Machine Learning Homework 5

May 7, 2025

2023141460251 Haoxiang Sun

Question 1: What are advantages of CCA over PCA?

1.1 Models Relationships Between Datasets

- **CCA:** Explicitly designed to find shared information or correlations between two distinct sets of variables (views). It seeks projections that maximize the correlation between these views.
- **PCA:** Operates on a single dataset, optimizing for variance within that dataset. It does not inherently model relationships between two separate datasets. If applied to concatenated datasets, it might not find the shared structure but rather the dominant variance.

1.2 Interpretation in Cross-Modal Settings

- **CCA:** The canonical variates derived by CCA directly represent the axes of maximal shared correlation. This is highly interpretable when trying to understand how two different views of an object or phenomenon relate to each other (e.g., image features and text features).
- **PCA:** Principal components explain variance within their own dataset, which may not align with or illuminate the shared structure between two different views.

1.3 Finding Correlated Subspaces

- **CCA:** Can discover underlying latent variables or subspaces that are common to both datasets, even if these subspaces do not explain the maximum variance within each individual dataset. It focuses on the covariance/correlation between sets.
- **PCA:** Focuses solely on variance within one set, potentially missing subtle but highly correlated signals if they are not high-variance signals.

1.4 Implicit Supervision for Feature Extraction

- **CCA:** While often termed unsupervised, CCA can be seen as "implicitly supervised" if one set of variables can be considered a target or outcome related to the other. It learns projections that make the two views maximally similar (correlated), which can be a form of guidance.
- **PCA:** Purely unsupervised, ignoring any external information or relationships between variable sets.

Question 2: What are the limitations/requirements of CCA?

2.1 Data Requirements

a) **Paired Data:** CCA requires that the samples across the two datasets (views) are paired. For each sample i, measurements must be available for both sets of variables, $\mathbf{x}_i$ and $\mathbf{y}_i$.

b) **Sufficient Samples:** CCA can be prone to overfitting if the number of samples (n) is not sufficiently larger than the number of features in each set (p and q). If $n < p + q$, CCA might find spurious correlations. Regularization can help mitigate this.

2.2 Assumptions

a) **Linearity:** Standard CCA assumes that the relationship between the two sets of variables is linear, as it seeks linear combinations of variables. If the true underlying relationship is non-linear, CCA may not capture it effectively (Kernel CCA can address this).

b) **Correlation as a Metric:** CCA optimizes for Pearson correlation. If other types of dependency (e.g., non-linear, higher-order) are more relevant, CCA might not be the optimal choice.

2.3 Practical/Mathematical Limitations

a) **Invertibility of Covariance Matrices:** The solution often involves inverting covariance matrices ($\mathbf{S}_{XX}, \mathbf{S}_{YY}$). If these matrices are singular (e.g., due to multicollinearity or $p > n$ or $q > n$) or ill-conditioned, regularization techniques (like ridge CCA) are necessary.

b) **Interpretation Difficulty:** The canonical variates are linear combinations of all original variables. If many variables have non-zero weights, interpreting precisely what these variates represent can be challenging.

c) **Sensitivity to Scaling:** CCA results can be sensitive to the scaling of the input variables. It is generally recommended to standardize variables (e.g., to zero mean and unit variance) before applying CCA.

d) **Number of Meaningful Components:** CCA finds $\min(p, q)$ pairs of canonical variates. Not all of these may represent statistically significant or practically meaningful correlations; often only the first few are considered important.

Question 3: The objective function of CCA, and how to solve it? (binary and multiple case)

3.1 Objective Function Formulation

Let $\mathbf{X}$ be an $n \times p$ matrix and $\mathbf{Y}$ be an $n \times q$ matrix, both centered. The covariance matrices are:

$$\mathbf{S}_{XX} = \frac{1}{n-1}\mathbf{X}^\top\mathbf{X}$$
$$\mathbf{S}_{YY} = \frac{1}{n-1}\mathbf{Y}^\top\mathbf{Y}$$
$$\mathbf{S}_{XY} = \frac{1}{n-1}\mathbf{X}^\top\mathbf{Y}$$

CCA seeks projection vectors $\mathbf{w}_x$ (for $\mathbf{X}$) and $\mathbf{w}_y$ (for $\mathbf{Y}$) such that the correlation between the projected variables $\mathbf{U} = \mathbf{X}\mathbf{w}_x$ and $\mathbf{V} = \mathbf{Y}\mathbf{w}_y$ is maximized. The correlation is:

$$\rho = \text{corr}(\mathbf{U}, \mathbf{V}) = \frac{\mathbf{w}_x^\top \mathbf{S}_{XY} \mathbf{w}_y}{\sqrt{(\mathbf{w}_x^\top \mathbf{S}_{XX} \mathbf{w}_x)(\mathbf{w}_y^\top \mathbf{S}_{YY} \mathbf{w}_y)}} \tag{1}$$

To make the maximization well-defined, constraints are imposed:

$$\mathbf{w}_x^\top \mathbf{S}_{XX} \mathbf{w}_x = 1 \tag{2}$$

$$\mathbf{w}_y^\top \mathbf{S}_{YY} \mathbf{w}_y = 1 \tag{3}$$

With these constraints, the objective simplifies to maximizing:

$$J(\mathbf{w}_x, \mathbf{w}_y) = \mathbf{w}_x^\top \mathbf{S}_{XY} \mathbf{w}_y \tag{4}$$

3.2 Solution via Lagrangian Multipliers

Using Lagrangian multipliers for the constrained optimization problem leads to the following system of equations:

$$\mathbf{S}_{XY} \mathbf{w}_y = \lambda \mathbf{S}_{XX} \mathbf{w}_x \tag{5}$$

$$\mathbf{S}_{YX} \mathbf{w}_x = \mu \mathbf{S}_{YY} \mathbf{w}_y \tag{6}$$

It can be shown that $\lambda = \mu = \rho$, the maximized correlation. Substituting one into the other (assuming invertibility of $\mathbf{S}_{XX}$ and $\mathbf{S}_{YY}$) leads to generalized eigenvalue problems. For instance:

$$\mathbf{S}_{XX}^{-1} \mathbf{S}_{XY} \mathbf{S}_{YY}^{-1} \mathbf{S}_{YX} \mathbf{w}_x = \rho^2 \mathbf{w}_x \tag{7}$$

And symmetrically:

$$\mathbf{S}_{YY}^{-1} \mathbf{S}_{YX} \mathbf{S}_{XX}^{-1} \mathbf{S}_{XY} \mathbf{w}_y = \rho^2 \mathbf{w}_y \tag{8}$$

The eigenvalues ρ^2 are the squared canonical correlations, and the eigenvectors $\mathbf{w}_x, \mathbf{w}_y$ are the canonical weight vectors.

3.3 Binary vs. Multiple Case

- **Binary (First Pair):** The above solution finds the first pair of canonical variates $(\mathbf{w}_{x1}, \mathbf{w}_{y1})$ corresponding to the largest canonical correlation ρ_1. This involves finding the eigenvector corresponding to the largest eigenvalue ρ_1^2.
- **Multiple Case (Subsequent Pairs):** To find the k-th pair of canonical variates, we solve the same eigenvalue problem. The solutions yield multiple eigenvectors and eigenvalues. These are ordered by decreasing eigenvalues $\rho_1^2 \geq \rho_2^2 \geq \cdots \geq \rho_m^2$, where $m = \min(p, q)$. The corresponding eigenvectors provide the successive canonical weight vectors $(\mathbf{w}_{xk}, \mathbf{w}_{yk})$. These subsequent variates $U_k = \mathbf{X}\mathbf{w}_{xk}$ and $V_k = \mathbf{Y}\mathbf{w}_{yk}$ are constructed to be uncorrelated with all previous variates U_j, V_j for $j < k$ (within their own set and across sets for $j \neq k$).

Question 4: Implement CCA

4.1 Conceptual Implementation Steps

Implementing CCA involves the following key steps:

a) **Input Data:** Obtain two data matrices, $\mathbf{X}$ $(n \times p)$ and $\mathbf{Y}$ $(n \times q)$, with n paired samples.

b) **Preprocessing:**

- Center the data: Subtract the mean from each column of $\mathbf{X}$ and $\mathbf{Y}$.
- (Recommended) Standardize the data: Divide each column by its standard deviation, especially if features have different scales.

c) **Compute Covariance Matrices:** Calculate $\mathbf{S}_{XX}$, $\mathbf{S}_{YY}$, and $\mathbf{S}_{XY}$ (and $\mathbf{S}_{YX} = \mathbf{S}_{XY}^\top$) from the preprocessed data. For example, $\mathbf{S}_{XX} = \frac{1}{n-1}\mathbf{X}^\top\mathbf{X}$.

d) **Solve the Generalized Eigenvalue Problem:** This is the core step. One common approach is to solve for $\mathbf{w}_x$ from equation (7):

$$(\mathbf{S}_{XX}^{-1}\mathbf{S}_{XY}\mathbf{S}_{YY}^{-1}\mathbf{S}_{YX})\mathbf{w}_x = \rho^2\mathbf{w}_x$$

Numerically stable methods often involve SVD of a transformed matrix, e.g., $\mathbf{K} = \mathbf{S}_{XX}^{-1/2}\mathbf{S}_{XY}\mathbf{S}_{YY}^{-1/2}$. The singular values of $\mathbf{K}$ are the canonical correlations ρ_k. If $\mathbf{S}_{XX}$ or $\mathbf{S}_{YY}$ are singular or ill-conditioned, pseudo-inverses or regularization (e.g., adding a small multiple of the identity matrix, $\alpha\mathbf{I}$, to $\mathbf{S}_{XX}$ and $\mathbf{S}_{YY}$) should be used.

e) **Calculate Canonical Weight Vectors for the Second View:** Once $\mathbf{w}_x$ vectors are found, the corresponding $\mathbf{w}_y$ vectors can be calculated using a rearranged form of equation (5) or (6), for example:

$$\mathbf{w}_y = \frac{1}{\rho}\mathbf{S}_{YY}^{-1}\mathbf{S}_{YX}\mathbf{w}_x$$

(Ensuring $\rho \neq 0$).

f) **Sort and Select Components:** Sort the canonical correlations ρ_k (square roots of eigenvalues ρ_k^2) in descending order. Sort the corresponding $\mathbf{w}_x$ and $\mathbf{w}_y$ vectors (columns) accordingly. One may choose to keep only the top k' components with significant correlations.

g) **Normalize Canonical Weight Vectors (Optional but good practice):** Ensure that the canonical weight vectors satisfy the unit variance constraints, e.g., $\mathbf{w}_{xk}^\top\mathbf{S}_{XX}\mathbf{w}_{xk} = 1$. This is often handled by how eigenvectors are normalized, or may require an explicit scaling step.

h) **Compute Canonical Variates (Projections):** Transform the original (centered) data using the calculated weight vectors:

$$\mathbf{U} = \mathbf{X}_{\text{centered}}\mathbf{W}_x$$
$$\mathbf{V} = \mathbf{Y}_{\text{centered}}\mathbf{W}_y$$

where $\mathbf{W}_x$ and $\mathbf{W}_y$ are matrices whose columns are the canonical weight vectors $\mathbf{w}_{xk}$ and $\mathbf{w}_{yk}$.

Libraries like Scikit-learn in Python provide robust implementations ('sklearn.cross$_d$*ecomposition.CCA*').

Question 5: What is the major difference between LDA and PCA, give two at least.

5.1 Supervision (Use of Labels)

- **PCA (Principal Component Analysis):** Is an **unsupervised** dimensionality reduction technique. It does not use class labels. PCA finds directions (principal components) that maximize the variance in the data, irrespective of any class structure.

- **LDA (Linear Discriminant Analysis):** Is a **supervised** dimensionality reduction technique (also used for classification). It explicitly uses class labels to find a subspace that maximizes the separability between classes.

5.2 Objective/Goal

- **PCA:** Aims to maximize the variance of the projected data. The components are ordered by the amount of variance they explain. The objective is to find projection $\mathbf{w}$ that maximizes $\mathbf{w}^\top \mathbf{S}_T \mathbf{w}$, where $\mathbf{S}_T$ is the total scatter/covariance matrix.
- **LDA:** Aims to maximize the ratio of between-class scatter to within-class scatter. It seeks projections where classes are far apart from each other, and each class is compact. The objective is to find projection $\mathbf{w}$ that maximizes $\frac{\mathbf{w}^\top \mathbf{S}_B \mathbf{w}}{\mathbf{w}^\top \mathbf{S}_W \mathbf{w}}$, where $\mathbf{S}_B$ is the between-class scatter and $\mathbf{S}_W$ is the within-class scatter.

5.3 Dimensionality of Output

- **PCA:** Can project data onto d' dimensions, where d' can be any number up to the original number of features p (or $n-1$ if $n < p$).
- **LDA:** Reduces dimensionality to at most $K-1$ dimensions, where K is the number of classes.

5.4 Application Focus

- **PCA:** General purpose dimensionality reduction, data compression, noise reduction, visualization.
- **LDA:** Primarily for classification-oriented dimensionality reduction or as a classifier itself. It is effective when class separability is the main concern.

Question 6: What is the key idea of LDA and how LDA utilize the label to perform DR?

6.1 Key Idea of LDA

The key idea of Linear Discriminant Analysis (LDA) is to find a lower-dimensional subspace onto which the data can be projected such that the **ratio of between-class variance (or scatter) to within-class variance (or scatter) is maximized.** In simpler terms, LDA aims to:

- Make the means of different classes as far apart as possible (maximize between-class scatter).
- Make the data points within each class as close to their class mean as possible (minimize within-class scatter).

This results in a projection that emphasizes class separability.

6.2 How Labels are Utilized for Dimensionality Reduction (DR)

Class labels are fundamental to LDA and are used in the following ways to perform DR:

a) **Calculating Class Means (μ_k):** Labels are used to group data points into their respective classes (C_k for class k). For each class k, the mean vector μ_k is computed using only the samples belonging to that class:

$$\mu_k = \frac{1}{N_k} \sum_{\mathbf{x}_i \in C_k} \mathbf{x}_i$$

where N_k is the number of samples in class k.

b) **Calculating Overall Mean (μ):** The overall mean of all data points is computed: $\mu = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$.

c) **Calculating Within-Class Scatter Matrix ($\mathbf{S}_W$):** Labels determine which samples contribute to the scatter calculation for each class. $\mathbf{S}_W$ measures the compactness of each class.

$$\mathbf{S}_W = \sum_{k=1}^{K} \mathbf{S}_k = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top$$

where K is the number of classes.

d) **Calculating Between-Class Scatter Matrix ($\mathbf{S}_B$):** Labels (via class means μ_k and class sizes N_k) are used to measure how far apart the class means are from the overall mean. $\mathbf{S}_B$ measures the separation between class means.

$$\mathbf{S}_B = \sum_{k=1}^{K} N_k (\mu_k - \mu)(\mu_k - \mu)^\top$$

e) **Finding the Optimal Projection:** LDA seeks projection vectors $\mathbf{w}$ (forming columns of a projection matrix $\mathbf{W}$) that maximize the Fisher criterion (see Q7). The labels, by defining $\mathbf{S}_B$ and $\mathbf{S}_W$, directly guide the determination of these optimal projection directions. The resulting projected features are designed to be maximally discriminative for the given classes.

Question 7: The objective function of LDA, and how to solve it? (binary and multiple case)

7.1 Objective Function (Fisher's Criterion)

LDA aims to find a projection vector $\mathbf{w}$ (or a matrix $\mathbf{W}$ for multiple dimensions) that maximizes the ratio of the between-class scatter to the within-class scatter in the projected space. For a single projection vector $\mathbf{w}$ (reducing to 1 dimension), this is Fisher's criterion:

$$J(\mathbf{w}) = \frac{\text{between-class scatter of projected data}}{\text{within-class scatter of projected data}} = \frac{\mathbf{w}^\top \mathbf{S}_B \mathbf{w}}{\mathbf{w}^\top \mathbf{S}_W \mathbf{w}} \tag{1}$$

where:

- $\mathbf{S}_B = \sum_{k=1}^{K} N_k (\mu_k - \mu)(\mu_k - \mu)^\top$ is the between-class scatter matrix.
- $\mathbf{S}_W = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top$ is the within-class scatter matrix.

7.2 Solution Method

To maximize $J(\mathbf{w})$, we take the derivative with respect to $\mathbf{w}$ and set it to zero. This leads to the generalized eigenvalue problem:

$$\mathbf{S}_B\mathbf{w} = \lambda\mathbf{S}_W\mathbf{w} \tag{2}$$

If $\mathbf{S}_W$ is invertible (which is usually the case if $N > p$ and features are not perfectly collinear within classes), this can be rewritten as a standard eigenvalue problem:

$$\mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{w} = \lambda\mathbf{w} \tag{3}$$

The solutions $\mathbf{w}$ are the eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$, and λ are the corresponding eigenvalues. The eigenvectors corresponding to the largest eigenvalues are the desired projection directions (linear discriminants). These form the columns of the projection matrix $\mathbf{W}$.

7.3 Binary Case (K=2 classes)

- The class means are μ_1 and μ_2.
- The within-class scatter matrix is $\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$, where $\mathbf{S}_k = \sum_{\mathbf{x}_i \in C_k} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top$.
- The between-class scatter matrix $\mathbf{S}_B$ simplifies (up to a scaling factor that doesn't affect $\mathbf{w}$'s direction) to:
$$\mathbf{S}_B \propto (\mu_1 - \mu_2)(\mu_1 - \mu_2)^\top$$
- The solution for the optimal projection vector $\mathbf{w}$ is proportional to:
$$\mathbf{w} \propto \mathbf{S}_W^{-1}(\mu_1 - \mu_2)$$
In the binary case, there is only one such discriminant direction (or its negative). Thus, LDA reduces the dimensionality to 1.

7.4 Multiple Case (K ¿ 2 classes)

- The within-class scatter matrix $\mathbf{S}_W$ and between-class scatter matrix $\mathbf{S}_B$ are calculated as defined in section 7.1.
- We solve the generalized eigenvalue problem $\mathbf{S}_B\mathbf{w} = \lambda\mathbf{S}_W\mathbf{w}$, or equivalently equation (3).
- There will be at most $K - 1$ non-zero eigenvalues and corresponding eigenvectors (see Question 8).
- The eigenvectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{K-1}$ corresponding to the $K - 1$ largest eigenvalues are chosen as the basis vectors for the new lower-dimensional space. These form the columns of the projection matrix $\mathbf{W}$.
- The data is then projected as $\mathbf{Z} = \mathbf{X}\mathbf{W}$.

Question 8: Why LDA could reduce the data into a K-1 dimensional space at most?

The reason LDA can reduce data to at most $K - 1$ dimensions, where K is the number of classes, lies in the rank of the between-class scatter matrix $\mathbf{S}_B$.

8.1 Definition and Rank of $\mathbf{S}_B$

The between-class scatter matrix $\mathbf{S}_B$ is defined as:

$$\mathbf{S}_B = \sum_{k=1}^{K} N_k(\mu_k - \mu)(\mu_k - \mu)^\top \tag{1}$$

where μ_k is the mean of class k, μ is the overall mean of all data, and N_k is the number of samples in class k.

Each term $(\mu_k - \mu)(\mu_k - \mu)^\top$ is an outer product of a vector $(\mu_k - \mu)$ with itself. If $(\mu_k - \mu)$ is a non-zero vector, this outer product results in a matrix of rank 1. $\mathbf{S}_B$ is a sum of K such (at most) rank-1 matrices.

8.2 Linear Dependence of $(\mu_k - \mu)$ Vectors

The K vectors $(\mu_k - \mu)$ are not linearly independent. The overall mean μ can be expressed as a weighted average of the class means:

$$\mu = \frac{1}{N} \sum_{k=1}^{K} N_k \mu_k \tag{2}$$

where $N = \sum N_k$ is the total number of samples. Therefore, the weighted sum of these difference vectors is zero:

$$\sum_{k=1}^{K} N_k(\mu_k - \mu) = \sum_{k=1}^{K} N_k \mu_k - \sum_{k=1}^{K} N_k \mu = N\mu - N\mu = \mathbf{0} \tag{3}$$

This linear dependency implies that the K vectors $(\mu_k - \mu)$ span a subspace of dimension at most $K - 1$. (Imagine K points μ_k in the original feature space; their means define an affine subspace. The vectors pointing from the global mean μ to each μ_k live in a vector subspace of dimension at most $K - 1$.)

8.3 Rank of $\mathbf{S}_B$ and Eigenvalues

Since the vectors $(\mu_k - \mu)$ span a space of at most $K - 1$ dimensions, the sum of their outer products, $\mathbf{S}_B$, will have a rank of at most $K - 1$.

$$\text{rank}(\mathbf{S}_B) \leq K - 1 \tag{4}$$

LDA finds its projection vectors by solving the generalized eigenvalue problem $\mathbf{S}_B\mathbf{w} = \lambda\mathbf{S}_W\mathbf{w}$, or $\mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{w} = \lambda\mathbf{w}$ (assuming $\mathbf{S}_W$ is invertible). The number of non-zero eigenvalues λ is limited by the rank of the matrix $\mathbf{S}_W^{-1}\mathbf{S}_B$. If $\mathbf{S}_W$ is full rank (invertible), then $\text{rank}(\mathbf{S}_W^{-1}\mathbf{S}_B) = \text{rank}(\mathbf{S}_B)$. Thus, there are at most $K - 1$ non-zero eigenvalues.

8.4 Conclusion

Each non-zero eigenvalue corresponds to a distinct discriminant direction (an eigenvector $\mathbf{w}$). Since there are at most $K - 1$ non-zero eigenvalues, there are at most $K - 1$ useful discriminant directions that capture the between-class variance. Therefore, LDA can reduce the dimensionality of the data to a $(K - 1)$-dimensional subspace at most. If the original dimensionality p is less than $K - 1$, then LDA can reduce to at most p dimensions. So, more precisely, the dimensionality of the projected space is $\min(p, K - 1)$.