A new Locality Preserving Canonical Correlation Analysis ALPCCA

Yashi Magre - 0801CS171095
Sonal Dangi - 0801CS171082

Contents

1. Introduction

1.1. Multiview Learning and CCA

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.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.

1.2. ALPCCA (A new Locality Preserving Canonical Correlation Analysis)

ALPCCA aims to find low-dimensional embedding that maintains the local neighbor information between different views.The ALPCCA algorithm works only on two views,is nonlinear and its unsupervised method therefore fails to explore the discriminative information about the samples.ALPCCA starts with the original covariance matrix and adds additional terms explaining the contribution of the neighbors that use the adding strategy.

1.3. ALPCCA V/S LPCCA

For high-dimensional small-sample problems (e.g., face recognition), the '**adding strategy**' used in ALPCCA may be advantageous than the 'deleting strategy' used in LPCCA. For high-dimensional data with low sample count the neighborhoods and hence the similarity matrices **Sx** and **Sy** will typically not be estimated accurately. Because LPCCA only relies on the local neighborhoods for computing the correlation matrices, it will be more severely affected by the inaccurate estimation of those similarity matrices than ALPCCA,which uses both the local neighborhoods and the original covariance matrices. There is a regularization term, which alleviates the effect of inaccurate estimation of $\mathbf{S}^x$ and $\mathbf{S}^y$ to some extent and thus leads to more robust results than LPCCA. Also, the separate use of $\mathbf{S}^x_{ij}$ and $\mathbf{S}^y_{ij}$ in ALPCCA is more robust than the dependent use of them as the product in LPCCA.

2. Mathematical Formulation

2.1. Formulation

ALPCCA is developed directly from the original objective function of CCA.Calculate neighbour for each data point using K nearest Neighbour Algorithm.Let Nei(xi) be the neighbor set of xi . Define a similarity matrix S^x, whose ij-th element

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

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

The objective function of ALPCCA is given in the following form:

$$\max_{\mathbf{w}_x, \mathbf{w}_y} \frac{\mathbf{w}_x^T \tilde{\mathbf{C}}_{xy} \mathbf{w}_y}{\sqrt{(\mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x)(\mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y)}}$$

where

$$\begin{aligned} \tilde{\mathbf{C}}_{xy} &= \sum_{i=1}^{n} \mathbf{x}_i \mathbf{y}_i^T + \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{S}_{ij}^x \mathbf{x}_i \mathbf{y}_j^T + \sum_{j=1}^{n} \sum_{i=1}^{n} \mathbf{S}_{ji}^y \mathbf{x}_i \mathbf{y}_j^T \\ &= \mathbf{X}\mathbf{Y}^T + \mathbf{X}\mathbf{S}^x\mathbf{Y}^T + \mathbf{X}\mathbf{S}^y\mathbf{Y}^T \\ &= \mathbf{X}(\mathbf{I} + \mathbf{S}^x + \mathbf{S}^y)\mathbf{Y}^T \end{aligned}$$

Here, the covariance matrices C_{xx} and C_{yy} are defined as before, i.e., $\mathbf{C}_{xx} = \mathbf{X}^T\mathbf{X}$ and
$\mathbf{C}_{yy} = \mathbf{Y}^T\mathbf{Y}$, $\mathbf{S}^x$ and $\mathbf{S}^y$ are the similarity matrices., and I denote the identity matrix.

To solve ALPCCA, we first rewrite the objective function in as the following equivalent form

$$\max_{\mathbf{w}_x, \mathbf{w}_y} \quad \mathbf{w}_x^T \tilde{\mathbf{C}}_{xy} \mathbf{w}_y$$

$$s.t. \quad \mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x = 1$$

$$\mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y = 1$$

Similarly as standard CCA, by applying the Lagrangian equation, the optimization problem of ALPCCA can be converted to the following generalized eigenvalue decomposition problem.In this paper we will solve the equation using singular vector decomposition.

$$\begin{bmatrix} & \tilde{\mathbf{C}}_{xy} \\ \tilde{\mathbf{C}}_{xy}^T & \end{bmatrix} \begin{bmatrix} \mathbf{w}_x \\ \mathbf{w}_y \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{C}_{xx} & \\ & \mathbf{C}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{w}_x \\ \mathbf{w}_y \end{bmatrix}$$

After this we get projecting matrices $\mathbf{W}_{\mathbf{x}}$ and $\mathbf{W}_{\mathbf{y}}$.

3. Algorithm

3.1. ALPCCA Algorithm

Algorithm for A new Locality Preserving Canonical Correlation Analysis
Input : Training sets X,Y represents the data matrices for Two views
Output : Projection vectors W_x , W_y

1. Construct S^x , S^y using the formula given in mathematical formulation
2. Define $P = I + S^x + S^y$
3. Covariance matrix $C_{xy} = X^TPY$
4. Compute covariance matrices $C_{xx} = X^TX$,$C_{yy} = Y^TY$
5. Compute matrix $H = C_{xx}^{-1/2}C_{xy}C_{yy}^{-1/2}$
6. Perform SVD decomposition on H: $H = UDV^T$
7. Choose U = [U1...Ud],V = [V1...Vd],d < n;
8. $W_x=C_{xx}^{-1/2}U$, $W_y=C_{yy}^{-1/2}V$
9. Dimensionally reduced data new X $=XW_x$, new Y$=YW_y$

4. Documentation of API

4.1. Package Organization

class Alpcca.ALPCCA (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

covariance_matrices(X,Y,k)

To get covariance matrices. It starts with the original covariance matrix and adds additional terms explaining the contribution of the neighbors.
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

alpcca=ALPCCA(X,Y,4)
x,y=alpcca.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:
```
[[-3.89568618e-01  2.13556612e-03 -4.17675651e-03]
 [-4.31846932e-01 -6.58903451e-03  4.99618852e-03]
 [-2.36052549e+00 -2.68020548e-03  1.58106910e-03]
 [-4.64805125e+00  4.34198921e-03 -1.86294779e-03]]
```

Reduced Y:
```
[[ 1.02884411e-02 -6.85580837e-06]
 [-2.08428353e-01  4.44121843e-06]
 [-1.26168393e+00 -7.67186522e-06]
 [-2.52290932e+00  7.53126077e-06]]
```

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:
```
[[-6.88909943e-01 -1.06153608e-03 -3.30302871e-03]
 [-7.32477554e-01 -4.88602738e-03  3.80084631e-03]
 [-2.45463577e+00 -2.14592024e-03  1.32496153e-03]
 [-4.50810869e+00  4.44713467e-03 -1.46458793e-03]]
```

Reduced Y:
```
[[ 1.02884411e-02 -6.85741966e-06]
 [-2.08428353e-01  4.47386062e-06]
 [-1.26168393e+00 -7.47427153e-06]
 [-2.52290932e+00  7.92637635e-06]]
```

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.

Appendix A
References

1. 11A New Locality-Preserving Canonical Correlation Analysis Algorithm for Multi-View Dimensionality Reduction Fengshan Wang · Daoqiang Zhang Published online: 27 July 2012 © Springer Science+Business Media, LLC. 2012

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

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