Locality Preserving Canonical Correlation Analysis (LPCCA)

Sonal Dangi (0801CS171082)
Yashi Magre (0801CS1710950

Contents

1. Introduction

1.1. CCA

Canonical correlation analysis (CCA) is a learning method to find linear relationships between two groups of multidimensional variables. The goal of CCA is to seek two bases which would maximize the correlation of data by projecting two-view data obtained from various information sources, e.g. sound and image. In the past decades, CCA and its variants have been successfully applied to many fields such as image processing, pattern recognition, medical image analysis and data regression analysis. It has been shown that LPCCA performs better than CCA in discovering intrinsic structure of data for some applications, e.g., data visualization and pose estimation. Nevertheless, LPCCA only concerns the correlation between sample pairs and the discrimination of the extracted features which is important in subsequent classification task, while LPCCA is dependent on the parameter k which is manually chosen through experience.

1.2. LPCCA

Locality preserving CCA (LPCCA) is a nonlinear CCA extension, which preserves the local linear structure of the data while performing global nonlinear dimensionality reduction.It can be applied on two views only. LPCCA assumes that the corresponding instances of different views should be as close as possible in the common latent space. In LPCCA, the locality means that the global nonlinear problem is decomposed into local linear ones. So the local structure information can be preserved in the canonical subspace.

The locality methods take into account the local neighborhood structure of the data, and can discover the intrinsic structure of data to a better degree, which benefits subsequent computation. Inspired by the locality based methods, we incorporate such an idea into CCA and propose locality preserving CCA (LPCCA) to discover the local manifold structure of the data and further apply it to data visualization and pose estimation. The experiments show that LPCCA can both capture the intrinsic structure characteristic of the given data and achieve higher pose estimation accuracy than both CCA and KCCA.

2. Mathematical Formulation

2.1. Formulation

The objective function of LPCCA can be expressed as-

$$\min_{\mathbf{w}_x,\mathbf{w}_y} \sum_{i=1}^{N} \left\| \mathbf{w}_x^T \left(\mathbf{x}_i - \overline{\mathbf{x}}\right) - \mathbf{w}_y^T \left(\mathbf{y}_i - \overline{\mathbf{y}}\right) \right\|_2^2,$$

$$s.t. \sum_{i=1}^{N} \left\| \mathbf{w}_x^T \left(\mathbf{x}_i - \overline{\mathbf{x}}\right) \right\|^2 = 1, \sum_{i=1}^{n} \left\| \mathbf{w}_y^T \left(\mathbf{y}_i - \right.\right.$$

where $\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$ and $\overline{\mathbf{y}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{y}_i$ represent the mean vectors of X and Y , respectively. Calculate neighbourhood for each data point using K nearest Neighbour Algorithm. Let Nei(xi) be the neighbor set of xi . Define a similarity matrix Sx, whose ij-th element

$$S_{x,ij} = \begin{cases} \exp\left(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 / t_x\right), & \mathbf{x}_j \in \mathrm{Nei}\left(\mathbf{x}_i\right) \\ 0, & \text{otherwise} \end{cases},$$

$t_x = \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{2\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{N(N-1)}$ represents the mean squared distance between all instances. The similarity matrix S^y of Y can be computed in a similar manner.

Substituting Sx and Sy into (19), the global correlation is decomposed into many local linear correlations between the neighboring instances. The objective of LPCCA is:

$$\max_{\mathbf{w}_x,\mathbf{w}_y} \mathbf{w}_x^T \cdot \sum_{i=1}^{N}\sum_{j=1}^{N} S_{x,ij}\left(\mathbf{x}_i - \mathbf{x}_j\right) S_{y,ij}\left(\mathbf{y}_i - \mathbf{y}_j\right)^T \cdot \mathbf{w}_y$$

$$s.t.\ \mathbf{w}_x^T \cdot \sum_{i=1}^{N}\sum_{j=1}^{N} S_{x,ij}\left(\mathbf{x}_i - \mathbf{x}_j\right) S_{x,ij}\left(\mathbf{x}_i - \mathbf{x}_j\right)^T \cdot \mathbf{w}_x = 1, \quad (21)$$

$$\mathbf{w}_y^T \cdot \sum_{i=1}^{N}\sum_{j=1}^{N} S_{y,ij}\left(\mathbf{y}_i - \mathbf{y}_j\right) S_{y,ij}\left(\mathbf{y}_i - \mathbf{y}_j\right)^T \cdot \mathbf{w}_y = 1,$$

which can be transformed into a generalized eigen decomposition problem

$$\begin{bmatrix} \mathbf{0} & XS_{xy}Y^T \\ YS_{yx}X^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w}_x \\ \mathbf{w}_y \end{bmatrix} = \lambda \begin{bmatrix} XS_{xx}X^T & \mathbf{0} \\ \mathbf{0} & YS_{yy}Y^T \end{bmatrix} \begin{bmatrix} \mathbf{w}_x \\ \mathbf{w}_y \end{bmatrix},$$

where

$$S_{xy} = D_{xy} - S_x \odot S_y,$$
$$S_{yx} = D_{yx} - S_y \odot S_x,$$
$$S_{xx} = D_{xx} - S_x \odot S_x,$$
$$S_{yy} = D_{yy} - S_y \odot S_y,$$

3. Algorithm

3.1. LPCCA Algorithm

Algorithm for Locality Preserving Canonical Correlation Analysis (LPCCA)
Input : Training sets X,Y represents the Two views
Output : Projection vectors W_x , W_y

1. Construct S^x , S^y using the formula given in mathematical formulation

2. Calculate $S_x \odot S_y$, $S_y \odot S_x$, $S_x \odot S_x$, $S_y \odot S_y$)

in which $\odot$ denotes the element wise product operation

3. Calculate D_{YX} , D_{YY} , D_{XX} , D_{XY}

Dxy is a diagonal matrix, with $(D_{xy})_{ii} = \sum_{j=1}^{N} (S_x \odot S_y)_{ij}$ .D_{YX} , D_{YY} , D_{XX} are defined similarly

4. Compute the 4 matrices as shown below

$$S_{xy} = D_{xy} - S_x \odot S_y,$$
$$S_{yx} = D_{yx} - S_y \odot S_x,$$
$$S_{xx} = D_{xx} - S_x \odot S_x,$$
$$S_{yy} = D_{yy} - S_y \odot S_y,$$

5. Compute covariance matrices $C_{xx} = X^T S_{XX} X$, $C_{YY} = Y^T S_{YY} Y$,$C_{XY}=X^T S_{XY} Y$

6. Compute the eigenvectors W_X and W_Y using the formula below

$$C_{xx}^{-1} C_{xy} C_{yy}^{-1} C_{yx} \hat{w}_x = \rho^2 \hat{w}_x$$

7. $C_{yy}^{-1} C_{yx} C_{xx}^{-1} C_{xy} \hat{w}_y = \rho^2 \hat{w}_y$

4. Documentation of API

4.1. Package Organization

class Lpcca.LPCCA (X, Y, k=5)

Parameters:

X: First View of the data
Y: Second View of the data
k: Number of neighbors to be considered

4.2. Methods

__init__(X,Y,k)

To Initialize self.
Where self represents the class object itself.
Prameters:
X and Y are the two views of the same data.
K represents the number of neighbor to be considered.(Default = 5)

meansquareddistances(X)

To find the mean squared distances to find the neighbors associated to a data point.
It takes the raw data as an input and returns the mean squared distances with other data points.
Parameter:
X represent the view of the data.

denom(mean_squared_distances)

To minimize the complexity of the calculations.
It takes mean squared distances as input and returns ans to a mathematical equation for the simplicity of the algorithm.
Parameter:
Mean_squared_distances is the mean squared distances

similarity_matrix(X,k)

It gives similarity matrices reflecting the neighborhood information between samples. It uses k-nearest neighbor to find the neighbors. If xi is

one of the k-nearest neighbors of x j or xj is one of the k-nearest neighbors of xi , they are local neighbors.
Parameter:
X represent the view of the data.
k represent the Number of neighbors to be considered

SdotS()

It gives the intermediate equations to make the algorithm less complex.

lpcca_Covariance_matrices(X,Y,k)

To get covariance matrices. The global correlation is decomposed into many local linear correlations between the neighboring instances. It embeds the neighborhood information into CCA by deleting the contribution of non-neighbors when computing correlation matrices
Parameter:
X represents the view of the data
Y represents the another view of the data
k represent the Number of neighbors to be considered

fit()

Fit model to data. Uses Covariance matrices and SVD decomposition of the data views and returns the weights associated.

fit_transform()

Learn and apply the dimension reduction on the train data. Uses wights obtained from the training and returns the reduced/expected data.

transform(W_x,W_y)

To get the reduced data, with the given weights.
Parameters:
W_x is the weights associated with the view X.
W_y is the weights associated with the view Y.

5. Examples

lpcca=LPCCA(X,Y,4)
X_new,Y_new=lpcca.fit_transform()

5.1. Example 1

Input:
X = [[0., 0., 1.], [1.,0.,0.], [2.,2.,2.], [3.,5.,4.]]
Y = [[0.1, -0.2], [0.9, 1.1], [6.2, 5.9], [11.9, 12.3]]

Output:
Reduced X:
```
[[ 0.86537029  1.24428534  0.14600908]
 [ 0.78912431  1.15294729 -0.04861639]
 [ 6.65602742  2.38296086  0.07179918]
 [14.19644965  2.40722223  0.13072165]]
```

Reduced Y:
```
[[  0.08082053  -0.2573569 ]
 [ -1.72612779   0.1657403 ]
 [-10.47163744  -0.29415229]
 [-20.92435811   0.27055708]]
```

5.2. Example 2

Input:
X = [[0.3, 0.2, 1.5], [1.3,0.2,0.5], [2.3,2.2,2.5], [3.3,5.2,4.5]]
Y = [[0.1, -0.2], [0.9, 1.1], [6.2, 5.9], [11.9, 12.3]]

Output:
Reduced X:
```
[[ 0.86537029  1.24428534  0.14600908]
 [ 0.78912431  1.15294729 -0.04861639]
 [ 6.65602742  2.38296086  0.07179918]
 [14.19644965  2.40722223  0.13072165]]
```

Reduced Y:
```
[[  0.08082053  -0.2573569 ]
 [ -1.72612779   0.1657403 ]
 [-10.47163744  -0.29415229]
 [-20.92435811   0.27055708]
```

6. Learning Outcomes

- Capacity to integrate knowledge and to analyse, evaluate and manage the different data points at a local and global level.

- Capacity to design and perform research on the different aspects of the events while demonstrating insight into the potential and limitations of CCA and thus exploring LPCCA & ALPCCA.

- Analysed and evaluated higher mathematical concepts with the ability to clearly implement and present the conclusions and the knowledge behind it.

A Appendix
References

1. Canonical Correlation Analysis (CCA) Based Multi-View Learning: An Overview
   Chenfeng Guo and Dongrui Wu

2. Multiset Canonical Correlations Analysis With Global Structure Preservation
   HONGJIE ZHANG1 , JINXIN ZHANG2 , YANWEN LIU1 , AND LING JING 1