Multiview Canonical Correlation Analysis (MCCA)

Mansi Verma (0801CS171041)
gs0801cs171041@sgsitsindore.in

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. To understand the multiple views Multiview dimensionality reduction has become one of the active topic to avoid the curse of dimensionality. As a powerful tool for multimodal feature fusion, canonical correlation analysis (CCA) has received widespread attentions.

1.2 CCA

Canonical correlation analysis (CCA)is the main technique for two- set data dimensionality reduction such that the correlation between the pairwise variables in the common subspace is mutually maximized. CCA was originally proposed by H . Hotelling in 1935. For finding correlation between more than two views CCA fails so in order to solve the problem MCCA is introduced.

1.3 MCCA

MCCA is a powerful technique for analyzing linear relationships between more (than two) sets of random variables It is an extension of CCA from two view scenario to include three or more number of data channels. Defining a measure of cross correlation for more than two random variables is not straightforward and many possible measures have been proposed. . Typical approaches define cross-correlation as a function of pairwise correlations between variables (for example the sum, product or sum of squares).

According to the definition of cross-view correlation for multi-view variables, we roughly split the multi-view CCA models into three groups, i.e., 1) the pairwise-correlation based methods 2) the "zero-order-correlation models and 3) the high-order-correlation based methods. Here only pairwise-correlation based method is mentioned.

For the pairwise-correlation-based group, there are still many different ways to measure the correlation degree, such as, by maximizing 1) the sum of all entries in the correlation matrix (SUMCOR), 2) the sum of squares of all entries in the correlation matrix (SSQCOR) and 3) the largest eigenvalue of the correlation matrix (MAXVAR); or by minimizing 4) the smallest eigenvalue of correlation matrix (MINVAR) and 5) the determinant of the correlation matrix (GENVAR).

Considering the objective functions of these models are highly related, this subsection will take only the SUMCOR as an example for elaboration.

The SUMCOR aims to find a projection for each view such that the sum of all possible pairwise correlation, after projecting each view onto the subspace is maximized.

Chapter 2

Mathematical Formulation

2.1 Formulation

The MCCA finds a set of m projections so that the correlation between paired data sets is maximized in the common feature sub- space. Given the m sets of variables $X = (X^1, X^2, \ldots \ldots . X^m)$,where $X^i = (x_1^i, x_2^i, \ldots \ldots \ldots . x_N^i) \in R^{d_i \times N}$ is the i^{th} view data with the dimensionality of d_i. The SUMCOR aims to find a projection w_i for each view such that the sum of all possible pairwise correlation, after projecting each view onto the subspace $X^{i'} w_i$ is maximized.

$$\arg max \ \ \rho = \sum_{i=1}^{m} \sum_{j=1}^{m} w_i^i C_{ij} w_j \qquad (2.1)$$

$$s.t. \ \ w_i^i C_{ij} w_j = 1 , i = 1,2, \ldots \ldots \ldots . m$$

Where the Cij denotes the within-view covariance matrix $(i = j)$ or between-view covariance matrix $(i \neq j)$.Solving this problem by Lagrange multiplier technique, we can obtain a generalized multivariate eigenvalue problem (MEP).

Existing study has proven that the MEP problem has no analytical solutions when m ≥ 3. The approximate solutions are found by resorting to the power iteration method with several restarts the initial vector for approaching to the local optimum. This method was proposed by Horst.

The second form of SUMCOR reduces the couple constraint as follows

$$\arg max \ \ \rho = \sum_{i=1}^{m} \sum_{j=1}^{m} w_i^i C_{ij} w_j \qquad (2.2)$$

$$s.t. \sum_{i=1}^{m} w_i^i C_{ii} w_i = 1$$

This model can be easily solved by the Lagrange multiplier technique, and it is equivalent to optimizing a generalized eigenvalue problem below

$$\begin{bmatrix} C_{11} & \dots & C_{1m} \\ \vdots & \ddots & \vdots \\ C_{m1} & \dots & C_{mm} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} = \lambda \begin{bmatrix} C_{11} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & C_{mm} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} \qquad (2.3)$$

where,

$$C_{ij} = \frac{1}{N} X^i X^{j^T} \quad for\ i = 1,2,\dots \mathrm{m}\ \ j = 1,2,\dots m$$

$$C_{ii} = \frac{1}{N} X^i X^{i^T} \ for\ i = 1,2,\dots,m$$

When the feature dimensionality is high, especially when $d_i > N$. the covariance matrix $X^i X^{i^T}$ is singular, and hence the optimization problem is underdetermined. Regularizations can be added to the covariance matrices to remedy this problem by introducing:

$$C_{ii} = \frac{1}{N} X^i X^{i^T} + r_x \mathrm{I}\ for\ i = 1,2,\dots\dots m$$

Here r_x is non-negative regularization coefficient. These two MCCA models measure the multi-view correlation degree by defining the sum of all possible pairwise correlation.

2.2 Horst Algorithm

Existing study has proven that the MEP problem has no analytical solutions when m ≥ 3. Horst has introduced the iterative power method for solving multivariate eigen vale problem. The approximate solutions are found by using this power iteration method with several restarts the initial vector for approaching to the local optimum.

Input : matrices A_{ij}, initial vectors, α_1^0 , where $i, j : 1 \dots m$

Output : $\alpha_1^{maxiter}, \dots \alpha_{m1}^{maxiter}$

$for\ i = 1\ to\ maxiter\ do:$

$for\ j = 1\ to\ m\ \ do$:

$\alpha_j^i \leftarrow \sum_k A_{j,k}\ \alpha_k^{i-1}$

$\alpha_j^i \leftarrow \frac{\alpha_j^i}{\sqrt{\alpha_j^i\, \alpha_j^{i^T}}}$

$end\ for$

$end\ for$

Here A_{ij} is the covariance matrix and $\alpha_1^{maxiter} \dots . \alpha_{m1}^{maxiter}$ are projection vectors for m views.

2.3 Universe subspace

MCCA is viewed as learning a common subspace such that correlation between multiple data views is maximized. The common subspaces between m views can be formulated as

$$U = \frac{1}{m}\sum_{v=1}^{m}(X_v\ w_v) \tag{2.4}$$

Where, m is total number of views, and U is universe subspace of all views.

Chapter 3

Algorithm

In this section, we give an overview of the MCCA algorithms where we formulate the optimization problem as a standard eigen problem.

3.1 MCCA algorithm

Algorithm 1: MCCA algorithm

Input : Multiview training datasets $X = (X1, X2,\cdots, Xm)$
Where $X^i = \left(x_1^i, x_2^i, \ldots\ldots\ldots . x_N^i \right) \in R^{d_i \times N}$ and m are number of views d_i is dimensionality of X^i and
n_components = number of reduced dimensions
Output : $\alpha_j \;\; s.t. j = 1,2, \ldots m$

1. Normalize all the $\boldsymbol{X^i} \;\; \boldsymbol{i.e.X^i = X^i - \overline{X^i}}$
2. Calculate between-view covariance matrix of all pairs

$$\boldsymbol{C_{ij} = \frac{1}{N} X^i X^{j^T} \quad for\ i = 1, 2, \ldots \mathrm{m} \;\; j = 1, 2, \ldots m}$$

3. Calculate within-view covariance matrix of all pairs

$$\boldsymbol{C_{ii} = \frac{1}{N} X^i X^{i^T} \quad for\ i = 1, 2, \ldots, m}$$

   add regularization parameter to it

$$\boldsymbol{C_{ii} = \frac{1}{N} X^i X^{i^T} + r_x \mathrm{I} \quad for\ i = 1, 2, \ldots\ldots m}$$

4. Calculate alpha (projection vector) corresponding to each view

   using horst algorithm (Iterative power method)

   4.1 Initialize $\boldsymbol{\alpha_i^0 \;\; for\ i = 1, 2, \ldots m}$ with some random values.

   4.2 Calculate $\boldsymbol{\alpha_j^i}$

$$\boldsymbol{\alpha_j^i} = \textstyle\sum_k A_{j,k} \; \alpha_k^{i-1} \quad \boldsymbol{for\ k = 1, 2, \ldots\ldots m}$$
$$\boldsymbol{for\ i = 1, 2, \ldots\ldots n\_components \quad for\ j = 1, 2, \ldots\ldots m}$$

$$\boldsymbol{\alpha_j^i = \frac{\alpha_j^i}{\sqrt{\alpha_j^i \alpha_j^{i^T}}}}$$

5. Return the $\boldsymbol{\alpha_j}$(projection vectors) for each $\boldsymbol{j^{th}}$ view having n_components (reduced d)

Chapter 4

Documentation of API

4.1 Package organization

MCCA

Class MCCA(n_components = 2 , reg_param = 0.001)

4.2 Parameters :

- **n_components** : int (default = 2)
  numbers of components to keep i.e. up to how many components are needed in the reduced dimensionality

- **reg_param** : int,float (default = 0.001)
  a non-negative regularization coefficient. In case of high dimensionality features within-view covariance matrix is singular. To resolve this issue reg_param is added.

4.3 Attributes:

m--> corresponds to the number of view

n--> corresponds to the number of samples in each view

- **dimen**: array,[1,m]
  array of feature dimension of all views

- **C** : array,[m,m]
  Covariance matrix of each pair of views

- **weights**: array[m,n_components]
  array of canonical covariate or projection vectors for each view

- **weights_transform** : array[n,n_components]
  array that stores the transformed views by applying the fitted model on it

4.4 Methods :

- **fit (X_list) :**
  Fit the MCCA model to the passed data

- **transform (X_list)**
  Apply the learned model or dimensionality reduction to the trained data

- **fit_tranform (X_list)**
  Fit the MCCA on the passed data arguments and apply learned dimensionality reduction on the passed data (on which the model is learned)

- **__init__(n_components =2 , re_param = 0.001)**

Intialize self for MCCA class and set parameters for instantiating objects

- **fit(X_list):**

Fit the model to the data
Parameters : X_list : list of m arrays of shape(n_samples,n_features)

Training vector to learn the model where n_samples is the number of data samples and n_features is dimensionality of each sample

- **transform (X_list) :**

Apply the learned model to the data and transform it.
Parameter: X_list: list of m arrays of shape(n_samples,n_features)

Training vector to learn the model where n_samples is number of data samples in each view and n_features is dimensionality of each view.

Returns: X_reduced : list of views having reduced dimensionality.

- **fit_tranform (X_list)**

Fit the model to the data and transform the fitted data on the basis of the learned model

Parameter : X_list : list of m array of shape(n_samples,n_features)

Training vector to learn the model where n_samples is number of data samples in each view and n_features is dimensionality of each view.

Chapter5

Example

5.1 Example 1

from MCCA import MCCA

X1= np.array([[-1, -4,-5], [-25, -1,29], [-33, -24,-1], [1, 1,8], [23, 1,10]])

X2 = np.array([[-1, -1,2,3], [-2,2, -1,4], [-3,4, -2,5], [1,2, 1,7], [2,3, 1,7]])

X3= np.array([[-1, -1,3,2,5], [2, 3,-1,5,1], [-3, 4,2,7,4], [1, 1,3,1,3], [2, 1,3,4,-1]])

X_list = np.array([X1,X2,X3])

mcca = MCCA()

mcca.fit(X_list)

X_reduced = mcca.transform(X_list)

for i in range(len(X_list)):

    print(X_reduced[i])

```
In [5]: from MCCA import MCCA
X1= np.array([[-1, -4,-5], [-25, -1,29], [-33, -24,-1], [1, 1,8], [23, 1,10]])
X2 = np.array([[-1, -1,2,3], [-2,2, -1,4], [-3,4, -2,5], [1,2, 1,7], [2,3, 1,7]])
X3= np.array([[-1, -1,3,2,5], [2, 3,-1,5,1], [-3, 4,2,7,4], [1, 1,3,1,3], [2, 1,3,4,-1]])
X_list =[X1,X2,X3]
mcca = MCCA()
mcca.fit(X_list)
X_reduced = mcca.transform(X_list)
for i in range(len(X_list)):
    print(X_reduced[i])

[[ 0.12818527  0.14597102]
 [-0.06308171 -0.20361597]
 [-2.37267616 -2.26314926]
 [ 0.7330569   0.69347911]
 [ 1.57451571  1.62731509]]
[[-0.25957385 -1.19858451]
 [-1.30921409 -1.19971609]
 [-1.71471927 -1.00795624]
 [ 1.45302112  1.44773255]
 [ 1.83048608  1.95852429]]
[[ 0.13631027 -0.41328243]
 [-0.23973613 -0.09578937]
 [-2.48896888 -2.04430887]
 [ 0.98367103  0.70922801]
 [ 1.60872372  1.84415265]]
```

Chapter 6 Learning Outcome

- Learnt the concept of Multiview data and how to map data from multiple views into common subspace so that correlation between them remain maximum.

- Learnt the concept of Canonical Correlation Analysis and using it how we can analyze the Multiview data.

- But CCA only maximizes correlation between two set of variables so I learnt how to apply MCCA to maximize correlation between more than two set of variables

- For m>=3 i.e for maximizing correlation between more than 2 views Multivariate Eigen Value Problem was generated. But to solve it incase of m>2 it was difficult so I learnt the approximation algorithm proposed by Horst to solve Multivariate eigen value problem

References

- https://www.researchgate.net/publication/228836443_Multi-View_Canonical_Correlation_Analysis
- A Survey on Canonical Correlation Analysis - IEEE Journals & Magazine
- https://arxiv.org/abs/1907.01693
- https://mtchu.math.ncsu.edu/Research/Lectures/natalk_multvariate.pdf