CSCI 567, Fall 2025
Haipeng Luo

Written Assignment #3

Due: Nov 05, 2025, 11:59 pm, PT

Instructions

Total points: 50

Submission: Solutions must be typewritten or neatly handwritten and submitted through gradescope. You can submit multiple times, but only the last submission counts. It is your responsibility to make sure that you submit the right things, and we will *not* consider any regrading requests regarding mistakes in making submissions.

Recall that you have a total of three "late days" for the entire semester, and you can use at most one late day for each written assignment.

Notes on notation:

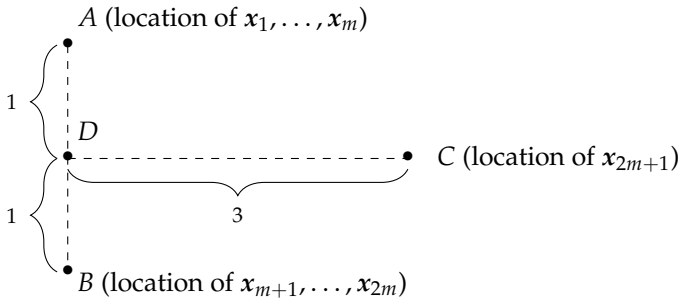
- Unless stated otherwise, scalars are denoted by small letter in normal font, vectors are denoted by small letters in bold font, and matrices are denoted by capital letters in bold font.
- $\|\cdot\|$ means L2-norm unless specified otherwise, i.e., $\|\cdot\| = \|\cdot\|_2$.

Academic integrity: Our goal is to maintain an optimal learning environment. You can discuss the written assignments at a high level with others, but you should not look at any other's solutions. Trying to find solutions online or from any other sources (including ChatGPT and other similar tools) is prohibited, will result in zero grade and will be reported. To prevent any future plagiarism, uploading any materials from this course to the Internet is also prohibited, and any violations will be reported. Please be considerate and help us help everyone get the best out of this course.

Problem 1 Clustering (18 points)

In the initialization of K-means++, a point that is farther away from existing centers is selected with higher probability. So why not always deterministically select the farthest point? In this problem, you will answer this question by comparing the performance of these two initialization methods on a simple example.

Specifically, you are given a dataset with $2m+1$ points shown in the figure below. Here, points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ are all at the exact same location A, points $\boldsymbol{x}_{m+1}, \ldots, \boldsymbol{x}_{2m}$ are all at the exact same location B, and point $\boldsymbol{x}_{2m+1}$ (a single outlier) is at location C. There are no points at location D, but you might need to refer to this location in your answers. The distances between these locations are shown in the figure (the two dashed lines are perpendicular). We will consider running K-means with $K = 2$ clusters and two different initialization methods.



1.1 First, consider running the greedy initialization method, where the first center is selected uniformly at random from the training set, and then the point farthest away from the first center is selected as the second center (tie broken arbitrarily).

- Out of the three configurations of the initial locations of the two centers: A and B, A and C, and B and C, which ones are possible outcomes of this greedy initialization method? Explain briefly. (2 points)

  The only two possibilities are: A and C, B and C. This is because if the first center happens to be at A or B, then $\boldsymbol{x}_{2m+1}$ at C is the (only) farthest point and will be selected as the second center. (On the other hand, if the first center happens to be at C, then obviously the second center has to a point from either A or B.)

  Rubrics: One point for the answer and another point for the reasoning. Any reasonable explanation is acceptable.

- What is the final clustering that K-means will output? Describe both the locations of the two centers and the assignment of each point, and explain briefly. (3 points)

  The final centers will be at C and D, with only the outlier $\boldsymbol{x}_{2m+1}$ assigned to C and all other points assigned to D.

  Due to symmetry, we only need to consider the case when A and C are the initial locations of the two centers. In the first iteration of K-means, all points at A and B will be assigned to the same cluster, while $\boldsymbol{x}_{2m+1}$ is assigned to the other cluster. After averaging, the centers now become D and C, and the assignment of the next iteration remains the same, meaning that the algorithm has converged.

  Rubrics: Two points for the final locations/assignments and another one point for the reasoning. Again, any reasonable explanation is acceptable.

- What is the value of the final K-means objective, that is, the sum of squared L2 distances between each point and its corresponding center? Explain briefly. (2 points)

  The distance between x_{2m+1} and its center C is obviously 0, and the distance between every other point and their center D is 1, so the final K-means objective is $2m$.

  Rubrics: One point for the final answer and another point for the explanation.

1.2 Next, consider running K-means++.

- For the three possible configurations of the initial location of the two centers: A and B, A and C, and B and C, what is the probability of each of them when running the K-means++ initialization method? Show your calculation. Hint: due to symmetry, the probability of A and C and that of B and C are the same, so you only need to calculate either one of them. (6 points)

  The case A and B happens only when the first center is selected from $x_1, \ldots, x_{2m}$, which happens with probability $\frac{2m}{2m+1}$. Without loss of generality, assume that this is from A. Then, the squared L2 distance from this center is 0 to $x_1, \ldots, x_m$, 4 to $x_{m+1}, \ldots, x_{2m}$, and 10 to x_{2m+1}. Therefore, the probability of the second center being selected from B is $\frac{4m}{4m+10} = \frac{2m}{2m+5}$. To sum up, the probability of having A and B as the initial locations is $\frac{2m}{2m+1} \times \frac{2m}{2m+5}$.

  The case A and C happens either when the first center is C and the second center is A, which happens with probability $\frac{1}{2m+1} \times \frac{1}{2}$, or when the first center is A and the second center is C, which happens with probability $\frac{m}{2m+1} \times \frac{10}{4m+10}$ (based on the earlier distance calculation). To sum up, the probability of having A and C as the initial locations is $\frac{1}{2(2m+1)} + \frac{m}{2m+1} \times \frac{5}{2m+5}$. (Due to symmetry, this is clearly also the probability of having B and C as the initial centers.)

  Rubrics: 3 points for each of the two cases: 1 point for the correct answer and 2 points for showing the calculation.

- When the initial centers happen to be at A and B, the outlier x_{2m+1} will eventually be clustered together with either the points at A or the points at B (depending on how you break tie). For simplicity, we assume that m is very large such that the final centers of the two cluster can be approximately treated as A and B. Under this assumption, calculate the final K-means objective value in this case. (1 point)

  Clearly, only the distance between x_{2m+1} and its center is nonzero. The square of this distance is exactly 10, which is also the final K-means objective value.

- Combining your solutions from all previous questions, write down the expected final K-means objective value when running K-means++, and calculate its limit when m goes to infinity. (3 points)

  Based on previous questions, the expected final K-means objective value is

$$\left(\frac{2m}{2m+1} \times \frac{2m}{2m+5}\right) \times 10 + \left(\frac{1}{2m+1} + \frac{2m}{2m+1} \times \frac{5}{2m+5}\right) \times 2m.$$

  Writing this as

$$\left(\frac{2}{2+\frac{1}{m}} \times \frac{2}{2+\frac{5}{m}}\right) \times 10 + \left(\frac{2}{2+\frac{1}{m}} + \frac{2}{2+\frac{1}{m}} \times \frac{10}{2+\frac{5}{m}}\right),$$

  we see that the limit when $m \to \infty$ is clearly $10 + (1 + 5) = 16$.

  Rubrics: 2 points for the K-means objective value, and 1 point for the limit.

1.3 Based on the last two questions, explain briefly why K-means++ is better than the greedy initialization method for this particular dataset. (1 point)

For the greedy initialization method, the K-means objective value $2m$ is unbounded as m increases. On the other hand, the K-means objective for K-means++ is bounded no matter how large m is. Therefore, K-means++ is better and much more robust to outliers.

Rubrics: Any similar reasoning is acceptable.

Problem 2 EM (14 points)

Consider the following probabilistic model to generate a vector $\boldsymbol{x} \in \mathbb{R}^D$. First, sample a number z (hidden variable) according the standard Gaussian distribution (thus zero mean and unit variance) with density

$$p(z) \propto \exp\left(-\frac{1}{2}z^2\right).$$

Then, $\boldsymbol{x}$ is a sampled according to a D-dimensional Gaussian with mean $z\boldsymbol{v}$ and covariance matrix $\sigma^2\boldsymbol{I}$. Here, $\boldsymbol{v} \in \mathbb{R}^D$ is the parameter of the model, $\sigma > 0$ is a fixed number (i.e. not the parameter of the model), and $\boldsymbol{I}$ is the D by D identity matrix. This means that the density of $\boldsymbol{x}$ given z is

$$p(\boldsymbol{x} \mid z; \boldsymbol{v}) \propto \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x} - z\boldsymbol{v}\|_2^2\right).$$

Now you are given a set of N points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^D$ independently sampled according to this model (with corresponding hidden variables $z_1, \ldots, z_N$ unobserved). Finding the exact MLE for the parameter $\boldsymbol{v}$ is difficult, and we will thus use the EM algorithm to solve it approximately following the steps below.

2.1 E-step 1: fixing the parameter $\boldsymbol{v}$, prove that the posterior distribution $q_n(z) \triangleq p(z_n = z \mid \boldsymbol{x}_n; \boldsymbol{v})$ is also a one-dimensional Gaussian distribution. More specifically, show that

$$q_n(z) \propto \exp\left(-\frac{1}{2b^2}(z - a_n)^2\right),$$

where the mean $a_n = \frac{\boldsymbol{x}_n^\top \boldsymbol{v}}{\sigma^2 + \|\boldsymbol{v}\|_2^2}$ and the variance $b^2 = \frac{\sigma^2}{\sigma^2 + \|\boldsymbol{v}\|_2^2}$. (6 points)

Direct calculation shows

$$\begin{aligned}
p(z_n = z \mid \boldsymbol{x}_n; \boldsymbol{v}) &\propto p(z_n = z, \boldsymbol{x}_n; \boldsymbol{v}) && \text{(1 point)} \\
&= p(z)p(\boldsymbol{x}_n \mid z; \boldsymbol{v}) && \text{(1 point)} \\
&\propto \exp\left(-\frac{1}{2}z^2\right) \cdot \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x}_n - z\boldsymbol{v}\|_2^2\right) && \text{(1 point)} \\
&\propto \exp\left(-\frac{1}{2}\left(1 + \frac{\|\boldsymbol{v}\|_2^2}{\sigma^2}\right)z^2 + \frac{\boldsymbol{x}_n^\top \boldsymbol{v}}{\sigma^2}z\right) \\
&\qquad \text{(expand the square and drop a term independent of } z\text{, 1 point)} \\
&\propto \exp\left(-\frac{1}{2}\left(1 + \frac{\|\boldsymbol{v}\|_2^2}{\sigma^2}\right)\left(z - \frac{\boldsymbol{x}_n^\top \boldsymbol{v}}{\sigma^2 + \|\boldsymbol{v}\|_2^2}\right)^2\right) \\
&\qquad \text{(complete the square and drop a term independent of } z\text{, 1 point)} \\
&= \exp\left(-\frac{1}{2b^2}(z - a_n)^2\right). && \text{(1 point)}
\end{aligned}$$

Rubrics: The breakdown of points is only for reference.

2.2 E-step 2: write down the expected complete log-likelihood $Q(\boldsymbol{v})$. Express it in terms of $\boldsymbol{v}, \sigma, \boldsymbol{x}_n, a_n$, and b^2 (for $n = 1, \ldots, N$), and feel free to drop any terms independent of $\boldsymbol{v}$. You can solve this using conclusions from **E-step 1**, even if you have not solved it yet. (Note that in the lecture, we write Q in terms of the parameter, which is $\boldsymbol{v}$ here, and also its previous value; here, the previous value of $\boldsymbol{v}$ is already used in defining a_n and b^2, which is why you do not need to write Q using the previous value of $\boldsymbol{v}$ explicitly.) (5 points)

First, we have

$$Q(\boldsymbol{v}) = \sum_{n=1}^{N} \mathbb{E}_{z_n \sim q_n}[\ln p(\boldsymbol{x}_n, z_n; \boldsymbol{v})] = \sum_{n=1}^{N} \mathbb{E}_{z_n \sim q_n}[\ln p(z_n) + \ln p(\boldsymbol{x}_n \mid z_n; \boldsymbol{v})]. \quad \text{(1 point)}$$

Plugging in the probabilistic model and dropping all terms independent of $\boldsymbol{v}$, we have

$$-\frac{1}{2\sigma^2} \sum_{n=1}^{N} \mathbb{E}_{z_n \sim q_n}[\|\boldsymbol{x}_n - z_n \boldsymbol{v}\|_2^2]. \quad \text{(1 point)}$$

Expanding the square and dropping the terms independent of $\boldsymbol{v}$ again gives:

$$-\frac{1}{2\sigma^2} \sum_{n=1}^{N} \mathbb{E}_{z_n \sim q_n}\left[z_n^2 \|\boldsymbol{v}\|_2^2 - 2 z_n \boldsymbol{x}_n^\top \boldsymbol{v}\right]. \quad \text{(1 point)}$$

Finally, we use the facts

$$\mathbb{E}_{z_n \sim q_n}[z_n] = a_n, \quad \mathbb{E}_{z_n \sim q_n}[z_n^2] = b^2 + (\mathbb{E}_{z_n \sim q_n}[z_n])^2 = b^2 + a_n^2, \quad \text{(1 point)}$$

to arrive at

$$-\frac{1}{2\sigma^2} \sum_{n=1}^{N} \left((b^2 + a_n^2)\|\boldsymbol{v}\|_2^2 - 2 a_n \boldsymbol{x}_n^\top \boldsymbol{v}\right) = -\frac{1}{2\sigma^2} \left(\left(\sum_{n=1}^{N} (b^2 + a_n^2)\right) \|\boldsymbol{v}\|_2^2 - 2 \left(\sum_{n=1}^{N} a_n \boldsymbol{x}_n\right)^\top \boldsymbol{v}\right). \quad \text{(1 point)}$$

Rubrics:

- The breakdown of points is only for reference.
- Having some $\boldsymbol{v}$-independent terms is fine.
- It is okay to even drop the factor $\frac{1}{2\sigma^2}$, but deduct 0.5 point if the negative sign is dropped as well.

2.3 M-step: Find the maximizer of $Q(\boldsymbol{v})$ by setting its gradient to $\mathbf{0}$. Express it in terms of $\boldsymbol{x}_n, a_n$, and b^2 (for $n = 1, \ldots, N$). (3 points)

Simply setting the gradient to zero gives

$$2\left(\sum_{n=1}^{N}(b^2 + a_n^2)\right)\boldsymbol{v} - 2\left(\sum_{n=1}^{N} a_n \boldsymbol{x}_n\right) = \mathbf{0}.$$

Solving for $\boldsymbol{v}$ gives

$$\boldsymbol{v} = \frac{\sum_{n=1}^{N} a_n \boldsymbol{x}_n}{\sum_{n=1}^{N}(b^2 + a_n^2)}.$$

Rubrics: 2 points for the correct gradient and 1 point for the correct final answer.

Final note: in case you are wondering why this model is useful, it is in fact a greatly simplified version of something called "probabilistic PCA".

Problem 3 Principal Component Analysis (18 points)

In the class we showed that PCA is finding the directions with the most variance. In this problem, you will show that PCA is in fact also minimizing reconstruction error in some sense.

3.1 Specifically, suppose we have a dataset $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^D$ with zero mean, and we would like to compress it into a one-dimensional dataset $c_1, \ldots, c_N \in \mathbb{R}$. To reconstruct the dataset (approximately), we also keep a direction vector $\mathbf{v} \in \mathbb{R}^D$ with unit norm (i.e. $\|\mathbf{v}\|_2 = 1$) so that the reconstructed dataset is $c_1\mathbf{v}, \ldots, c_N\mathbf{v} \in \mathbb{R}^D$.

The way we find $c_1, \ldots, c_N$ and $\mathbf{v}$ is to minimize the reconstruction error in terms of the squared L2 distance, that is, we solve

$$\arg\min_{c_1,\ldots,c_N,\mathbf{v}:\|\mathbf{v}\|_2=1} \sum_{n=1}^{N} \|\mathbf{x}_n - c_n\mathbf{v}\|_2^2. \tag{1}$$

Prove that the solution of (1) is exactly the following

1) $c_n = \mathbf{x}_n^T\mathbf{v}$ for each $n = 1, \ldots, N$; (3 points)

2) $\mathbf{v}$ is the first principal component of the dataset, that is, $\arg\max_{\mathbf{v}:\|\mathbf{v}\|_2=1} \mathbf{v}^T \left(\sum_{n=1}^N \mathbf{x}_n\mathbf{x}_n^T\right)\mathbf{v}$. (3 points)

Hint: first prove 1) by fixing $\mathbf{v}$, then prove 2) using the conclusion of 1).

For any fixed $\mathbf{v}$ (with unit norm), clearly we can optimize over each c_n independently: (1 point)

$$\begin{aligned}
\arg\min_{c_n} \|\mathbf{x}_n - c_n\mathbf{v}\|_2^2 &= \arg\min_{c_n} \left(c_n^2\|\mathbf{v}\|_2^2 - (2\mathbf{x}_n^T\mathbf{v})c_n + \|\mathbf{x}_n\|_2^2\right) \\
&= \arg\min_{c_n} \left(c_n^2 - (2\mathbf{x}_n^T\mathbf{v})c_n\right) && (\|\mathbf{v}\|_2^2 = 1) \\
&= \arg\min_{c_n} \left(c_n - \mathbf{x}_n^T\mathbf{v}\right)^2 && \text{(1 point)} \\
&= \mathbf{x}_n^T\mathbf{v}. && \text{(1 point)}
\end{aligned}$$

Next, plugging $c_n = \mathbf{x}_n^T\mathbf{v}$ back into the objective, we see that $\mathbf{v}$ is the solution of

$$\begin{aligned}
\arg\min_{\mathbf{v}:\|\mathbf{v}\|_2=1} \sum_{n=1}^{N} \|\mathbf{x}_n - (\mathbf{x}_n^T\mathbf{v})\mathbf{v}\|_2^2 &= \arg\min_{\mathbf{v}:\|\mathbf{v}\|_2=1} \sum_{n=1}^{N} \left(\|\mathbf{x}_n\|_2^2 - 2(\mathbf{x}_n^T\mathbf{v})^2 + (\mathbf{x}_n^T\mathbf{v})^2\|\mathbf{v}\|_2^2\right) && \text{(1 point)} \\
&= \arg\min_{\mathbf{v}:\|\mathbf{v}\|_2=1} \sum_{n=1}^{N} \left(-(\mathbf{x}_n^T\mathbf{v})^2\right) && (\|\mathbf{x}_n\|_2^2 \text{ is irrelevant and } \|\mathbf{v}\|_2^2 = 1, \text{1 point}) \\
&= \arg\min_{\mathbf{v}:\|\mathbf{v}\|_2=1} -\sum_{n=1}^{N} \mathbf{v}^T\mathbf{x}_n\mathbf{x}_n^T\mathbf{v} \\
&= \arg\max_{\mathbf{v}:\|\mathbf{v}\|_2=1} \mathbf{v}^T\left(\sum_{n=1}^{N} \mathbf{x}_n\mathbf{x}_n^T\right)\mathbf{v}. && \text{(1 point)}
\end{aligned}$$

Rubrics: The breakdown of points is only for reference. One can of course also find the solution for c_n by setting the gradient to 0.

3.2 Next, you are asked to generalize the same idea to an arbitrary compression dimension $p < D$. Specifically, we would like to compress the same zero-mean dataset into a p-dimensional dataset $\mathbf{c}_1, \ldots, \mathbf{c}_N \in \mathbb{R}^p$. To reconstruct the dataset (approximately), we also keep p orthogonal direction vectors $\mathbf{v}_1, \ldots, \mathbf{v}_p \in \mathbb{R}^D$ with unit norm. For notational convenience, we stack these vectors together as a matrix $\mathbf{V} \in \mathbb{R}^{D \times p}$ whose j-th column is $\mathbf{v}_j$.

1) Write down the reconstructed dataset using $\mathbf{c}_1, \ldots, \mathbf{c}_N$ and $\mathbf{V}$ (note: this is a set of points in $\mathbb{R}^D$). Then write down the analogue of (1), that is, the optimization problem (with variables $\mathbf{c}_1, \ldots, \mathbf{c}_N$ and $\mathbf{V}$) that minimizes the reconstruction error in terms of the squared L2 distance. Make sure to include the correct constraints in this optimization problem. No reasoning needed. (3 points)

   The reconstructed dataset is $\mathbf{V}\mathbf{c}_1, \ldots, \mathbf{V}\mathbf{c}_N \in \mathbb{R}^D$. Thus the optimization problem is

$$\arg\min_{\substack{\mathbf{c}_1,\ldots,\mathbf{c}_N \in \mathbb{R}^p \\ \mathbf{V} \in \mathbb{R}^{D\times p}:\ \mathbf{V}^\top \mathbf{V} = \mathbf{I}}} \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{V}\mathbf{c}_n\|_2^2,$$

   where $\mathbf{I}$ is the $p \times p$ identity matrix.

   Rubrics:

   - One point for the correct form of the reconstructed dataset.
   - One point for the correct constraint of the optimization problem. It is okay to omit the dimension constraints, but the "orthonormal" constraint $\mathbf{V}^\top \mathbf{V} = \mathbf{I}$ is important to have (and there are of course different acceptable ways to express this constraint).
   - One point for the correct objective. Do not deduct points if the mistake is only from the incorrect form of the reconstructed dataset.

2) Find the optimal solution of $\mathbf{c}_1, \ldots, \mathbf{c}_N$ while fixing $\mathbf{V}$. (4 points)

   Similarly, the optimization can be done independently for each $\mathbf{c}_n$: (1 point)

$$\arg\min_{\mathbf{c}_n} \|\mathbf{x}_n - \mathbf{V}\mathbf{c}_n\|_2^2 = \arg\min_{\mathbf{c}_n} \left(\|\mathbf{x}_n\|_2^2 - 2\mathbf{x}_n^\top \mathbf{V}\mathbf{c}_n + \mathbf{c}_n^\top \mathbf{V}^\top \mathbf{V}\mathbf{c}_n \right) \quad \text{(1 point)}$$

$$= \arg\min_{\mathbf{c}_n} \left(\mathbf{c}_n^\top \mathbf{c}_n - 2\mathbf{x}_n^\top \mathbf{V}\mathbf{c}_n \right). \quad \text{(1 point)}$$

   For the last step, simply setting the gradient $2\mathbf{c}_n - 2\mathbf{V}^\top \mathbf{x}_n$ to $\mathbf{0}$ gives the solution $\mathbf{c}_n = \mathbf{V}^\top \mathbf{x}_n$. (1 point)

   Rubrics: The breakdown of points is only for reference.

3) Plug the solution of the previous question into the optimization problem and conclude that the optimal solution of $\mathbf{V}$ is (5 points)

$$\underset{\mathbf{V}\in\mathbb{R}^{D\times p}:\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}}{\arg\max} \sum_{j=1}^{p}\left(\mathbf{v}_j^\top\left(\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top\right)\mathbf{v}_j\right).$$

(This means that $\mathbf{v}_1,\ldots,\mathbf{v}_p$ are exactly the top p eigenvectors of $\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top$, that is, the top p principal components. You do not need to prove this fact though.)

After plugging $\mathbf{c}_n = \mathbf{V}^\top\mathbf{x}_n$, the problem becomes

$$\begin{aligned}
\underset{\mathbf{V}:\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}}{\arg\min} \sum_{n=1}^{N}\|\mathbf{x}_n-\mathbf{V}\mathbf{V}^\top\mathbf{x}_n\|_2^2 &= \underset{\mathbf{V}:\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}}{\arg\min} \sum_{n=1}^{N}\left(\|\mathbf{x}_n\|_2^2-2\mathbf{x}_n^\top\mathbf{V}\mathbf{V}^\top\mathbf{x}_n+\mathbf{x}_n^\top\mathbf{V}\mathbf{V}^\top\mathbf{V}\mathbf{V}^\top\mathbf{x}_n\right) && \text{(1 point)}\\
&= \underset{\mathbf{V}:\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}}{\arg\min} \sum_{n=1}^{N}\left(-2\mathbf{x}_n^\top\mathbf{V}\mathbf{V}^\top\mathbf{x}_n+\mathbf{x}_n^\top\mathbf{V}\mathbf{V}^\top\mathbf{x}_n\right) \\
& && (\|\mathbf{x}_n\|_2^2 \text{ is irrelevant and } \mathbf{V}^\top\mathbf{V}=\mathbf{I}, \text{1 point})\\
&= \underset{\mathbf{V}:\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}}{\arg\max} \sum_{n=1}^{N}\left(\mathbf{x}_n^\top\mathbf{V}\mathbf{V}^\top\mathbf{x}_n\right) && \text{(1 point)}\\
&= \underset{\mathbf{V}:\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}}{\arg\max} \sum_{n=1}^{N}\left(\mathbf{x}_n^\top\left(\sum_{j=1}^{p}\mathbf{v}_j\mathbf{v}_j^\top\right)\mathbf{x}_n\right) && \text{(1 point)}\\
&= \underset{\mathbf{V}:\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}}{\arg\max} \sum_{n=1}^{N}\sum_{j=1}^{p}\left(\mathbf{x}_n^\top\mathbf{v}_j\right)\left(\mathbf{v}_j^\top\mathbf{x}_n\right)\\
&= \underset{\mathbf{V}:\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}}{\arg\max} \sum_{n=1}^{N}\sum_{j=1}^{p}\left(\mathbf{v}_j^\top\mathbf{x}_n\right)\left(\mathbf{x}_n^\top\mathbf{v}_j\right)\\
&= \underset{\mathbf{V}:\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}}{\arg\max} \sum_{j=1}^{p}\left(\mathbf{v}_j^\top\left(\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top\right)\mathbf{v}_j\right). && \text{(1 point)}
\end{aligned}$$

Rubrics: The breakdown of points is only for reference.