Shri G.S. Institute of Technology and Science, Indore

Department of Computer Science

Data Science Project 2

SPCCA - Sparsity Preserving Canonical Correlation Analysis
LPCCA - Locality Preserving Canonical Correlation Analysis

Submitted By:

Rithik Sachdev

Shreya Mishra

Shekhar Sharma

Submitted To:

Prof. S.Gupta

Indore, India

13th December 2020

Contents

Introduction

Dimensionality reduction is an important procedure in the field of machine learning and pattern recognition along with data classification. The different algorithms such as K-nearest neighbours (KNN), Support vector machine, naive bayes, and etc. have been used for applications such as pattern recognition, image retrieval, text analysis and retrieval, and bioinformatics but the data used always have a high dimensionality which requires a lot of processing, memory and time. Moreover, a high dimension of data could lead to a curse of dimensionality. Therefore, dimensionality reduction strategy is used.

One of the major approaches involved in dimensionality reductions has been the computation of principal component analysis wherein higher dimensions can be reduced to lower dimensions using the concept of covariance matrix, eigenvalues and eigenvectors. It also gives details regarding the most important variable for clustering the graph and accuracy of the 2-dimensional graph. However, the use of principal components for reducing the dimensions in multi-view learning is limited.

PCA is a vector-based method, in which the image matrices need to be transformed into vectors before further computation. In order to use the image matrix directly, some two-dimensional based (2D-based) methods have been discovered such as 2D-PCA, 2-D LPCCA, 2-D SPCCA, etc. Here, the image vector is used directly rather than reshaping it prior to transformations.

1.1 Multi View learning

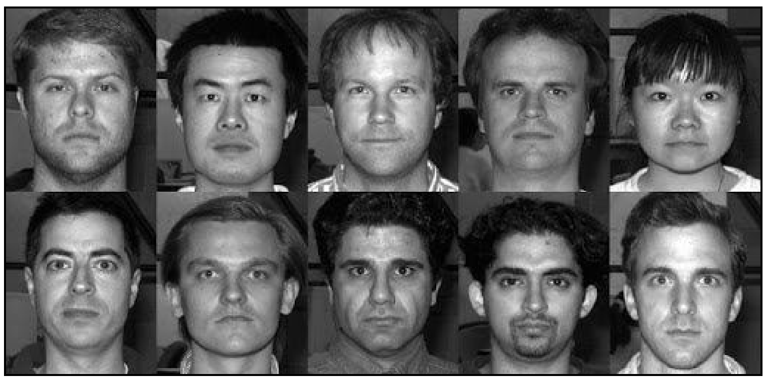
Nowadays, with the rapid development of the Internet, vast amounts of data with multiple views, i.e., sets of features are available.

Multi-view learning refers to fusion, analysis and exploitation of data from multiple data sources or simply from different views of the same data source, in order to achieve higher learning performance and accuracy.

Principal component analysis solely computes correlation for a single view, this is where Canonical Correlation Analysis comes into play.

1.2 CCA

CCA is being widely exploited for the purpose of image recognition, pattern recognition, image recovery, etc. Here, we have worked with the face database provided by Yale University.



Canonical Correlation Analysis was first proposed by Hotelling in 1936 and aimed at finding projection pairs within multiple views and thereby maximizing their correlation. Basically, it is used to compare a set of related variables with another set of related variables, that could be continuous or categorical. As we know, multiple regression can easily be used for finding correlation among different views but because of collinearity(Interaction between variables of the same set), it would not be able to handle the complexity of computations.

One of its advancement has been the General Canonical Correlation Analysis (GCCA) where we first extract two groups of low-dimensional feature vectors from two high-dimensional data sets, then fuse them using a feature fusion method which is either serial or parallel. However, the traditional CCA comes with certain drawbacks-

- It calculates only the linear correlation between two views whereas in many applications, there may be non-linear relationships between the views.

- It takes only 2 views into account.
- There is a wastage of information in CCA as it is an unsupervised algorithm, hence the labels provided in supervised classification are totally ignored by it.

Over the past few years, a large number of approaches have been proposed to deal with the complex nonlinear relationships between different views. Here, we aim to study the Locality preserving canonical correlation analysis (LPCCA) and Sparsity preserving canonical correlation analysis (SPCCA) frameworks.

1.3 LPCCA:

Locality preserving canonical correlation analysis incorporates the information of the neighbours in a multi-view data into CCA. It keeps the neighborhood structure of the data while reducing the dimensionality, and reveals the intrinsic structure information of data. LPCCA assumes that the corresponding instances of different views should be as close as possible in the common latent space. It defines a laplacian matrix S to incorporate the relationship between samples, wherein the larger the entry in S , the larger sample correlation between samples.

1.4 SPCCA:

As we know sparsity is an important way to improve the generalization capability of the model, it can also prevent the weight matrix from suffering from the difficulty of parameter selection as in the case of 2D-LPCCA, hence we can use sparse representation to design the weight matrix.

The reconstructive weight matrix Sx (or Sy) contains discriminant information naturally, and the entry Sx ij represents the contribution of each Xj to reconstruct Xi. In other words, if the entry S^{x}_{ij} is bigger, the sample Xj is more important for reconstructing Xi.

Mathematical Formulation:

2.1 LPCCA:

Let $X = \{X_1, \ldots, X_N\}$, $X_i \in R^{m\times p}$ and $Y = \{Y_1, \ldots, Y_N\}$, $Y_i \in R^{m\times q}$ be two groups of images

$$\begin{aligned} \max_{\alpha,\beta} \quad & \alpha^T \sum_{i,j} G^x_{ij}(X_i - X_j)^T G^y_{ij}(Y_i - Y_j)\beta \\ s.t. \quad & \alpha^T \sum_{i,j} (G^x_{ij})^2 (X_i - X_j)^T (X_i - X_j)\alpha = 1, \\ & \beta^T \sum_{i,j} (G^y_{ij})^2 (Y_i - Y_j)^T (Y_i - Y_j)\beta = 1, \end{aligned}$$

where $G^x = \{G^x_{ij}\}^N_{i,j=1}$ and $G^y = \{G^y_{ij}\}^N_{i,j=1}$ are similarity matrices on graph G. There are many choices to define the similarity matrices, some of them are listed as follows:

(a) 0–1 method: $G_{ij} = 1$ if x_i and x_j are neighbors, otherwise $G_{ij} = 0$.
(b) Dot-product method: $G_{ij} = x_i^T x_j$ if x_i and x_j are neighbors, $G_{ij} = 0$ otherwise.
(c) Heat kernel method: $G_{ij} = e^{-\|x_i - x_j\|^2/t}$ if x_i and x_j are neighbors, $G_{ij} = 0$ otherwise.
(d) Cosine similarly method: $G_{ij} = \frac{x_i^T x_j}{\|x_i\| \|x_j\|}$ if x_i and x_j are neighbors, $G_{ij} = 0$ otherwise.

After some algebraic manipulations, we can get:

$$\begin{aligned} & \alpha^T \sum_{i,j} G^x_{ij}(X_i - X_j)^T G^y_{ij}(Y_i - Y_j)\beta \\ &= \alpha^T \sum_{i,j} G^x_{ij} G^y_{ij} (X_i^T Y_i + X_j^T Y_j - X_i^T Y_j - X_j^T Y_i)\beta \\ &= 2\alpha^T \sum_{i,j} G^x_{ij} G^y_{ij} X_i^T Y_i \beta - 2\alpha^T \sum_{i,j} G^x_{ij} G^y_{ij} X_i^T Y_j \beta \\ &= 2\alpha^T \sum_{i,j} G^{xy}_{ij} X_i^T Y_i \beta - 2\alpha^T \sum_{i,j} G^{xy}_{ij} X_i^T Y_j \beta \\ &= 2\alpha^T \mathcal{X}^T (D^{xy} \otimes I_m - G^{xy} \otimes I_m)\mathcal{Y}\beta \\ &= 2\alpha^T \mathcal{X}^T (L^{xy} \otimes I_m)\mathcal{Y}\beta \end{aligned}$$

where $G^{xy} = G^x \circ G^y$, the symbol $\circ$ denotes an operator such that $(G^x \circ G^y)_{ij} = G^x_{ij} G^y_{ij}$ for matrices G^x and G^y with the same size and G^x_{ij} denotes the ith row jth column entry of matrix G^x. $X = \{ X_1 , . . . , X_N \}$, $Y = \{ Y_1 , . . . , Y_N \}$, D^{xy} is a diagonal matrix with the ith diagonal entry being the ith row sum or column sum of G^{xy} , i.e., $D^{xy} = \mathrm{diag}\left(\sum_{j=1}^{N} G^{xy}_{1j} , . . . , \sum_{j=1}^{N} G^{xy}_{Nj} \right) = \mathrm{diag}\left(\sum_{i=1}^{N} G^{xy}_{1i} , . . . , \sum_{i=1}^{N} G^{xy}_{Ni} \right)$,

and

$$G^{xy} \otimes I_m = \begin{bmatrix} G^{xy}_{11} I_m & \cdots & G^{xy}_{1N} I_m \\ \vdots & \cdots & \vdots \\ G^{xy}_{N1} I_m & \cdots & G^{xy}_{NN} I_m \end{bmatrix}$$

where $\otimes$ denotes the Kronecker product of matrices. Then the maximization problem becomes

$$\begin{aligned} \max_{\alpha,\beta} \quad & \alpha^T \mathcal{X}^T (L^{xy} \otimes I_m) \mathcal{Y} \beta \\ s.t. \quad & \alpha^T \mathcal{X}^T (L^{xx} \otimes I_m) \mathcal{X} \alpha = 1, \\ & \beta^T \mathcal{Y}^T (L^{yy} \otimes I_m) \mathcal{Y} \beta = 1, \end{aligned}$$

where $L^{xx} = D^{xx} - G^{xx}$, $L^{yy} = D^{yy} - G^{yy}$, $G^{xx} = G^x \circ G^x$, $G^{yy} = G^y \circ G^y$, D^{xx} and D^{yy} are diagonal matrices whose definitions are similar to D^{xy}.

Since L^{xy},L^{xx} and L^{yy} are all symmetric and $L^{xy} = L^{yx}$,thus $L^{xy} \otimes I_m$,$L^{xx} \otimes I_m$ and $L^{yy} \otimes I_m$ are also symmetric matrices, and $L^{xy} \otimes I_m = L^{yx} \otimes I_m$. Using Lagrange multiplier method, problem can be solved by the following generalized eigenvalue problems

Once we obtained A = [α 1 , . . . , α d] and B = [β 1 , . . . , β d] which are eigenvectors corresponding to first d largest eigenvalues in Eq.(15) , the following feature fusion strategies FFS1 and

$$\hat{G_{xy}}(\hat{G_y})^{-1}\hat{G_{yx}}\alpha = \lambda^2\hat{G_x}\alpha, \qquad (15)$$
$$\hat{G_{yx}}(\hat{G_x})^{-1}\hat{G_{xy}}\beta = \lambda^2\hat{G_y}\beta,$$

where $\hat{G_{xy}} = \mathcal{X}^T(L^{xy}\otimes I_m)\mathcal{Y}$, $\hat{G_x} = \mathcal{X}^T(L^{xx}\otimes I_m)\mathcal{X}$, $\hat{G_y} = \mathcal{Y}^T(L^{yy}\otimes I_m)\mathcal{Y}$ and $\hat{G_{yx}} = \hat{G_{xy}}^T$.

FFS2 can be used to get the final extracted features.

FFS1: XA+YB,

FFS2: [XA,YB]

2.2 SPCCA:

we can get two sparse weight matrices S^x and S^y for the two sets of variables. Then the optimization

$$\begin{aligned} \max_{\alpha,\beta} \quad & \alpha^T \sum_{i,j} S^x_{ij}(X_i - X_j)^T S^y_{ij}(Y_i - Y_j)\beta \\ s.t. \quad & \alpha^T \sum_{i,j} (S^x_{ij})^2 (X_i - X_j)^T (X_i - X_j)\alpha = 1, \\ & \beta^T \sum_{i,j} (S^y_{ij})^2 (Y_i - Y_j)^T (Y_i - Y_j)\beta = 1, \end{aligned}$$

problem of 2D-SPCCA can be formulated as follows:

Using the similar method below, we can get the optimal solution.

$$\begin{aligned} \max_{\alpha,\beta} \quad & \alpha^T \sum_{i,j} G^x_{ij}(X_i - X_j)^T G^y_{ij}(Y_i - Y_j)\beta \\ s.t. \quad & \alpha^T \sum_{i,j} (G^x_{ij})^2 (X_i - X_j)^T (X_i - X_j)\alpha = 1, \\ & \beta^T \sum_{i,j} (G^y_{ij})^2 (Y_i - Y_j)^T (Y_i - Y_j)\beta = 1, \end{aligned}$$

Algorithm:

The algorithm that has been followed is-

1. Extract two feature matrices from the same pattern.
2. Construct two neighbor affinity weight matrices(LPCCA), or sparse reconstructive weight matrices(SPCCA), Sx and Sy.
3. Extract two groups of canonical correlation features.
4. Use feature fusion strategies to fuse the canonical correlation features.

Data: Training data $\mathbf{X} \in \mathbb{R}^{N \times p}$, $\mathbf{Y} \in \mathbb{R}^{N \times q}$, d, α, $\beta, iter_{max}, \epsilon_U, \epsilon_V$.
Result: The d pairs of loading vectors $\mathbf{W}_x \in \mathbb{R}^{p \times d}$ and $\mathbf{W}_y \in \mathbb{R}^{q \times d}$.
$\mu_x = \frac{1}{p} \sum_{j=1}^{p} \mathbf{x}_j$ and $\mu_y = \frac{1}{q} \sum_{i=1}^{q} \mathbf{y}_i$;
for $j \in [1, p]$ **do**
  $\mathbf{x}_j \leftarrow \mathbf{x}_j - \mu_x$
end
for $i \in [1, q]$ **do**
  $\mathbf{y}_i \leftarrow \mathbf{y}_i - \mu_y$
end
Compute $\mathbf{C}_{xy} = \mathbf{X}^\top \mathbf{Y}$ and $\mathbf{U}, \mathbf{V} = \mathrm{SVD}(\mathbf{C}_{xy})$;
Set $\mathbf{U}_d = [\mathbf{u}_1 \cdots \mathbf{u}_d]$, $\mathbf{V}_d = [\mathbf{v}_1 \cdots \mathbf{v}_d]$, $\mathbf{W}_x = \mathbf{U}_d$, $\mathbf{W}_y = \mathbf{V}_d$;
$\Delta_W x = 10\epsilon_w x, \Delta_W y = 10\epsilon_w y, iter = 0$
while $(\Delta_W x > 10\epsilon_U\ ) \vee (\Delta_W y > 10\epsilon_V\ ) \wedge (iter < iter_{max})$ **do**
  $\mathbf{U}_{old} \leftarrow \mathbf{U}, \mathbf{V}_{old} \leftarrow \mathbf{V}$;
  for $i \in [1, q]$ **do**
    Compute $\mathbf{w}_y^i$ using (4),
  end
  Rescale the columns of $\mathbf{W}_y$
  for $i \in [1, p]$ **do**
    Compute $\mathbf{w}_x^i$ using (5),
  end
  Rescale the columns of $\mathbf{W}_x$
  $\Delta_W x \leftarrow \|\mathbf{W}_x - \mathbf{W}_{xold}\|_F$,
  $\Delta_W y \leftarrow \|\mathbf{W}_y - \mathbf{W}_{yold}\|_F$;
  $iter <\leftarrow iter_{max}$
end

References:

- *et. al.* Xizhan Gao, Quansen Sun, Haitao Xu, Yanmeng Li, 2D-LPCCA and 2D-SPCCA: Two new canonical correlation methods for feature extraction, fusion and recognition. Neurocomputing, *Journal homepage: www. elsevier. com/locate/neucom*