{i=1}^n \phi(x_i)\phi(x_i)^\mathsf{T} = \lambda v$$

Let $a = \frac{1}{\lambda^n}\phi(x_j)^\mathsf{T}$. Then we see

$$v = \sum_{i=1}^n a_j \phi(x_j)$$

[2]When studying for the final for COMS4771 last semester, I did a practice problem involving kernelizing PCA that I found from the Berkeley Intro ML class. That process is essentially replicated here.

so therefore the eigenvalues of C can be expressed as a linear combination of the high dimensional embeddings. Similarly, if a is an eigenvector of the kernel matrix K with eigenvalue λ_a, then we have

$$Ka = \lambda_a a$$

$$\phi(x)\phi(x)^\mathsf{T} = \lambda a$$

$$\frac{1}{n}\phi(x)^\mathsf{T}\phi(x)\phi(x)^\mathsf{T}v = \frac{\lambda_a}{n}\phi(x)^\mathsf{T}a$$

$$Cv = \frac{\lambda_a}{n}v$$

where $v = \phi(x)^\mathsf{T}a$. Thus, $\phi(x)^\mathsf{T}a$ is also an eigenvector of C with eigenvalue $\frac{\lambda_a}{n}$. Therefore we can express the eigenvectors of the covariance of the high dimensional (lifted) points without needing to calculate the high dimension covariance matrix C explicitly.

Now to project a test point z onto a principle component $v = \phi(x)^\mathsf{T}a$, we simply take the dot product:

$$\phi(z)^\mathsf{T} \cdot v = \phi(z)\sum_{j=1}^{n} a_j\phi(x_j)$$

$$= \sum_{j=1}^{n} a_j\phi(z)^\mathsf{T} \cdot \phi(x_j)$$

$$= \sum_{j=1}^{n} a_j K(z, x_j)$$

for any kernel function K.

Part (v)

Lemma 1. *Let $c(S)$ be the center of mass of a set of points S, and let z be some arbitrary point. Then $\sum_{x\in S}||x - z||^2 - \sum_{x\in S}||x - c(S)||^2 = |S| \cdot ||c(S) - z||^2$.*

Proof. Note that $||u - v||^2 = (u - v)^T(u - v) = u^Tu - 2u^Tv + v^Tv$, so we can expand the

LHS.

$$\begin{aligned} LHS &= \sum_{x \in S} ||x - z||^2 - \sum_{x \in S} ||x - c(S)||^2 \\ &= \sum_{x \in S} (x^T x - 2x^T z + z^T z) + \sum_{x \in S} (x^T x - 2x^T c(S) + c(S)^T c(S)) \\ &= \sum_{x \in S} (x^T x - 2x^T z) + |S| z^T z + \sum_{x \in S} (x^T x - 2x^T c(S)) + |S| c(S)^T c(S) \end{aligned}$$

Now note that $\sum_{x \in S} x = |S| \cdot c(S)$ by definition, so

$$\begin{aligned} LHS &= \sum_{x \in S} x^T x - 2z^T \ |S| \cdot c(S) + |S| \cdot ||z||^2 - \sum_{x \in S} x^T x - 2c^T(S) \ |S| c(S) + |S| \cdot ||c(S)||^2 \\ &= -2z^T \ |S| \cdot c(S) + |S| \cdot ||z||^2 - 2|S| \cdot c(S)^T c(S) + |S| \cdot ||c(S)||^2 \\ &= -2z^T \ |S| \cdot c(S) + |S| \cdot ||z||^2 - 2|S| \cdot ||c(S)||^2 + |S| \cdot ||c(S)||^2 \\ &= |S| \Big[||z||^2 - 2z^T c(S) + ||c(S)||^2 \Big] \\ &= |S| \cdot ||c(S) - z||^2 \end{aligned}$$

□

Since the RHS of this expression is nonnegative due to the nonnegativeness of the norm, we conclude that we minimize the expression when $\|c(S) - z\|^2 = 0$, or equivalently if $c(S) = z$. Thus, the k-means cost function is minimized when $z = c(S)$ where $c(S) = \frac{1}{n} \sum_{i=1}^{n} x_i$.

Part (vi)

Again using the fact that $||u - v||^2 = u^T u - 2u^T v + v^T v$, we see that

$$\|x_i - c\|_2^2 = \|x_i\|_2^2 - 2x_i^\mathsf{T} c + \|c\|_2^2$$

so therefore, the k-means cost function can be written as

$$\begin{aligned} \sum_{i=1}^{n} \|x_i - c\|_2^2 &= \sum_{i=1}^{n} \|x_i\|^2 - 2(x \cdot c) + \sum_{i=1}^{n} \|c\|^2 \\ &= \sum_{i=1}^{n} \|x_i\|^2 - 2K(x, c) + n \cdot \|c\|^2 \end{aligned}$$

for the kernel function $K(x, y) = x \cdot y$.

Part (vii)

In the higher dimensional space we aim to minimize the cost function with respect to a higher dimensional embedding,

$$\text{cost} = \sum_{x \in \mathcal{X}} \min_{c \in \mathcal{C}} \|\phi(x) - c\|^2$$

Using the expression found above, we note that the Euclidean distance can be found as

$$\|\phi(x) - c\|^2 = \phi(x) \cdot \phi(x) - 2\phi(x) \cdot \phi(c) + n\phi(c) \cdot \phi(c)$$

which can be rewritten with the Kernel function $K(x, y)$ to yield

$$\|\phi(x) - c\|^2 = K(x, x) - 2K(x, c) + nK(c, c) \tag{1}$$

Then, the k-means algorithm can be rewritten straightforwardly where we instead choose cluster assignments based on this calculation of the "nearest" center.

Algorithm 1: Kernelized k-means

Data: points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, number of clusters k
Result: location of k centers $c \in \mathbb{R}^d$ to minimize cost function
while *not converged* **do**
  initialize k centers randomly;
  for *point x in X* **do**
    assign x to the nearest center based on 1
  end
  for *cluster c in C* **do**
    recompute cluster centers as average of all distances to points in cluster based on 1
  end
end
return cluster centers C

Part (viii)

Kernelizing these algorithms allows us to take advantage of higher-dimensional data representations (e.g. the cases discussed in part (i), where useful structure may only reveal themselves only when lifted into higher dimension) without too much computational overhead.

Specifically, recall that explicitly calculating the higher dimensional space embeddings of points can take up to $O(np)$ time, where p is the dimension of the higher dimensional space and n is the number of points. By using the kernel trick, we can implicitly vcalculate the values of interest in the higher dimensional space without needing the computational cost of actually transforming the data into the high dimensional space; for a low-dimensional space of dimension $d \ll p$, we can compute the kernel matrix in just $O(n^2 d)$ which is often much faster.

Part 4

Part (i)

The probability density function of the chi square distribution is

$$f_Y(y) = \frac{1}{2^{n/2}\Gamma(n/2)} y^{(n/2)-1} e^{-y/2}$$

The moment generating function can thus be computed as

$$\begin{aligned}
M(x) = \mathbb{E}[e^{tY}] &= \int_0^\infty e^{ty} f_Y(y)\ dy \\
&= \int_0^\infty e^{ty}\ \frac{1}{2^{n/2}\Gamma(n/2)} y^{(n/2)-1} e^{-y/2}\ dy \\
&= \frac{1}{2^{n/2}\Gamma(n/2)} \int_0^\infty e^{ty}\ y^{(n/2)-1} e^{-y/2}\ dy \\
&= \frac{1}{2^{n/2}\Gamma(n/2)} \int_0^\infty\ y^{(n/2)-1} e^{y(t-1/2)}\ dy \\
\text{(u-sub with } u = y(t-1/2)) &= \frac{1}{2^{n/2}\Gamma(n/2)} \int_0^\infty\ \Big(\frac{u}{t-1/2}\Big)^{(n/2)-1} e^u\ du \\
&= \frac{1}{2^{n/2}\Gamma(n/2)(t-1/2)^{n/2}} \int_0^\infty\ u^{(n/2)-1} e^u\ du \\
\text{(by definition of the gamma function)} &= \frac{\Gamma(n/2)}{2^{n/2}\Gamma(n/2)(t-1/2)^{n/2}(-1)^{n/2}} \\
&= \boxed{(1-2t)^{-n/2}}
\end{aligned}$$

Part (ii)

Via the Chernoff bounding technique,

$$\Pr[Y > a] \leq \frac{\mathbb{E}[e^{tY}]}{e^{ta}}$$

so for $a = (1+\epsilon)^2 n$ and $\mathbb{E}[e^{tY}] = (1-2t)^{-n/2}$ we have

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

We choose t to minimize the expression on the right side by taking the derivative with respect to t and equating it with zero.

$$\frac{d}{dt}\Big[\frac{(1-2t)^{-n/2}}{e^{t(1+\epsilon)^2 n}}\Big] = \frac{e^{t(1+\epsilon)^2 n}\frac{d}{dt}[(1-2t)^{-n/2}] - (1-2t)^{-n/2}\frac{d}{dt}e^{t(1+\epsilon)^2 n}}{e^{2t(1+\epsilon)^2 n}}$$

$$= \frac{e^{t(1+\epsilon)^2 n} n\,(1-2t)^{-\frac{n}{2}-1} - (1-2t)^{-n/2}(1+\epsilon)^2 n e^{t(1+\epsilon)^2 n}}{e^{2t(1+\epsilon)^2 n}}$$

The minimum is found at the value of t such that

$$e^{t(1+\epsilon)^2 n} n\,(1-2t)^{-\frac{n}{2}-1} - (1-2t)^{-n/2}(1+\epsilon)^2 n e^{t(1+\epsilon)^2 n} = 0$$

$$\Rightarrow (1-2t)^{-1} = (1+\epsilon)^2$$

$$\Rightarrow t = \frac{1-(1+\epsilon)^{-2}}{2}$$

Substituting this value of t into the Chernoff bound yields

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

$$\leq \frac{(1+\epsilon)^n}{e^{-2\epsilon^2 n}}$$

$$\leq e^{-2n\epsilon^2}$$

where we use the fact that $(1+\epsilon)^n \to 1$ as $\epsilon \to 0$. Thus, we have shown the upper bound of the probability indeed has the form $e^{-cn\epsilon^2}$.

Problem 5

Part (i)

The graph Laplacian G is defined as $L = D - A$, where D is the degree matrix of G and A is the adjacency matrix of G. Note, then, that the Laplacian can alternatively be expressed as

$$L_{ij} = \begin{cases} \deg(i) & \text{if } i = j \\ -1 & \text{if } (i,j) \in E \\ 0 & \text{else} \end{cases}$$

Let f be a vector in $\mathbb{R}^V$. Then

$$(Lf)_u = f_u \cdot \deg(u) - \sum_{(u,v)\in E} f(v)$$

$$= \sum_{(u,v)\in E} f(u) - f(v)$$

The first line can be seen with any toy example of a Laplacian multiplied by a vector in $\mathbb{R}^d$; the i-th diagonal term in L multiplies with the i-th element of f, which corresponds to the degree of i times $f(i)$, and the remaining terms are the weighted sums of the edges.

Similarly, if we multiply by f^T in the front, we see

$$f^\mathsf{T} L f = f^\mathsf{T} \sum_{(u,v)\in E} f(u) - f(v)$$

$$= \sum_v f(v) \cdot \sum_{(u,v)\in E} f(u) - f(v)$$

Notice that the second term summation is overcounting all of the vertex pairs, so we can rewrite it as

$$f^\mathsf{T} L f = \sum_{v<u:(u,v)\in E} f(u)(f(u) - f(v)) + \sum_{u<v:(u,v)\in E} f(v)(f(v) - f(u))$$

$$= \sum_{e\in E} (f(u) + f(v))(f(u) - f(v))$$

$$= \boxed{\sum_{e\in E} (f(u) - f(v))^2}$$

Part (ii)

The adjacency matrix of a graph with k connected components is block diagonal with k blocks, because each connected component will only have nonzero entries in the "block" that corresponds to the connected vertices. Then note that the Laplacian for a connected component is guaranteed to have exactly one eigenvector with eigenvalue 1, as

$$L\mathbf{1} = (D - A)\mathbf{1} = D\mathbf{1} - A\mathbf{1}$$

and since the $A\mathbf{1}$ just sums up every adjacent vertex, each entry of $A\mathbf{1}$ is just the degree of the vertex. Therefore

$$L\mathbf{1} = 0$$

so $\mathbf{1}$ is an eigenvector of L with eigenvalue 0.

Then since the characteristic polynomial of a block diagonal matrix obeys [3]

$$\det(A - \lambda\mathbf{I}) = \det(A_1 - \lambda\mathbf{I}) \dots \det(A_k - \lambda\mathbf{I})$$

where A_i is the i-th block diagonal matrix, it follows that the overall Laplacian matrix has eigenvalue 0 with multiplicity k, one for each connected component of the Laplacian. Therefore, the overall Laplacian has exactly $n - k$ nonzero eigenvalues.

[3] https://math.stackexchange.com/questions/1307998/how-to-find-the-eigenvalues-of-a-block-diagonal-ma

Problem 6

Part (i)

True. By definition the VC dimension of a hypothesis class $\mathcal{H}$ is the cardinality of the largst set that the hypothesis class can shatter, e.g. perfectly classify. Since for a binary classification algorithm there can be up to 2^d different possible classifications, the VC dimension is $\log_2 |\mathcal{H}|$.

Part (ii)

False. Intuitively, since lasso regression encourages sparse solutions, we should not expect it to approach the error of the optimal L_2 regressor since if the true weights parameter should *not* be sparse, then the L_1 penalty may not give us the correct weights.

Recall that lasso regression aims to find a weights vector w for a given data set (x, y) according to the objective

$$\text{minimize } \|X\vec{w} - y\|^2 + \lambda\|w\|_1$$

Let β_{lasso} be the lasso loss and β^* be the optimal loss. Then via the bias-variance tradeoff we have

$$\begin{aligned}
\mathbb{E}[\lim_{n\to\infty} \|\beta_{\text{lasso}} - \beta^*\|_2^2] &= \lim_{n\to\infty} \left[\|\mathbb{E}[\beta_{\text{lasso}}] - \beta^*\|_2^2 + \text{Var}(\beta_{\text{lasso}})\right] \\
&= \lim_{n\to\infty} \|\mathbb{E}[\beta_{\text{lasso}}] - \beta^*\|_2^2 \\
&> 0
\end{aligned}$$

since the variance goes to zero as the number of samples approaches infinity, but since the lasso estimate is not generally the same as the optimal L_2 estimate, the expected difference between the error of the classifications is greater than 0.

Part (iii)

False. The Bayes optimal classifier is

$$\begin{aligned}
f(x) &= \arg\max_{y\in\mathcal{Y}} \Pr[Y = y \mid X = \vec{x}] \\
&= \arg\max_{y\in\mathcal{Y}} \left(\Pr[X = \vec{x} \mid Y = y] \cdot \frac{\Pr[Y = y]}{\Pr[X = \vec{x}]}\right) \\
&= \arg\max_{y\in\mathcal{Y}} \left(\Pr[Y = y] \cdot \frac{\Pr[X = \vec{x} \mid Y = y]}{\Pr[X = \vec{x}]}\right)
\end{aligned}$$

which is not necessarily the same as the expression in the problem statement.

Part (iv)

True. By definition, the covariance matrix of a vector of random variables is

$$\mathrm{Cov}(X) = \mathbb{E}[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^\mathsf{T}]$$

Recall that the component of a vector $X \in \mathbb{R}^d$ in the direction of a unit vector v can be found with the (linear operator) dot product. Therefore we are interested in the covariance of $v \cdot X = v^\mathsf{T} X$.

By linearity of expectation,

$$\mathbb{E}[v^\mathsf{T} X] = v^\mathsf{T} \mathbb{E}[X]$$

so the covariance is

$$\begin{aligned}
\mathrm{Cov}(X) &= \mathbb{E}[(v^\mathsf{T} X - v^\mathsf{T}\mathbb{E}[X])(v^\mathsf{T} X - v^\mathsf{T}\mathbb{E}[X])^\mathsf{T}] \\
&= \mathbb{E}[(v^\mathsf{T} X)(v^\mathsf{T} X)^\mathsf{T}] \\
&= \mathbb{E}[(v^\mathsf{T} X)(X^\mathsf{T} v)] \\
\text{(linearity of expectation)} \quad &= \boxed{v^\mathsf{T}\mathbb{E}[XX^\mathsf{T}]v}
\end{aligned}$$

Part (v)

True. By definition, a function $f : \mathbb{R}^d \to \mathbb{R}$ is strictly convex[4] if for all $x, y \in \mathbb{R}^d$ and $\theta \in \{0, 1\}$,

$$\theta f(x) + (1 - \theta) f(y) > f(\theta x + (1 - \theta) y)$$

It suffices to prove that f has exactly one local minimum, as then f is bounded from below at the minimum value.

Lemma 2. *If f is strictly convex, then it has no more than one minimum.*

Proof. By contradiction. Assume f attains minimums at x_1 and x_2, with $x_1 \neq x_2$. Without

[4] `https://en.wikipedia.org/wiki/Convex_function`

loss of generality, let $f(x_1) \leq f(x_2)$. Then we have

$$\theta f(x_1) \leq \theta f(x_2)$$

$$\theta f(x_1) - \theta f(x_2) \leq 0$$

$$\theta f(x_1) + (1-\theta) f(x_2) \leq f(x_2)$$

Since f is strictly convex, the LHS is bounded.

$$f(\theta x_1 + (1-\theta) x_2) < f(x_2)$$

but this can only be the case if $x_1 = x_2$, which is a contradiction. Thus, f has at most one local minimum. □

Since f has a unique minimizer, it is bounded from below by the value of this minimum. (An example of a convex but not strictly convex function is a non-horizontal line, which is clearly not bounded from below).