STAT 37710: Homework 2

Caleb Derrickson

April 17, 2024

Collaborators: The TA's of the class, as well as Kevin Hefner, and Alexander Cram, and Alice Yang.

Contents

Problem 1

Let A be a symmetric $d \times d$ matrix.

Problem 1, part a

Show that if **v** and **v**' are two eigenvectors of A with corresponding eigenvalues $\lambda \neq \lambda'$, then **v** is orthogonal to **v**'.

Solution:

We have that $A \in \mathbb{R}^{d \times d}$, so $A^\intercal \in \mathbb{R}^{d \times d}$. Then, we have the following:

$$(\mathbf{v}')^\intercal A\mathbf{v} = (\mathbf{v}')^\intercal \mathbf{v}$$

Note that the quadratic form also has the following form

$$(\mathbf{v}')^\intercal A\mathbf{v} = (A^\intercal \mathbf{v}')^\intercal \mathbf{v} = \lambda'(\mathbf{v}')^\intercal \mathbf{v}.$$

These two equations are equivalent, so we must then have

$$(\mathbf{v}')^\intercal \mathbf{v} = \lambda'(\mathbf{v}')^\intercal \mathbf{v} \iff (\lambda - \lambda')(\mathbf{v}')^\intercal \mathbf{v} = 0.$$

Since $\lambda \neq \lambda'$, we have that $(\mathbf{v}')^\intercal \mathbf{v} = 0$.

Problem 1, part b

Show that if $\mathcal{S}$ is a set of eigenvectors of A with the same eigenvalue λ and $\mathcal{S}$ spans a subspace V of $\mathbb{R}^d$ of dimension k, then one can find k mutually orthogonal vectors $\mathbf{v}^1, \mathbf{v}^2, ..., \mathbf{v}^k$ such that (a) $V = \text{span}\{\mathbf{v}^1, ..., \mathbf{v}^k\}$, and (b) each $\mathbf{v}^{(i)}$ is an eigenvector of A with eigenvalue λ.

Solution:

Let $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k$ be the eigenvectors with same eigenvalue λ. These eigenvectors are not guaranteed to by mutually orthogonal - we can however, recover a set or orthogonal vectors $\mathbf{w}_1, ..., \mathbf{w}_k$ via Gram Schmidt. These $\mathbf{w}_i$ are a linear combination of the $\mathbf{v}_i$'s , thus will have the same eigenvalue as the $\mathbf{v}_i$'s. Since the $\mathbf{w}_i$'s are orthogonal, we have that $V = \text{span}\{\mathbf{w}_1, ..., \mathbf{w}_k\}$ with dimension k. Therefore, a set of vectors has been found.

Problem 1, part c

Explain why the above two statements imply that A has an eigenvector decomposition of the form

$$A = \sum_{i=1}^{d} \lambda_i \mathbf{v}_i \mathbf{v}_i^\intercal,$$

where each $\mathbf{v}_i$ is a unit eigenvector of A and λ_i is the corresponding eigenvalue.

Solution:

Since A is symmetric, all its eigenvalues are real valued. Thus, we assume we have d real valued eigenvalues λ_i. Note that these eigenvalues are not necessarily unique, however. To this extent, suppose we have m distinct eigenvalues, each with at least one corresponding eigenvector. Suppose for $\mathbf{v}_i$, we have that $A\mathbf{v}_i = \lambda \mathbf{v}_i$. By the previous part, we saw that we could recover a set of orthogonal vectors $\mathbf{w}_1, ..., \mathbf{w}_k$, which I will call these $\mathbf{v}_1, ..., \mathbf{v}_k$ for the sake of simplicity. If we gather all eigenvectors corresponding to the same eigenvalue, say $\mathbf{v}_1, ..., \mathbf{v}_k$, we then have $A\mathbf{v}_1 = \lambda \mathbf{v}_1, ..., A\mathbf{v}_k = \lambda \mathbf{v}_k$. Expressing these in the form of matrices, we have $AV_i = \Lambda_i V_i$, where $\Lambda_i = \lambda_i \mathbb{I}$ and V_i is the matrix whose columns are formed by the eigenvectors of λ_i. Assume without loss of generality that the eigenvectors have been normalized so that $\mathbf{v}_i^\intercal \mathbf{v}_j = \delta_{ij}$. This implies that $V_i^\intercal V_i = \mathbb{I}$, but not necessarily that $V_i V_i^\intercal = \mathbb{I}$. We will show that this is the case.

Suppose we have a vector $\mathbf{x}$ which is a linear combination of our eigenvectors $\mathbf{v}_1, ..., \mathbf{v}_k$. Then we see that

$$V_i V_i^\intercal \mathbf{x} = \left(\sum_{i=1}^{k} \mathbf{v}_i \mathbf{v}_i^\intercal \right) \sum_{i=1}^{k} \alpha_i \mathbf{v}_i = \sum_{i=1}^{k} \mathbf{v}_i \mathbf{v}_i^\intercal (\alpha_i \mathbf{v}_i) + 0 = \sum_{i=1}^{k} \alpha_i \mathbf{v}_i = \mathbf{x}.$$

Note the cross terms in the third expression cancel out, since the eigenvectors are orthogonal. Therefore, the matrix $V_i V_i^\intercal$ acts like the identity matrix of size k for vectors which span the given eigenvectors. Therefore,

$$AV_i = \Lambda_i V_i \iff A = AV_i V_i^\intercal = V_i \Lambda V_i^\intercal = \sum_{j=1}^{k} \lambda_i \mathbf{v}_j^{(i)} (\mathbf{v}_j^{(i)})^\intercal$$

This isn't *necessarily* true, however, since the dimensions don't match up. This is no problem, since we can simply sum over all subspaces V_i. Since we said that there will be m distinct eigenvalues, we have m distinct subspaces, which we are summing over. Therefore, we have that

$$A = \sum_{i=1}^{d} \lambda_i \mathbf{v}_i \mathbf{v}_i^\intercal,$$

since eigenvectors corresponding to difference eigenvalues are inherently orthogonal. Therefore, the statement has been shown.

Problem 1, part d

Assume that $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_d$. Show that

$$\underset{\mathbf{w}\in\mathbb{R}^d\setminus\{0\}}{\operatorname{argmin}} \frac{\mathbf{w}^\intercal A\mathbf{w}}{\|\mathbf{w}\|^2} = \mathbf{v}_1 \quad \text{and} \quad \underset{\mathbf{w}\in\mathbb{R}^d\setminus\{0\}}{\operatorname{argmax}} \frac{\mathbf{w}^\intercal A\mathbf{w}}{\|\mathbf{w}\|^2} = \mathbf{v}_d.$$

Solution:

Suppose we have that $\mathbf{w}$ is a linear combination of the eigenvectors of A. That is, $\mathbf{w} = \sum_{i=1}^d \alpha_i \mathbf{v}_i$. Then, we have

$$\mathbf{w}^\intercal A\mathbf{w} = \mathbf{w}^\intercal \left(\sum_{i=1}^d \alpha_i\lambda_i\mathbf{v}_i\right) = \sum_{i=1}^d \alpha_i^2\lambda_i.$$

By the definition of the two norm on $\mathbb{R}^d$, the inner product of $\mathbf{w}$ with itself is its norm squared. Therefore, $\|\mathbf{w}\|^2 = \langle\mathbf{w}|\mathbf{w}\rangle = \sum_{i=1}^d \alpha_i^2$. Therefore, the given ratios are of the form,

$$\frac{\mathbf{w}^\intercal A\mathbf{w}}{\|\mathbf{w}\|^2} = \frac{\sum_{i=1}^d \alpha_i^2\lambda_i}{\sum_{i=1}^d \alpha_i^2}.$$

Suppose that λ_j is the smallest eigenvalue of A. Certainly, this implies

$$\frac{\mathbf{w}^\intercal A\mathbf{w}}{\|\mathbf{w}\|^2} = \frac{\sum_{i=1}^d \alpha_i^2\lambda_i}{\sum_{i=1}^d \alpha_i^2} \geq \frac{\lambda_j\sum_{i=1}^d \alpha_i^2}{\sum_{i=1}^d \alpha_i^2} = \lambda_j.$$

This bound is achievable only when $\mathbf{w}$ is a multiple of only the eigenvector corresponding to λ_j, which I will denote as $\mathbf{v}_j$. Aside from $\mathbf{w}$ being the zero vector, this is the lowest bound achievable by any $\mathbf{w}$ in the span of A, since if there were, then λ_j wouldn't be the smallest eigenvalue. Therefore, the first equality has been shown. The second one can be similarly shown with instead utilizing the largest eigenvalue of A, giving us an achievable upper bound.

Problem 2

Let $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ be a dataset of n vectors in $\mathbb{R}^d$ that have already been centered, i.e., $\sum_{i=1}^n \mathbf{x}_i = 0$. Let $\mathbf{p}_1, ..., \mathbf{p}_k$ be a set of k mutually orthogonal unit vectors, and V the subspace that they span.

Problem 2, part a

Given any $\mathbf{x} \in \mathbb{R}^d$, let $\mathbf{x}_V$ be the closest point to $\mathbf{x}$ in V, i.e., $\mathbf{x}_V = \operatorname{argmin}_{\mathbf{y}\in V} \|\mathbf{x} - \mathbf{y}\|^2$. Show that $\mathbf{x}_V$ is given by the orthogonal projection of $\mathbf{x}$ to V, i.e.,

$$\mathbf{x}_V = \sum_{i=1}^k (\mathbf{x} \cdot \mathbf{p}_i)\mathbf{p}_i.$$

Solution:

Denote the projection $P : \mathbb{R}^d \to \mathbb{R}^d$ which projects any vector $\mathbf{x} \in \mathbb{R}^d$ to V. Define P as $P = \sum_{i=1}^k \mathbf{p}_i \mathbf{p}_i^\intercal$.[1] This is equivalent to saying that $P\mathbf{x} = \mathbf{x}_V$ *in spirit* - to show that $P\mathbf{x}$ is of the given form, we need to do some work. First, we will show that $\mathbf{x} - P\mathbf{x} \in V^\perp$. This is straightforward, since for any $\mathbf{p}_j$, we have that

$$(\mathbf{x} - P\mathbf{x})^\intercal \mathbf{p}_j = \mathbf{x}^\intercal \mathbf{p}_j - \left(\sum_{i=1}^d (\mathbf{x}^\intercal \mathbf{p}_i)\mathbf{p}_i \right)^\intercal \mathbf{p}_j = \mathbf{x}^\intercal \mathbf{p}_i - (\mathbf{x}^\intercal \mathbf{p}_i)(\mathbf{p}_i^\intercal \mathbf{p}_i) = 0.$$

Since V is a linear subspace, then for any $\mathbf{y} \in V$, $P\mathbf{x} - \mathbf{y} \in V$. Therefore, the following can be shown: for any $\mathbf{y} \in V$,

$$\begin{aligned}
\|\mathbf{x} - \mathbf{y}\|^2 &= \|\mathbf{x} - P\mathbf{x} + P\mathbf{x} - \mathbf{y}\|^2 \\
&= (\mathbf{x} - P\mathbf{x} + P\mathbf{x} - \mathbf{y})^\intercal (\mathbf{x} - P\mathbf{x} + P\mathbf{x} - \mathbf{y}) \\
&= \|\mathbf{x} - P\mathbf{x}\|^2 + \|P\mathbf{x} - \mathbf{y}\|^2 + 2(\mathbf{x} - P\mathbf{x})^\intercal (P\mathbf{x} - \mathbf{y}).
\end{aligned}$$

What I did above was simple expansion. Since $\mathbf{x} - P\mathbf{x} \in V^\perp$, and $P\mathbf{x} - \mathbf{y} \in V$, then their inner product is zero by definition. The equation above results in $\|\mathbf{x} - \mathbf{y}\|^2 = \|\mathbf{x} - P\mathbf{x}\|^2 + \|P\mathbf{x} - \mathbf{y}\|^2$. [2] This can be bounded below by $\|\mathbf{x} - P\mathbf{x}\|^2$, which can be achieved by setting $y = P\mathbf{x}$. This is then the minimum of the function, since the second term is ≥ 0, and only zero when its argument is zero. Therefore, $P\mathbf{x} = \mathbf{x}_V$. Note by the given form of the projection, we have

$$\mathbf{x}_V = P\mathbf{x} = \sum_{i=1}^k \mathbf{p}_i \mathbf{p}_i^\intercal \mathbf{x} = \sum_{i=1}^k (\mathbf{p}_i \cdot \mathbf{x})\mathbf{p}_i.$$

[1] This is clearly well defined, since $P^2 = P$.

[2] This is pretty much the Pythagorean theorem.

Problem 2, part b

Let $\Phi(V)$ be the mean squared error of projecting $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ to V,

$$\Phi(V) = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i - (\mathbf{x}_i)_V\|^2.$$

Show that $\Phi(V)$ is minimized by setting $\mathbf{p}_1, ..., \mathbf{p}_k$ to be the k leading eigenvectors of the sample covariance matrix $\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\mathsf{T}$.

Solution:

We will go straight into calculations.

$$\operatorname{argmin} \Phi(V) = \operatorname{argmin} \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i - (\mathbf{x}_i)_V\|^2 \qquad \text{(Given.)}$$

$$= \operatorname{argmin} \frac{1}{n}\sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{j=1}^{k} (\mathbf{x}_i^\mathsf{T}\mathbf{p}_j)\mathbf{p}_j \right\|^2 \qquad \text{(Last part.)}$$

$$= \operatorname{argmin} \frac{1}{n}\sum_{i=1}^{n} \left[\mathbf{x}_i^\mathsf{T}\mathbf{x}_i - 2\mathbf{x}_j^\mathsf{T} \sum_{j=1}^{k} (\mathbf{x}_i^\mathsf{T}\mathbf{p}_j)\mathbf{p}_j + \sum_{j=1}^{k} (\mathbf{x}_i^\mathsf{T}\mathbf{p}_j)^2 \right] \qquad \text{(Expanding.)}$$

$$= \operatorname{argmin} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} -(\mathbf{x}_i^\mathsf{T}\mathbf{p}_j)^2 \qquad \text{(Simplifying.)}$$

$$= \operatorname{argmax} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} (\mathbf{x}_i^\mathsf{T}\mathbf{p}_j)^2 \qquad \text{(Flipping signs.)}$$

$$= \operatorname{argmax} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} \mathbf{p}_j^\mathsf{T}(\mathbf{x}_i\mathbf{x}_i^\mathsf{T})\mathbf{p}_j \qquad \text{(Associativity.)}$$

$$= \operatorname{argmax} \sum_{j=1}^{k} \mathbf{p}_j^\mathsf{T} \left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\mathsf{T} \right) \mathbf{p}_j \qquad \text{(Rearranging.)}$$

$$= \operatorname{argmax} \sum_{j=1}^{k} \mathbf{p}_j^\mathsf{T}\hat{\Sigma}\mathbf{p}_j \qquad \text{(Given.)}$$

Supposing that all $\mathbf{p}_j$'s are unique, in order to maximize this function, we just take $\mathbf{p}_1$, ..., $\mathbf{p}_k$ to be the k largest eigenvectors of $\hat{\Sigma}$. This is fine to do, since the covariance matrix is symmetric, so our work in problem 1 is applicable.

Problem 3

Let K be the Gram matrix of n points in $\mathbb{R}^d$ (with $n \geq d$).

Problem 3, part a

Show that $\mathrm{rank}(K) \leq d$.

Solution:

This can be seen via a reinterpretation of matrix multiplication. Suppose we have the matrices A and B, where A has rows of the given $\mathbf{x}$, and B has columns of the given $\mathbf{x}$. Their product is then given as

$$AB = \begin{bmatrix} — & \mathbf{x}_1 & — \\ & \vdots & \\ — & \mathbf{x}_n & — \end{bmatrix} \begin{bmatrix} | & & | \\ \mathbf{x}_1 & \dots & \mathbf{x}_n \\ | & & | \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^\intercal \mathbf{x}_1 & \dots & \mathbf{x}_1^\intercal \mathbf{x}_n \\ & | & \\ \mathbf{x}_n^\intercal \mathbf{x}_1 & \dots & \mathbf{x}_n^\intercal \mathbf{x}_n \end{bmatrix}$$

Therefore, the product of A and B is equal to the Gram matrix. Let us examine the dimensions of the above matrices: we have that $A \in \mathbb{R}^{d \times n}$, $B \in \mathbb{R}^{n \times d}$, and $K \in \mathbb{R}^{d \times d}$. An elementary proof in Linear algebra states that

$$\mathrm{rank}(AB) \leq \min\{\mathrm{rank}(A), \mathrm{rank}(B)\}.$$

Plugging in what we have, that is, $\mathrm{rank}(A) = d, \mathrm{rank}(B) = n$, and $n \geq d$, we have that $\mathrm{rank}(K) = \mathrm{rank}(AB) \leq d$, which is what we wanted to show.

Problem 3, part b

Let $K \in \mathbb{R}^{n \times n}$ be a (symmetric) positive semi-definite matrix of rank r, and let $d \geq r$. Find n points $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$ such that their Gram matrix is K.

Solution:

Since K is symmetric, we have by problem 1 that K can be rewritten as

$$K = Q\Lambda Q^\intercal,$$

where Q is the matrix of eigenvectors of K, and Λ its eigenvalues. Since K is again, symmetric, all entries of the diagonal matrix are real and positive. We can then define $\Lambda^{1/2}$ such that $\Lambda^{1/2}\Lambda^{1/2} = \Lambda$. Define $\mathbf{x}_i$ such that

$$\mathbf{x}_i = \left(\left[Q\Lambda^{1/2}\right]_i\right)^\intercal$$

And define A to have its columns as these $\mathbf{x}_i$'s. From the previous part, we can see that $A^\intercal = B$, therefore,

$$K = AB = AA^\intercal = \left(\left[Q\Lambda^{1/2}\right]\right)\left(\left[Q\Lambda^{1/2}\right]\right)^\intercal = Q\Lambda Q^\intercal.$$

Therefore, a matrix A has been found.

Problem 4

Define the n dimensional centering matrix $P = \mathbb{I} - \frac{1}{n}\mathbb{1}\mathbb{1}^\intercal$, where $\mathbb{1}$ is the n dimensional all ones vector.

Problem 4, part a

Show that P is a projection operator, i.e., that $P^2 = P$.

Solution:

We will go straight into calculations. Note the difference between the characters $\mathbb{I}$ and $\mathbb{1}$. I wanted to use $\mathbb{1}$ since it looks cool.

$$\begin{aligned}
P^2 &= (\mathbb{I} - \frac{1}{n}\mathbb{1}\mathbb{1}^\intercal)(\mathbb{I} - \frac{1}{n}\mathbb{1}\mathbb{1}^\intercal) && \text{(Given.)} \\
&= \mathbb{I} - \frac{2}{n}\mathbb{1}\mathbb{1}^\intercal + \frac{1}{n^2}\mathbb{1}\mathbb{1}^\intercal\mathbb{1}\mathbb{1}^T && \text{(Expanding.)} \\
&= \mathbb{I} - \frac{2}{n}\mathbb{1}\mathbb{1}^\intercal + \frac{1}{n}\mathbb{1}\mathbb{1}^\intercal && (\mathbb{1}^\intercal\mathbb{1} = n.) \\
&= \mathbb{I} - \frac{1}{n}\mathbb{1}\mathbb{1}^\intercal && \text{(Simplifying.)} \\
&= P && \text{(Given.)}
\end{aligned}$$

Problem 4, part b

Show that the kernel of P is the line $U = \{\lambda \mathbb{1}\}$, i.e., $P\mathbf{v} = 0$ if and only if $\mathbf{v} \in U$ or $\mathbf{v} = 0$.

Solution:

If $\mathbf{v} = 0$, then it is easy to see that $P\mathbf{v} = 0$, since P is a linear operator. We then assume that $\mathbf{v} \neq 0$. Then

$$P\mathbf{v} = \mathbf{v} - \frac{1}{n}\mathbb{1}\mathbb{1}^\mathsf{T}\mathbf{x} = 0 \implies \mathbf{x} = \left(\frac{\mathbb{1}^\mathsf{T}\mathbf{x}}{n}\right)\mathbb{1} = \bar{\mathbf{x}}\mathbb{1}$$

where $\bar{\mathbf{x}}$ is the average value of $\mathbf{x}$. Therefore, the kernel of P is shown to be the following form.

Problem 4, part c

Let $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ be a set of n points in $\mathbb{R}^d$, and let $\tilde{G}$ be their centered Gram matrix, $\tilde{G}_{i,j} = (\mathbf{x}_i - \boldsymbol{\mu})^\intercal(\mathbf{x}_j - \boldsymbol{\mu})$, where $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$. Show that

$$\tilde{G} = -\frac{1}{2}PDP,$$

where $D_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$.

Solution:

For the sake of analysis, I will show that $-2\tilde{G} = PDP$. We then have the following:

$$PDP = \left(\mathbb{I} - \frac{1}{n}\mathbb{1}\mathbb{1}^\intercal\right) D \left(\mathbb{I} - \frac{1}{n}\mathbb{1}\mathbb{1}^\intercal\right) \qquad \text{(Given.)}$$

$$= D - \frac{1}{n}\mathbb{1}\mathbb{1}^\intercal D - \frac{1}{n}D\mathbb{1}\mathbb{1}^\intercal + \frac{1}{n^2}\mathbb{1}\mathbb{1}^\intercal D\mathbb{1}\mathbb{1}^\intercal \qquad \text{(Expanding.)}$$

$$\implies [PDP]_{i,j} = \left[D - \frac{1}{n}\mathbb{1}\mathbb{1}^\intercal D - \frac{1}{n}D\mathbb{1}\mathbb{1}^\intercal + \frac{1}{n^2}\mathbb{1}\mathbb{1}^\intercal D\mathbb{1}\mathbb{1}^\intercal\right]_{i,j}$$

$$= \|x_i - x_j\|^2 - \frac{1}{n}\sum_{k=1}^n \|x_k - x_j\|^2 - \frac{1}{n}\sum_{k=1}^n \|x_i - x_k\|^2$$

$$+ \frac{1}{n^2}\sum_{k=1}^n\sum_{\ell=1}^n \|x_k - x_\ell\|^2 \qquad \text{(Expanding.)}$$

$$= x_i^\intercal x_i - 2x_i^\intercal x_j + x_j^\intercal x_j - \frac{1}{n}\sum_{k=1}^n \left[x_k^\intercal x_k - 2x_k^\intercal x_j + x_j^\intercal x_j\right]$$

$$- \frac{1}{n}\sum_{k=1}^n [x_i^\intercal x_i - 2x_i^\intercal x_k + x_k^\intercal x_k]$$

$$+ \frac{1}{n^2}\sum_{k=1}^n\sum_{\ell=1}^n [x_k^\intercal x_k - 2x_k^\intercal x_\ell + x_\ell^\intercal x_\ell] \qquad \text{(Expanding.)}$$

$$= x_i^\intercal x_i - 2x_i^\intercal x_j + x_j^\intercal x_j - \left(\frac{2}{n}\sum_{k=1}^n x_k^\intercal x_k\right) + 2\boldsymbol{\mu}^\intercal x_j - x_j^\intercal x_j$$

$$- x_i^\intercal x_j + 2x_i^\intercal \boldsymbol{\mu} + \sum_{k=1}^n \left[\frac{x_k^\intercal x_k}{n} - \frac{2}{n}x_k^\intercal \boldsymbol{\mu} + \frac{1}{n^2}\sum_{\ell=1}^n x_\ell^\intercal x_\ell\right] \qquad \text{(Grouping, Expanding.)}$$

$$= -2c_i^\intercal x_j - \frac{1}{n}\sum_{k=1}^n x_k^\intercal x_k + 2\boldsymbol{\mu}^\intercal x_j + 2x_i^\intercal \boldsymbol{\mu} - 2\boldsymbol{\mu}^\intercal\boldsymbol{\mu} + \frac{1}{n}\sum_{\ell=1}^n x_\ell^\intercal x_\ell \qquad \text{(Grouping.)}$$

$$= -2(x_i^\intercal x_j + \boldsymbol{\mu}^\intercal\boldsymbol{\mu} - \boldsymbol{\mu}^\intercal x_j - x_i^\intercal\boldsymbol{\mu}) \qquad \text{Simplifying.)}$$

$$= -2(x_i - \boldsymbol{\mu})^\intercal(x_j - \boldsymbol{\mu}) \qquad \text{(Grouping.)}$$

$$= -2\tilde{G}_{i,j} \qquad \text{(Given.)}$$

Problem 5

Locally Linear Embedding (LLE) finds an embedding that maps n high dimensional input vectors $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$, to lower dimensional output vectors $\mathbf{y}_1, ..., \mathbf{y}_n \in \mathbb{R}^p$. In the second phase of the algorithm, $(\mathbf{y}_1, ..., \mathbf{y}_i)$ are found by minimizing the cost function

$$\Psi(\mathbf{y}_1, ..., \mathbf{y}_n) = \sum_{i=1}^{n} \left\| \mathbf{y}_i - \sum_j w_{i,j} \mathbf{y}_j \right\|^2$$

given the weights $(w_{i,j})_{i,j}$ found in the first phase. To make the problem well posed, this optimization is performed subject to the constraints

$$\sum_{i=1}^{n} \mathbf{y}_i = 0, \quad \text{and} \quad \frac{1}{n} \sum_{i=1}^{n} \mathbf{y}_i \mathbf{y}_i^\intercal = \mathbb{I},$$

where $\mathbb{I}$ is the $p \times p$ identity matrix. Derive the eigenvector problem that LLE reduces to.

Solution:

We will go straight into calculations.

$$\Psi = \sum_{i=1}^{n} \left\| \mathbf{y}_i - \sum_j w_{i,j} \mathbf{y}_j \right\|^2 \qquad \text{(Given.)}$$

$$= \sum_{i=1}^{n} \left(\mathbf{y}_i - \sum_j w_{i,j} \mathbf{y}_j \right)^\intercal \left(\mathbf{y}_i - \sum_j w_{i,j} \mathbf{y}_j \right) \qquad \text{(Expanding.)}$$

$$= \sum_{i=1}^{n} \mathbf{y}_i^\intercal \mathbf{y}_i - \mathbf{y}_i^\intercal \sum_j w_{i,j} \mathbf{y}_j - \sum_j w_{j,i} \mathbf{y}_j^\intercal \mathbf{y}_i + \left(\sum_j w_{i,j} \mathbf{y}_j \right)^\intercal \left(\sum_j w_{i,j} \mathbf{y}_j \right) \qquad \text{(Expanding.)}$$

$$= \sum_{i=1}^{n} \mathbf{y}_i^\intercal \mathbf{y}_i - \mathbf{y}_i^\intercal \sum_j w_{i,j} \mathbf{y}_j - \sum_j w_{j,i} \mathbf{y}_j^\intercal \mathbf{y}_i + \left(\sum_j w_{j,i} \mathbf{y}_j^\intercal \right) \left(\sum_j w_{i,j} \mathbf{y}_j \right) \qquad \text{(Transpose into sum.)}$$

$$= \sum_{i=1}^{n} \mathbf{y}_i^\intercal \mathbf{y}_i - \mathbf{y}_i^\intercal \sum_j w_{i,j} \mathbf{y}_j - \sum_j w_{j,i} \mathbf{y}_j^\intercal \mathbf{y}_i + \sum_j \mathbf{y}_j^\intercal \left(\sum_k w_{k,i} w_{i,k} \right) \mathbf{y}_j \qquad \text{(Simplifying, cancelling.)}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} \left[\mathbf{y}_i^\intercal \left(\delta_{i,j} - w_{i,j} - w_{j,i} + \sum_k w_{k,i} w_{i,k} \right) \mathbf{y}_j \right] \qquad \text{(Adding zeros.)}$$

$$= \sum_{i,j} \mathbf{y}_i^\intercal M \mathbf{y}_j \qquad (M_{i,j} \text{ is given above.})$$

Thus, the eigenvector problem has been derived, as in the notes.

Problem 6

The file train35.digits contains 2000 images of 3's and 5's from the famous MNIST database of handwritten digits in text format. The size of each image is 28 × 28 pixels. Each row of the file is a representation one image, with the 28 × 28 pixels flattened into a vector of size 784. A value of 1 for a pixel represents black, and value of 0 represents white. The corresponding row of train35.labels is the class label: +1 for the digit 3, or -1 for the digit 5. The file test35.digits contains 200 testing images in the same format as train35.digits.

Implement the perceptron algorithm and use it to label each test image in test35.digits. Submit the predicted labels in a file named test35.predictions. In the lectures, the perceptron was presented as an online algorithm. To use the perceptron as a batch algorithm, train it by simply feeding it the training set M times. The value of M can be expected to be less than 10, and should be set by cross validation. Naturally, in this context, the "mistakes" made during training are not really errors. Nonetheless, it is instructive to see how the frequency of mistakes decreases as the hypothesis improves. Include in your write-up a plot of the cumulative number of "mistakes" as a function of the number of examples seen.

Since the data is fairly large, for debugging purposes it might be helpful to run your code on just subsets of the 2000 training test images. Depending on your implementation, each run of each algorithm can take several minutes. It may be helpful to normalize each example to unit norm.

Solution:

I have (hopefully) implemented the single class perceptron correctly, where the predictions are either 3 or not 3. I implemented the perceptron in C++, so it will be a little difficult to attach my files to the homework: hopefully I can find a solution. If not, please refer to my Github page[3]. I wrote my code with my local machine in mind - I cannot guarantee that my code will run, or even compile, on non Windows machines. You should be able to download my code, open it up in VsCode, and hit Ctrl + Shift + B to build my code. Then you can hit F5 to run my code. If you cannot build it, or think its too much a hassle, please let me know. The total time for my code to run, which includes all instances and reading and writing data, took less than 2 seconds. Since the problem mentions that it would take several minutes for each run to complete, I find this an improvement. I love C++.

My results for the perceptron are shown in a plot below. This plot shows the accuracy of the perceptron's prediction of training data based on successive training runs. The separate lines represents how much of the training data (in percentage) I am supplying to the perceptron for a given training instance. Something I find interesting is for small percentages of the training data, the perceptron has a prediction accuracy of 100% after a few runs. I would chalk this up to overfitting, and I don't suspect the given predictions to be good on the test data. Because of this, for my test35.predictions, I ran the perceptron on 100% of the training data.

[3] https://github.com/CalebDerrickson/GradCourses/tree/main/Quarter%203/Machine%20Learning/Homework/Homework2

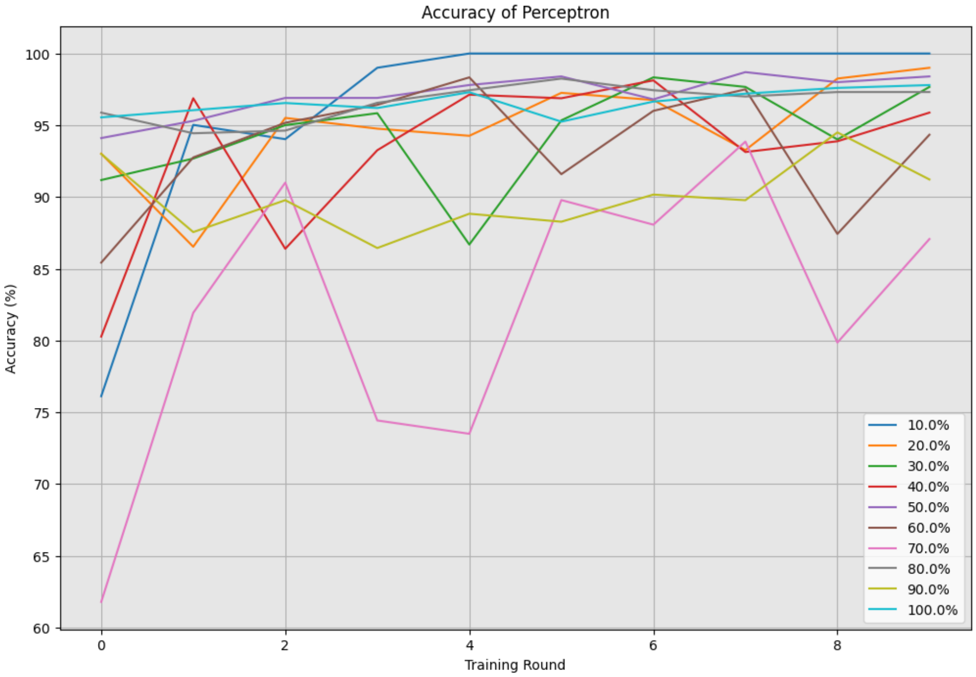

Figure 1: Accuracy of the perceptron on over 10 Training Rounds. Each line represents a different percentage of the training data revealed to the perceptron.