Laplacian Multiset Canonical Correlation Analysis

(LapMCCA)

Monali Koshta (0801CS171044)

December 20, 2020

Contents

Chapter 1

Introduction

1.1 Multiview Learning

Multi-view learning is an emerging direction in machine learning which considers learning with multiple views to improve the generalization performance. Multi-view learning is also known as data fusion or data integration from multiple feature sets.

1.2 MCCA

MCCA (Multi-set Canonical Correlation Analysis) is a powerful technique for analyzing linear relationships between more (than two) sets of random variables. Up to now, researchers have developed many models for MCCA based on different criterion functions and constraints. The following form is regarded as a natural and direct extension of CCA, which has been applied to multiview learning.
Specifically, given m zero-mean random vectors $\{x_i \in \Re^{di}\}$ for i=1,2,…m , a set of linear projection directions $\{\alpha_i \in \Re^{di}\}$ for i=1,2,…m, is found by maximizing the sum of pair-wise correlations between the projected variables $\{\alpha_i^T x_i\}$ for i=1,2,…m, as follows:

$$\rho(\tilde{\alpha}) = \sum_{i=1}^{m} \sum_{i=1}^{m} \alpha_i^T S_{ij} \alpha_j \qquad (1)$$

where $\tilde{\alpha}^T = (\alpha_1^T, \alpha_2^T, \ldots, \alpha_m^T)$, S_{ii} is the within-set covariance matrix of vector x_i, and $S_{ij}(i \neq j)$ is the between-set covariance matrix between vectors x_i and x_j.

1.3 LapMCCA

LapMCCA algorithm assumes that sample points are linearly correlated within local neighborhood, which is intuitively reasonable. With the assumption, the globally nonlinear problem from multiple data spaces can naturally be decomposed into multiple locally linear sub-problems. Thus, the LapMCCA algorithm can discover the nonlinear correlation information of the input spaces by combining these locally linear sub-problems together.

Chapter 2

Mathematical Formulation

2.1 Formulation

Since our LapMCCA algorithm is fundamentally based on the locality idea, we first construct the nearest neighbor affinity graph for each view. Then, we build the local within-view and between-view correlations for the LapMCCA model using the equivalent description of MCCA.

2.1.1 The nearest neighbor affinity graph

For n sample points {xi,1,xi,2,...,xi,n } from the ith view Xi, we build a p-nearest neighbor graph Gi with n vertices, where each vertex corresponds to a sample point xi,j , i=1,2,⋯,m, j=1,2,…,n. For each point xi,j , we find its p nearest neighbors in the same class and put edges between xi,j and its neighbors. Currently, there are many choices to define the weight matrix Wi=(wjk)∈ℜn×n on graph Gi. Four of the most widely used are as follows:

- **0–1 Weighting:** w_{jk}^{i}=1 if there is an edge between sample points x_j^i and x_k^i, and w_{jk}^{i}=0 otherwise.
- **Heat Kernel Weighting:** If there is an edge between sample points x_j^i and x_k^i, $w_{jk}^{i}= e^{-\| x_j^i - x_k^i \|^2 / 2\sigma^2}$, and w_{jk}^{i}= 0 otherwise.
- **Cosine Similarity Weighting**: If there is an edge between sample points x_j^i and x_k^i, $w_{jk}^{i} = \frac{x_j^T x_k}{||x_j||.||x_k||}$, and w_{jk}^{i}=0 otherwise.
- **Dot-Product Weighting**: If there is an edge between sample points x_j^i and x_k^i, $w_{jk}^{i} = x_j^T x_k$, and w_{jk}^{i}=0 otherwise.

Here we will use cosine similarity weighting.

2.1.2 Characterize the local within-view correlation

The local within-view correlation can be characterized from the p-nearest neighbor graph G(i) with the term

$$\rho_{ii}^{L} = \alpha_i^T S_{ii}^{L} \alpha_i \qquad (2)$$

where S_{ii}^{L} is referred to as local within-view covariance matrix and

$$S_{ii}^{L} = \frac{1}{n^2} X^i L^i X^{iT} \tag{3}$$

where $L^i = D^i - W^i$ is called the Laplacian matrix and D^i = diag($d_{11}^i, d_{22}^i, \ldots, d_{nn}^i$) where $d_{jj}^i = \sum_{k=1}^{n} w_{jk}^i$

2.1.3 Characterize the local between-view correlation

Similarly, the local within-view correlation can be characterized from the p-nearest neighbor graph G(i) with the term

$$\rho_{ij}^{L} = \alpha_i^T S_{ij}^L \alpha_j \tag{4}$$

where S_{ij}^L is referred to as local between-view covariance matrix and

$$S_{ij}^{L} = \frac{1}{n^2} X^i L^{ij} X^{jT} \tag{5}$$

where $L^{ij} = D^{ij} - W^{ij}$ is called the Laplacian matrix and D^{ij} = diag($d_{11}^{ij}, d_{22}^{ij}, \ldots, d_{nn}^{ij}$) where $d_{kk}^{ij} = \sum_{k=1}^{n} w_{kt}^{ij} = \sum_{k=1}^{n} w_{kt}^i w_{kt}^j$ and $W^{ij} = W^i \circ W^j$ and the operator ° is defined as $(W^i \circ W^j)_{kt} = w_{kt}^i w_{kt}^j$

2.1.4 Model and Solution of LapMCC

With the step 2 and 3 we can now construct LapMCC model as follows:

$$maxJ(\tilde{\alpha}) = \sum_{i=1}^{m} \sum_{i=1}^{m} \alpha_i^T S_{ij}^L \alpha_j \tag{6}$$

$$\text{s.t. } \alpha_i^T S_{ii}^L \alpha_i = 1, \text{ i=1,2,3…m}$$

Now with the Lagrange multiplier technique, the optimization model can be equivalently transformed into the following Lagrangian function:

$$\boldsymbol{F}(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\lambda}) = \sum_{i=1}^{m} \sum_{i=1}^{m} \alpha_i^T S_{ij}^L \alpha_j - \lambda \left(\sum_{i=1}^{m} \alpha_i^T S_{ii}^L \alpha_i - 1 \right) \tag{7}$$

With λ as the Lagrange multiplier. Let ∂F/∂αi=0, then we have:

$$\frac{\partial F}{\partial \alpha_i} = 2 \sum_{j=1}^{m} S_{ij}^L \alpha_j - 2\lambda S_{ii}^L \alpha_i = 0 \text{ , i=1,2,…,m.} \tag{8}$$

After some operation, we get m equations as:

$$S^L \tilde{\alpha} = \lambda S_D^L \tilde{\alpha} \tag{9}$$

where $S^L \in \Re d \times d$ is a block matrix with (i, j)th block-element as S_{ij}^L, and $S_D^L = \text{diag}(S_{11}^L, S_{22}^L, \dots, S_{mm}^L) \in \Re d \times d$, $d = \sum_{i=1}^{m} d_i$ and hence we get generalized eigenvalue problem.

Now if S_D^L is singular matrix we regularize it as:

$$S_D^L \leftarrow S_D^L + \delta I \tag{10}$$

Where I is Identity matrix and δ is a small positive number and chosen as 0.001.

If not then, projection directions can be selected as the generalized eigenvectors $\{\tilde{\alpha}_k^T = (\alpha_{1k}^T, \alpha_{2k}^T, \dots, \alpha_{mk}^T)\}_{k=1}^{h}$ corresponding to the top h eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_h$.

2.1.5 Horst Algorithm

Existing study has shown that MEP is very difficult and has no analytical solutions (i.e., exact solutions) in the m>2 case. In other words, we are only able to obtain approximate solutions of the problem by computing MEP and therefore, we used Horst Algorithm to solve MEP as in our case m>2.

Algorithm:

Input: Covariance matrices Sij, initial vectors α_0^i where i,j=1,2,…,m
Output: $\alpha_1^i, \dots, \alpha_{maxiter}^i$
for i=1 to maxiter do:
  for j=1 to m do:
    $\alpha_j^i \leftarrow \sum_{k=1}^{m} S_{j,k} \alpha_k^{i-1}$
    $\alpha_j^i \leftarrow \frac{\alpha_j^i}{\sqrt{\alpha_j^{i\prime} \alpha_j^i}}$
  end for
end for

After obtaining h sets of projection directions, m projection matrices for m views, $\{P_i = (\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{ih})\}_{i=1}^{m}$ can be formed to perform MDR by the form of $P_1^T X^1, P_2^T X^2, \dots, P_m^T X^m$ for classification tasks.

Note:

Since the algorithm needs, initial vectors α_0^i where i,j=1,2,…,m as inputs which we don't have in the LapMCCA hence generating it randomly.

Chapter 3

Algorithm

3.1 LapMCCA

Input:

1. m different view from the same n objects are given as X=$\{X^i\}_{i=1}^m$ where X^i=$(x_1^i, x_2^i, \dots, x_n^i)$ with $x_1^i \in R^{di}$ data matrix of the ith view containing d_i dimensional sample vectors in its column
2. Vector set given by $\{x_j^i\}_{i=1}^m$ containing labels from 1,2,…,c for the c classes and j=1,2,…,n

Output:

New m projection vectors.

Step 1: Construct p-nearest neighbor graphs:

In each view, for any two sample node in the same class compute the weight matrix for the graph using cosine similarity weighting.

Step 2: Compute local variance matrices:

Compute local variance matrices with the formula mentioned in 2.1.2 and 2.1.3

Step 3: Solving the generalized eigen-equation:

Compute a set of top h eigen vectors corresponding to the top h eigen values to form m projection matrices (computed with horst algorithm).

Step 4: Projecting Samples:

For a given multiview sample $(x^T = x^{(1)T}, x^{(2)T}, \dots, x^{(m)T})$ with $x^i \in R^{di}$, perform MDR by $P_1^T x^1, P_2^T x^2, \dots, P_m^T x^m$.

Step 5: Fusing low dimensional representation:

Use the fusion strategy given below to integrate m h-dimensional feature vectors of multiview sample x for recognition.

$$y = \sum_{i=1}^{m} P_i^T x^i = P^T x \tag{11}$$

where $P^T = (P_1^T, P_2^T, \ldots, P_m^T)$. The low-dimensional representation y is used to represent the multiview sample x for classification tasks.

Chapter 4

Documentation of API

4.1 Package Organization

LapMCCA

class LapMCCA(n_components = 2, p_neighbors = 3, reg=0.001)

Parameters:

- n_components: int, (default=2)
  number of components to keep.

- p_neighbors: int, (default=3)
  number of nearest neighbor in the graph of the same class.

- reg: int, float, (default=0.001)
  regularization parameter to be multiplied with identity matrix and then the product to be added to with-view covariance matrix.

Attributes:

m -> number of views
d -> number of samples in that view
n -> number of dimensions in each view

- W: array,[m,m]
  Weight matrix for the p nearest neighbour graph.

- D_w_view: array,[m,m]
  Diagonal matrix for the created from W.

- L_w_view: array,[m,m]
  Matrix given by L = D - W.

- S_w_view: array,[m,m]
  Within-view covariance matrix.

- Wij: array,[m,m]
  Weight matrix for the i and j view.

- Dij: array,[m,m]

Diagonal matrix for the created from Wij

- Lij: array,[m,m]
  Matrix given by Lij = Dij – Wij

- Sij: array,[m,m]
  Between-view covariance matrix

4.2 Methods

init (n_components = 2, p_neighbors = 3, reg=0.001)

initialize self

1. **fit(X,label)**

   fit model to data

   Parameter:

- X: array, [m,m]

  Contain m number of training vectors

- label: array, [m,n]

  Contains label for the view to create p neighbour affinity graph

  Returns:

- alpha: array, [m,m]

  which is the m projection direction with h coloums.

2. **fit_transform(X,label)**

   fit model to data and transform the training vectors.

   Parameter:

- X: array, [m,m]

  Contain m number of training vectors

- label: array, [m,n]

  Contains label for the view to create p neighbour affinity graph

  Returns:

- Xnew: array, [m,m]

  which is the transformed training vectors.

Note:

The X given as input to above methods contains m view vectors where the number of sample (row) in each view can vary but the number of dimensions i.e. n (coloum) needs to be same for every view.
And the label contain m rows for each view and n number of labels in its coloum each for a dimension i.e. n.

Chapter 5

Example

5.1 Example 1

```
lapmcca=LapMCCA(3,2)
a = np.array([[1, 4, 3,3], [25, 1,4,40], [33, 24,23,34], [1, 1,45,23], [23, 1,1,34], [3, 24,34,25]])
b = np.array([[-1, -1,2,3], [-2,2, -1,-3], [-3,4, -2,0], [1,2, 1,8], [2,3, 1,5], [3, 4,2,3],[0,1, 0,7]])
c = np.array([[1, 1,3,2], [2, 3,1,5], [3, 4,2,7], [1, 1,3,1], [2, 1,3,4], [3,4, 2,6],[0,2, 0,0],
                [2,4,7,90]])
X=np.array([a,b,c])
label=np.array([[1,1,2,2], [0,0,1,1], [0,1,1,0]])
alpha=lapmcca.fit(X,label)
```

5.2 Example 2

```
lapmcca=LapMCCA(3,2)
a = np.array([[1, 4, 3,3], [25, 1,4,40], [33, 24,23,34], [1, 1,45,23], [23, 1,1,34], [3, 24,34,25]])
b = np.array([[-1, -1,2,3], [-2,2, -1,-3], [-3,4, -2,0], [1,2, 1,8], [2,3, 1,5], [3, 4,2,3],[0,1, 0,7]])
c = np.array([[1, 1,3,2], [2, 3,1,5], [3, 4,2,7], [1, 1,3,1], [2, 1,3,4], [3,4, 2,6],[0,2, 0,0],
                [2,4,7,90]])
X=np.array([a,b,c])
label=np.array([[1,1,2,2], [0,0,1,1], [0,1,1,0]])
Xnew=lapmcca.fit_transform(X,label)
print(Xnew)
```

Chapter 6

Learning Outcomes

1. Learnt the concept of multiview and how multiview data looks like and the concept of Canonical Correlation Analysis (CCA) and Laplacian MCCA.

2. Ability to analyse and handle multiview data and apply the concepts of Laplacian MCCA which involves various steps to train the model.

3. Learnt the Horst Algorithm to solve the Multivariate Eigenvalue Problem for m > 2.

References

1. Yun-Hao Yuan1,2 & Yun Li 1 & Xiao-Bo Shen3,4 & Quan-Sen Sun3 & Jin-Long Yang2 Laplacian multiset canonical correlations for Multiview feature extraction and image recognition.

2. Jan Rupnik (1), John Shawe-Taylor (2) - Multi-View Canonical Correlation Analysis.