L 2-D Canonical Correlation Analysis (L 2-D CCA)

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

December 20, 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 effective features, CCA seeks a pair of linear transformations such that the projected variables have the maximal correlation. Up to now, CCA has been applied in many fields, for instance, statistical analysis, information retrieval, facial expression recognition, 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 proposed some two-dimensional (2D) data analysis methods, 2D methods use images directly without reshaping them into vectors, thus they can achieve better performance and need less computing time. But their disadvantage is that the matrices used in two variable groups must have the same number of rows, because there exist trace operator in the objective functions.

In the conventional CCA, each image data is reshaped into a long vector, Here we present 2D-CCA where we directly 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 nonlinear correlation 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 different parameters 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 correlation of the data due to its substantial linearity.

1.4 Local 2-D CCA

Like CCA, when the two sets of sample points are not linearly correlated, 2DCCA may fail to discover the intrinsic correlation of the data due to its substantial linearity. Motivated by the basic idea of locality-based learning and spectral clustering, we develop a local 2DCCA (L2DCCA) technique to discover the intrinsic correlation between two sets of images. L2DCCA is a manifold extension to 2DCCA. Different from 2DCCA that finds globally linear correlation, L2DCCA seeks locally linear correlation to model the underlying correlation. Computationally, L2DCCA is solved as a generalized eigenvalue problem.

Chapter 2

Mathematical Formulation

2.1 Formulation

Now we consider two sets of image data, $\{\boldsymbol{X}_t \in \mathbb{R}^{m_x \times n_x},\ t = 1, \ldots, N\}$ and $\{\boldsymbol{Y}_t \in \mathbb{R}^{m_y \times n_y},\ t = 1, \ldots, N\}$ that are realizations of random variable matrix $\boldsymbol{X}$ and $\boldsymbol{Y}$,

$$
\begin{aligned}
\mathbf{C}^r_{xy} &= \frac{1}{2n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{X}_i - \mathbf{X}_j)\mathbf{r}_x \mathbf{r}_y^T (\mathbf{Y}_i - \mathbf{Y}_j)^T, \\
\mathbf{C}^r_{xx} &= \frac{1}{2n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{X}_i - \mathbf{X}_j)\mathbf{r}_x \mathbf{r}_x^T (\mathbf{X}_i - \mathbf{X}_j)^T, \\
\mathbf{C}^r_{yy} &= \frac{1}{2n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{Y}_i - \mathbf{Y}_j)\mathbf{r}_y \mathbf{r}_y^T (\mathbf{Y}_i - \mathbf{Y}_j)^T.
\end{aligned}
$$

$$
\arg \max_{\mathbf{l}_x, \mathbf{r}_x, \mathbf{l}_y, \mathbf{r}_y} \sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{l}_x^T \mathbf{X}_i \mathbf{r}_x - \mathbf{l}_x^T \mathbf{X}_j \mathbf{r}_x)(\mathbf{l}_y^T \mathbf{Y}_i \mathbf{r}_y - \mathbf{l}_y^T \mathbf{Y}_j \mathbf{r}_y)
$$

$$
\begin{aligned}
s.\ t.\ & \sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{l}_x^T \mathbf{X}_i \mathbf{r}_x - \mathbf{l}_x^T \mathbf{X}_j \mathbf{r}_x)^2 = 1, \\
& \sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{l}_y^T \mathbf{Y}_i \mathbf{r}_y - \mathbf{l}_y^T \mathbf{Y}_j \mathbf{r}_y)^2 = 1.
\end{aligned}
$$

$$
\begin{aligned}
& \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{A}^x_{ij} (\mathbf{l}_x^T \mathbf{X}_i \mathbf{r}_x - \mathbf{l}_x^T \mathbf{X}_j \mathbf{r}_x) \mathbf{A}^y_{ij} (\mathbf{l}_y^T \mathbf{Y}_i \mathbf{r}_y - \mathbf{l}_y^T \mathbf{Y}_j \mathbf{r}_y) \\
&= \mathbf{l}_x^T \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{A}^{xy}_{ij} (\mathbf{X}_i \mathbf{r}_x - \mathbf{X}_j \mathbf{r}_x)(\mathbf{Y}_i \mathbf{r}_y - \mathbf{Y}_j \mathbf{r}_y)^T \right) \mathbf{l}_y \\
&= 2\mathbf{l}_x^T \mathbf{X}_r \mathbf{L}^{xy} \mathbf{Y}_r^T \mathbf{l}_y \qquad (15
\end{aligned}
$$

where X_r, Y_r and $L^{xy} = D^{xy} - A^{xy}$ is the Laplacian Matrix, where D^{xy} is the diagonal matrix with the diagonal entries being row sums of A^{xy}.

With above derivation, the formulation of L2DCCA can be rewritten as

$$\begin{aligned} &\arg\max\ \mathbf{l}_x^T \mathbf{X}_r \mathbf{L}^{xy} \mathbf{Y}_r^T \mathbf{l}_y \\ &\quad s.\,t.\ \mathbf{l}_x^T \mathbf{X}_r \mathbf{L}^{xx} \mathbf{X}_r^T \mathbf{l}_x = 1,\ \mathbf{l}_y^T \mathbf{Y}_r \mathbf{L}^{yy} \mathbf{Y}_r^T \mathbf{l}_y = 1 \end{aligned}$$

$$\begin{aligned} &\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{A}_{ij}^{x}(\mathbf{l}_x^T \mathbf{X}_i \mathbf{r}_x - \mathbf{l}_x^T \mathbf{X}_j \mathbf{r}_x)\mathbf{A}_{ij}^{y}(\mathbf{l}_y^T \mathbf{Y}_i \mathbf{r}_y - \mathbf{l}_y^T \mathbf{Y}_j \mathbf{r}_y) \\ &= \mathbf{r}_x^T \left(\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{A}_{ij}^{xy} (\mathbf{X}_i^T \mathbf{l}_x - \mathbf{X}_j^T \mathbf{l}_x)(\mathbf{Y}_i^T \mathbf{l}_y - \mathbf{Y}_j^T \mathbf{l}_y)^T \right) \mathbf{r}_y \end{aligned}$$

$$\begin{aligned} &\arg\max\ \mathbf{r}_x^T \mathbf{X}_l \mathbf{L}^{xy} \mathbf{Y}_l^T \mathbf{r}_y \\ &\quad s.\,t.\ \mathbf{r}_x^T \mathbf{X}_l \mathbf{L}^{xx} \mathbf{X}_l^T \mathbf{r}_x = 1,\ \mathbf{r}_y^T \mathbf{Y}_l \mathbf{L}^{yy} \mathbf{Y}_l^T \mathbf{r}_y = 1 \end{aligned}$$

$$\begin{aligned} &\begin{pmatrix} \mathbf{0} & \mathbf{X}_r \mathbf{L}^{xy} \mathbf{Y}_r^T \\ \mathbf{Y}_r \mathbf{L}^{xy} \mathbf{X}_r^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{l}_x \\ \mathbf{l}_y \end{pmatrix} \\ &\qquad = \lambda \begin{pmatrix} \mathbf{X}_r \mathbf{L}^{xx} \mathbf{X}_r^T & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_r \mathbf{L}^{yy} \mathbf{Y}_r^T \end{pmatrix} \begin{pmatrix} \mathbf{l}_x \\ \mathbf{l}_y \end{pmatrix}. \end{aligned}$$

Chapter 3

Algorithm

3.1 Local 2-D CCA algorithm

We start with reformulating 2DCCA using pair wise image samples. The covariance matrices defined in 2DCCA can be respectively expressed as

$$\mathbf{C}^r_{xy} = \frac{1}{2n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{X}_i - \mathbf{X}_j)\mathbf{r}_x\mathbf{r}_y^T(\mathbf{Y}_i - \mathbf{Y}_j)^T,$$
$$\mathbf{C}^r_{xx} = \frac{1}{2n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{X}_i - \mathbf{X}_j)\mathbf{r}_x\mathbf{r}_x^T(\mathbf{X}_i - \mathbf{X}_j)^T,$$
$$\mathbf{C}^r_{yy} = \frac{1}{2n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{Y}_i - \mathbf{Y}_j)\mathbf{r}_y\mathbf{r}_y^T(\mathbf{Y}_i - \mathbf{Y}_j)^T.$$

With above equations, we have that 2DCCA can be reformulated as a constrained maximization problem:

$$\arg\max_{\mathbf{l}_x,\mathbf{r}_x,\mathbf{l}_y,\mathbf{r}_y} \sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{l}_x^T\mathbf{X}_i\mathbf{r}_x - \mathbf{l}_x^T\mathbf{X}_j\mathbf{r}_x)(\mathbf{l}_y^T\mathbf{Y}_i\mathbf{r}_y - \mathbf{l}_y^T\mathbf{Y}_j\mathbf{r}_y)$$

$$s.\,t.\ \sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{l}_x^T\mathbf{X}_i\mathbf{r}_x - \mathbf{l}_x^T\mathbf{X}_j\mathbf{r}_x)^2 = 1,$$
$$\sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{l}_y^T\mathbf{Y}_i\mathbf{r}_y - \mathbf{l}_y^T\mathbf{Y}_j\mathbf{r}_y)^2 = 1.$$

So, 2DCCA is reinterpreted as the correlation of the differences between all the pairwise projected data points.

The 2DCCA is suited for linear canonical correlation. For complex correlation situation, the linear canonical correlation holds locally. The pairwise expression allows us to define correlation locally. Specifically, we consider maximizing the objective function:

eq(1)
$$\arg \max_{\mathbf{l}_x, \mathbf{r}_x, \mathbf{l}_y, \mathbf{r}_y} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{A}_{ij}^{x} (\mathbf{l}_x^T \mathbf{X}_i \mathbf{r}_x - \mathbf{l}_x^T \mathbf{X}_j \mathbf{r}_x) \times \mathbf{A}_{ij}^{y} (\mathbf{l}_y^T \mathbf{Y}_i \mathbf{r}_y - \mathbf{l}_y^T \mathbf{Y}_j \mathbf{r}_y)$$

eq(2)
$$s.\,t. \quad \sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{A}_{ij}^{x})^2 (\mathbf{l}_x^T \mathbf{X}_i \mathbf{r}_x - \mathbf{l}_x^T \mathbf{X}_j \mathbf{r}_x)^2 = 1,$$

eq(3)
$$\sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{A}_{ij}^{y})^2 (\mathbf{l}_y^T \mathbf{Y}_i \mathbf{r}_y - \mathbf{l}_y^T \mathbf{Y}_j \mathbf{r}_y)^2 = 1.$$

where A_{ij}^{x} is the weight imposed on the edge that connects sample points X_i and X_j.

It is noted that, in this letter, the distance between nodes i and j is defined as the Frobenius norm between the matrices X_i and X_j, i.e.

$d(i,j) = ||X_i – X_j||_F$

We term eq(1)-(3) as local 2DCCA.

The strategy introduced in L2DCCA is that the data points are not processed equally. Rather, a heavy weight is put between two close points.

To get the matrix A^x in symmetric form:

$A^x := (A^x + (A^x)^T)/2.$

Now, from eq(1),

$$\begin{aligned}
&\sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{A}_{ij}^{x} (\mathbf{l}_x^T \mathbf{X}_i \mathbf{r}_x - \mathbf{l}_x^T \mathbf{X}_j \mathbf{r}_x) \mathbf{A}_{ij}^{y} (\mathbf{l}_y^T \mathbf{Y}_i \mathbf{r}_y - \mathbf{l}_y^T \mathbf{Y}_j \mathbf{r}_y) \\
&= \mathbf{l}_x^T \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{A}_{ij}^{xy} (\mathbf{X}_i \mathbf{r}_x - \mathbf{X}_j \mathbf{r}_x)(\mathbf{Y}_i \mathbf{r}_y - \mathbf{Y}_j \mathbf{r}_y)^T \right) \mathbf{l}_y \\
&= 2 \mathbf{l}_x^T \mathbf{X}_r \mathbf{L}^{xy} \mathbf{Y}_r^T \mathbf{l}_y \qquad (15
\end{aligned}$$

where $X_r = [X_1 r_x, \ldots, X_n r_x]$, $Y_r = [Y_1 r_y, \ldots, Y_n r_y]$, and $L^{xy} = D^{xy} - A^{xy}$ is the Laplacian matrix, where D^{xy} is the diagonal matrix with the diagonal enteries being row sums of A^{xy}.

With above derivation, the formulation of eq(1) – (3) is rewritten as

eq(4)
eq(5)

$$\arg\max\ \mathbf{l}_x^T \mathbf{X}_r \mathbf{L}^{xy} \mathbf{Y}_r^T \mathbf{l}_y$$
$$s.\,t.\ \mathbf{l}_x^T \mathbf{X}_r \mathbf{L}^{xx} \mathbf{X}_r^T \mathbf{l}_x = 1,\ \mathbf{l}_y^T \mathbf{Y}_r \mathbf{L}^{yy} \mathbf{Y}_r^T \mathbf{l}_y = 1$$

where $L^{xx} = D^{xx} - A^{xx}$ is the Laplacian matrix, and D^{xx} is the diagonal matrix with the diagonal entries being row sums of A^{xx} whose entries are defined as $A_{ij}^{xx} = (A_{ij}^{x})^2$.

We also have,

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{A}_{ij}^{x}(\mathbf{l}_x^T \mathbf{X}_i \mathbf{r}_x - \mathbf{l}_x^T \mathbf{X}_j \mathbf{r}_x) \mathbf{A}_{ij}^{y}(\mathbf{l}_y^T \mathbf{Y}_i \mathbf{r}_y - \mathbf{l}_y^T \mathbf{Y}_j \mathbf{r}_y)$$
$$= \mathbf{r}_x^T \left(\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{A}_{ij}^{xy} (\mathbf{X}_i^T \mathbf{l}_x - \mathbf{X}_j^T \mathbf{l}_x)(\mathbf{Y}_i^T \mathbf{l}_y - \mathbf{Y}_j^T \mathbf{l}_y)^T \right) \mathbf{r}_y$$

We can write eq(1)-(3) alternatively like –

Eq(6)
Eq(7)

$$\arg\max\ \mathbf{r}_x^T \mathbf{X}_l \mathbf{L}^{xy} \mathbf{Y}_l^T \mathbf{r}_y$$
$$s.\,t.\ \mathbf{r}_x^T \mathbf{X}_l \mathbf{L}^{xx} \mathbf{X}_l^T \mathbf{r}_x = 1,\ \mathbf{r}_y^T \mathbf{Y}_l \mathbf{L}^{yy} \mathbf{Y}_l^T \mathbf{r}_y = 1$$

Like in 2DCCA, the projection directions of L2DCCA are determined iteratively.
We solve eq(4) and (5) with fixed r_x and r_y to obtain l_x and l_y and then solve eq(6) and (7) with fixed l_x and l_y to obtain r_x and r_y, the constrained maximization problem of (4) and (5) is solved as

$$\begin{pmatrix} \mathbf{0} & \mathbf{X}_r \mathbf{L}^{xy} \mathbf{Y}_r^T \\ \mathbf{Y}_r \mathbf{L}^{xy} \mathbf{X}_r^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{l}_x \\ \mathbf{l}_y \end{pmatrix}$$
$$= \lambda \begin{pmatrix} \mathbf{X}_r \mathbf{L}^{xx} \mathbf{X}_r^T & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_r \mathbf{L}^{yy} \mathbf{Y}_r^T \end{pmatrix} \begin{pmatrix} \mathbf{l}_x \\ \mathbf{l}_y \end{pmatrix}.$$

The projection vectors are well determined given a set of training images. In our algorithm, we find that the iterative algorithm is not sensitive with the initialization.

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

Input –
7 8 9
4 5 6
1 2 3
[[7, 8, 9], [4, 5, 6], [1, 2, 3]]
7 8 9
6 2 1
1 4 7
[[7, 8, 9], [6, 2, 1], [1, 4, 7]]

Output –

```
[[7. 6. 5.]
 [6. 5. 4.]
 [5. 4. 3.]]
[[7.  7.  5. ]
 [7.  2.  2.5]
 [5.  2.5 7. ]]
[[49. 42. 25.]
 [42. 10. 10.]
 [25. 10. 21.]]
[[1.81141623e+18 1.82633364e+17
2.66316197e+15]
 [4.87022305e+17 1.60986462e+16
4.51574089e+15]
 [2.15716120e+17 3.38680567e+15
4.15684543e+15]]
[[7.67203584e+08 1.98273600e+08
3.45744000e+07]
 [1.45604592e+08 1.07565000e+07
2.81232000e+06]
 [5.40216000e+07 6.39360000e+06
4.24569600e+06]]
[[1.81141623e+18 1.82633364e+17
2.66316197e+15]
 [4.87022305e+17 1.60986462e+16
4.51574089e+15]
 [2.15716120e+17 3.38680567e+15
4.15684543e+15]]
```

[[1.81141623e+18 1.82633364e+17
2.66316197e+15]
 [4.87022305e+17 1.60986462e+16
4.51574089e+15]
 [2.15716120e+17 3.38680567e+15
4.15684543e+15]]
[[175392], [177156]]

Chapter 6

Learning Outcomes

6.1 In this project, we have presented a L2DCCA technique to discover the intrinsic correlation between two sets of images. L2DCCA is a manifold extension to 2DCCA.

6.2 2DCCA seeks linear correlation based on images directly. It fails to identify nonlinear correlation between two sets of images. To overcome this, we have used L2DCCA, to identify local correlation. Different from 2DCCA in which images are globally equally treated, L2DCCA weights images differently according to their closeness. That is, the correlation is measured locally, which makes L2DCCA more accurate in finding correlative information.

Bibliography

[1] Haixian Wang, Member, IEEE: Local Two-Dimensional Canonical Correlation Analysis, vol. 17, NO. 11, NOVEMBER 2010.