2-D Square Canonical Correlation Analysis
$(2\text{-D})^2$CCA

Bhanupratap Baghel (0801CS183D04)
Lokesh Dohare (0801CS171038)
Sachi Parashar (0801CS183D14)

December 13, 2020

Contents

Chapter 1

Introduction

1.1 CCA

Canonical correlation analysis (CCA) is an important method in feature extraction and multivariate data analysis,which aims to find the correlation between two sets of variables.In order to extract the lower-dimensional and effectivefeatures, CCA seeks a pair of linear transformations such thatthe projected variables have the maximal correlation. Up tonow, CCA has been applied in many fields, for instance, statisticalanalysis, information retrieval, facial expressionrecognition, genomic data analysis, image dehazing,closed-loop data identification, fingerprint recognition, palmprint recognition [9], and machine learning.

1.2 Need of 2-D CCA

For the defects of the 1D method, researchers have proposedsome two-dimensional (2D) data analysis methods,2D methods use images directly without reshaping theminto vectors, thus they can achieve better performance and needless computing time.But their disadvantage is that the matrices used in two variable groups must have the same number of rows, becausethere exist trace operator in the objective functions.

In the conventional CCA, each image data isreshaped into a long vector, Here we present 2D-CCA where wedirectly use image data to determine relations between them.

1.3 (2D) square CCA

(2D) square CCA is a two directional two dimensional variant of CCA, it can reduce the computational load drastically. (2D)2mCCA extends (2D)2CCA to the multi-set case, and it can deal with the multi-view learning problem. In order to find the nonlinearcorrelation between two groups of images, L(2D)2CCA uses local spatial information to discover

the essential manifold structure. However, the shortcoming of L(2D)2CCA is that it is sensitive to the nearest neighbor parameter *k* and differentparameters may lead to different results.

Like CCA, when the two groups of samples are not linearly correlated, (2D)2CCA may fail to discover the intrinsic correlationof the data due to its substantial linearity.

Chapter 2

Mathematical Formulation

2.1 Formulation

Consider two sets of zero-mean random matrices $\{X_t \in \mathbb{R}^{m_x \times n_x}, t = 1, \cdots, N\}$ and $\{Y_t \in \mathbb{R}^{m_y \times n_y}, t = 1, \cdots, N\}$,

Specifically, the objective function to be maximized is given as follows:

$$\rho = \frac{\mathrm{cov}(l_x^T X r_x, l_y^T Y r_y)}{\sqrt{\mathrm{var}(l_x^T X r_x)}\sqrt{\mathrm{var}(l_y^T Y r_y)}},$$

eq(1)
$$\begin{bmatrix} 0 & \sum_{xy}^{r} \\ \sum_{yx}^{r} & 0 \end{bmatrix} \begin{bmatrix} l_x \\ l_y \end{bmatrix} = \lambda \begin{bmatrix} \sum_{xx}^{r} & 0 \\ 0 & \sum_{yy}^{r} \end{bmatrix} \begin{bmatrix} l_x \\ l_y \end{bmatrix},$$

eq(2)
$$\begin{bmatrix} 0 & \sum_{xy}^{1} \\ \sum_{yx}^{l} & 0 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \end{bmatrix} = \lambda \begin{bmatrix} \sum_{xx}^{l} & 0 \\ 0 & \sum_{yy}^{l} \end{bmatrix} \begin{bmatrix} r_x \\ r_y \end{bmatrix},$$

Where we define –

$$\boldsymbol{\Sigma}_{xy}^{r} = \left\langle \widetilde{\boldsymbol{X}} \boldsymbol{r}_x \boldsymbol{r}_y^\top \widetilde{\boldsymbol{Y}}^\top \right\rangle = \frac{1}{N} \sum_{t=1}^{N} \widetilde{\boldsymbol{X}}_t \boldsymbol{r}_x \boldsymbol{r}_y^\top \widetilde{\boldsymbol{Y}}_t^\top$$
$$\boldsymbol{\Sigma}_{xx}^{r} = \left\langle \widetilde{\boldsymbol{X}} \boldsymbol{r}_x \boldsymbol{r}_x^\top \widetilde{\boldsymbol{X}}^\top \right\rangle = \frac{1}{N} \sum_{t=1}^{N} \widetilde{\boldsymbol{X}}_t \boldsymbol{r}_x \boldsymbol{r}_x^\top \widetilde{\boldsymbol{X}}_t^\top$$
$$\boldsymbol{\Sigma}_{yy}^{r} = \left\langle \widetilde{\boldsymbol{Y}} \boldsymbol{r}_y \boldsymbol{r}_y^\top \widetilde{\boldsymbol{Y}}^\top \right\rangle = \frac{1}{N} \sum_{t=1}^{N} \widetilde{\boldsymbol{Y}}_t \boldsymbol{r}_r \boldsymbol{r}_y^\top \widetilde{\boldsymbol{Y}}_t^\top.$$

.
$$\boldsymbol{\Sigma}_{xy}^{l} = \left\langle \widetilde{\boldsymbol{X}}^\top \boldsymbol{l}_x \boldsymbol{l}_y^\top \widetilde{\boldsymbol{Y}} \right\rangle$$
$$\boldsymbol{\Sigma}_{xx}^{l} = \left\langle \widetilde{\boldsymbol{X}}^\top \boldsymbol{l}_x \boldsymbol{l}_x^\top \widetilde{\boldsymbol{X}} \right\rangle$$
$$\boldsymbol{\Sigma}_{yy}^{l} = \left\langle \widetilde{\boldsymbol{Y}}^\top \boldsymbol{l}_y \boldsymbol{l}_y^\top \widetilde{\boldsymbol{Y}} \right\rangle.$$

$$\widetilde{\boldsymbol{X}}_t = \boldsymbol{X}_t - \boldsymbol{M}_x, \qquad \widetilde{\boldsymbol{Y}}_t = \boldsymbol{Y}_t - \boldsymbol{M}_y.$$

2.2 Universe subspace

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

$$C = \frac{Xw_X + Y w_Y}{2} \tag{2.5}$$

Eq. (2.5) can be written in general form as:

$$U = \frac{1}{m} \sum_{v=1}^{m_{X_v}} (X_v w_v) \tag{2.6}$$

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

Chapter 3

Algorithm

3.1 (2-D) ^{2}CCA algorithm

The goal of $(2D)^2$ CCA is to seek left transform vectors and l_x,l_y and right transform vectors r_x, r_y to maximize the coorelation between the projections -

$$l_x^T X r_x \text{ and } l_y^T Y r_y.$$

Specifically, the objective function to be maximized is given as follows:

$$\rho = \frac{\text{cov}(l_x^T X r_x, l_y^T Y r_y)}{\sqrt{\text{var}(l_x^T X r_x)}\sqrt{\text{var}(l_y^T Y r_y)}},$$

where cov(·,·) means the covariance between two random variables, var(·) represents the variance of a random variable.

The following two generalized eigenvalue problems are used to solve the transform vectors l_x,l_y and r_x, r_y–

$$\begin{bmatrix} 0 & \sum_{xy}^{r} \\ \sum_{yx}^{r} & 0 \end{bmatrix} \begin{bmatrix} l_x \\ l_y \end{bmatrix} = \lambda \begin{bmatrix} \sum_{xx}^{r} & 0 \\ 0 & \sum_{yy}^{r} \end{bmatrix} \begin{bmatrix} l_x \\ l_y \end{bmatrix},$$
$$\begin{bmatrix} 0 & \sum_{xy}^{1} \\ \sum_{yx}^{l} & 0 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \end{bmatrix} = \lambda \begin{bmatrix} \sum_{xx}^{l} & 0 \\ 0 & \sum_{yy}^{l} \end{bmatrix} \begin{bmatrix} r_x \\ r_y \end{bmatrix},$$

where the terms used in above equations are defined as follows -

Do centering image data to get

$$\widetilde{\boldsymbol{X}}_t = \boldsymbol{X}_t - \boldsymbol{M}_x, \qquad \widetilde{\boldsymbol{Y}}_t = \boldsymbol{Y}_t - \boldsymbol{M}_y.$$

Now, we will calculate the following matrices –

$$\boldsymbol{\Sigma}_{xy}^{r} = \left\langle \widetilde{\boldsymbol{X}} \boldsymbol{r}_x \boldsymbol{r}_y^{\top} \widetilde{\boldsymbol{Y}}^{\top} \right\rangle = \frac{1}{N} \sum_{t=1}^{N} \widetilde{\boldsymbol{X}}_t \boldsymbol{r}_x \boldsymbol{r}_y^{\top} \widetilde{\boldsymbol{Y}}_t^{\top}$$

$$\boldsymbol{\Sigma}_{xx}^{r} = \left\langle \widetilde{\boldsymbol{X}} \boldsymbol{r}_x \boldsymbol{r}_x^{\top} \widetilde{\boldsymbol{X}}^{\top} \right\rangle = \frac{1}{N} \sum_{t=1}^{N} \widetilde{\boldsymbol{X}}_t \boldsymbol{r}_x \boldsymbol{r}_x^{\top} \widetilde{\boldsymbol{X}}_t^{\top}$$

$$\boldsymbol{\Sigma}_{yy}^{r} = \left\langle \widetilde{\boldsymbol{Y}} \boldsymbol{r}_y \boldsymbol{r}_y^{\top} \widetilde{\boldsymbol{Y}}^{\top} \right\rangle = \frac{1}{N} \sum_{t=1}^{N} \widetilde{\boldsymbol{Y}}_t \boldsymbol{r}_r \boldsymbol{r}_y^{\top} \widetilde{\boldsymbol{Y}}_t^{\top}.$$

$$\boldsymbol{\Sigma}_{xy}^{l} = \left\langle \widetilde{\boldsymbol{X}}^{\top} \boldsymbol{l}_x \boldsymbol{l}_y^{\top} \widetilde{\boldsymbol{Y}} \right\rangle$$

$$\boldsymbol{\Sigma}_{xx}^{l} = \left\langle \widetilde{\boldsymbol{X}}^{\top} \boldsymbol{l}_x \boldsymbol{l}_x^{\top} \widetilde{\boldsymbol{X}} \right\rangle$$

$$\boldsymbol{\Sigma}_{yy}^{l} = \left\langle \widetilde{\boldsymbol{Y}}^{\top} \boldsymbol{l}_y \boldsymbol{l}_y^{\top} \widetilde{\boldsymbol{Y}} \right\rangle.$$

By iteratively solving equation (1) and (2), l_x,l_y and r_x, r_y can be determined, respectively. The left transform matrices L_x and L_y can be determined by the eigen vectors associated with the d_1 largest eigenvalues in equation (2). With the similar manner, we can get the right transform matrices.

Chapter 4

Documentation of API

4.1 Package organization

sklearn.cross decomposition.CCA

xindim

Get the dimension of `X`, the first set of variables.

yindim

Get the dimension of `Y`, the second set of variables.

outdim

Get the output dimension, *i.e* that of the common space.

xmean

Get the mean vector of `X` (of length `dx`).

ymean

Get the mean vector of `Y` (of length `dy`).

xprojection

Get the projection matrix for `X` (of size `(dx, p)`).

yprojection

Get the projection matrix for `Y` (of size `(dy, p)`).

correlations

The correlations of the projected componnents (a vector of length `p`).

4.2 Methods

xtransform(*M*, *x*)

Transform observations in the X-space to the common space.

Here, `x` can be either a vector of length `dx` or a matrix where each column is an observation.

ytransform(*M*, *y*)

Transform observations in the Y-space to the common space.

Here, `y` can be either a vector of length `dy` or a matrix where each column is an observation.

fit(*CCA*, *X*, *Y*; ...)

Perform CCA over the data given in matrices `X` and `Y`. Each column of `X` and `Y` is an observation.

`X` and `Y` should have the same number of columns (denoted by `n` below).

This method returns an instance of `CCA`.

ccacov(*Cxx*, *Cyy*, *Cxy*, *xmean*, *ymean*, *p*)

Compute CCA based on analysis of the given covariance matrices, using generalized eigenvalue decomposition.

Chapter 5

Example

5.1 Example 1

```
from sklearn.cross_decomposition import CCA
X = [[1, 2], [4,5]]
Y= [[8, 9], [4,6]]
```

Covariance between left and right transform matrix w.r.t X and Y.

[[1624.5 4702.5], [4702.5 13612.5]]
[[351122. 209500.], [209500. 125000.]]

Cov(x,y) for left transform –
[[1152. -4176.], [-4176. 15138.]]

Cov(x,x) for left transform –
[[364.5 -850.5], [-850.5 1984.5]]

Cov(y,y) for left transform –
[[4186.125 -2264.625],[-2264.625 1225.125]]

Cov(x,y) for right transform –
[[229842. -261369.],[-261369.   297220.5]]

Cov(x,y) for right transform –
[[74884.5 -81850.5],[-81850.5 89464.5]]

Cov(x,y) for right transform –
[[2913094.53125 -2683184.84375],[-2683184.84375 2471420.28125]]

Max Correlation between projections

[[1.30137194e-02 3.76712931e-02 -5.98774845e-01 -3.57264227e-01]
 [3.76712931e-02 1.09048480e-01 -1.73329560e+00 -1.03418592e+00]
 [-5.98774845e-01 -1.73329560e+00 2.75502570e+01 1.64381008e+01
]
 [-3.57264227e-01 -1.03418592e+00 1.64381008e+01 9.80793606e+00
]]

Matrices for calculating eigen values and eigen vector for left transform

[[364.5   0.   ]
 [1225.125   0.   ]]

Eigen Values
[-1056.35660749 1420.85660749]

Eigen Vector
[[0.65301564 -0.75734442], [-0.75734442 -0.65301564]]

verify 1st e-val/vec pair
[927.84158598   0.   ]

verify 2nd e-val/vec pair
[800.02578517   0.   ]

Matrices for calculating eigen values and eigen vector for right transform

[[74884.5   0.   ]
 [2471420.28125   0.   ]]

Eigen Values
[-2434261.64178374 2509146.14178374]

Eigen Vector
[[0.70173059 -0.7124424], [-0.7124424 -0.70173059]]

verify 1st e-val/vec pair
[1.76074460e+06 -2.32830644e-10]

verify 2nd e-val/vec pair
[1734271.2141338   0.   ]

Chapter 6

Learning Outcomes

6.1 In this project, we have presented a two dimensional square extension of 2-D CCA that directly takes image data as inputs without reshaping them into vectors, in order to correlate relationships between them.

6.2 The main advantage of $(2D)^2$-CCA is:

- Multi-view learning problem.
- Preserving spatial structure of image data in the calculation of canonical variable matrices.

Bibliography

[1] Sun Xizhan Gao , *Student Member, IEEE*, Sijie Niu, and Quansen Sun: Two–Directional Two-Dimensional Kernel Canonical Correlation Analysis, vol. 26,No. 11, November 2019.