ORTHOGONAL CANONICAL CORRELATION ANALYSIS (OCCA)

Chaitanya Shrivastava (0801EC171023)
gs0801ec171023@sgsits.ac.in

December 13,2020

Contents:

1. Introduction: 2
- 1.1. Multiview Learning 2
- 1.2 Canonical Correlation Analysis. 2
- 1.3 Orthogonal Canonical Correlation Analysis. 3

2. Mathematical Formulation 5
- 2.1 Formulation. 5
- 2.2 Universe Subspace. 7

3. Algorithm: 8
- 3.1 OCCA Algorithm. 8

4. Documentation of API 9
- 4.1 Package Organization. 9
- 4.2 Methods. 10

5. Example 12
- 5.1 Example 1 11
- 5.2 Example 2 11

6. Learning Outcome 13

7. Appendix A: References 14

Chapter 1

Introduction

1.1 Multi-View Learning:

With this growing era of technology, the generation of data/ big data has taken a tremendous leap. The generated data often corresponds to different aspects or views of the same information and analyzing and fusing these interrelated multiple views together results in a more generic performance of any learning model. The whole idea behind multiview learning is to fuse the features from multiple aspects of a single information in such a manner which along with capturing the information from the individual datasets, also captures the information regarding the correlation between those multiple datasets.
The following paper is based on OCCA (Orthogonal Canonical Correlation Analysis), a variant of CCA(Canonical correlation Analysis) for fusion of information from multiple views.[1]

1.2 Canonical Correlation Analysis

Canonical Correlation Analysis is a multivariate statistical analysis method for linear dimensionality reduction and feature fusion.It represents two dimensional views of a single instance as a linear representation of the individual correlations of each of these views and also interrelated correlation of the two datasets. Suppose, X and Y corresponds to two separate views of a single image, let say X corresponds to its front view and Y corresponds to left view, then CCA tries to measure the relationship among these two views in terms of the combine correlation of view X with X, view Y with Y and view X with Y and aims at retaining the maximum correlation between these two different views[1][2]. For establishing this constraint, that there is maximum correlation between the different views/aspects, a new set of orthogonal bases or projection directions,known as orthogonal canonical projection pairs are computed along which if the views are projected have maximum correlation associated between them. An important feature of these canonical covariates or projection pairs are orthogonal to one another and also the information which is captured by prior covariate pairs is not contained or represented by the later projection pairs. These canonical projection pairs form a basis to represent the original aspects in a reduced dimensional space where maximum correlation is associated between each of these views. The whole process of feature fusion through CCA could be defined in three steps:

- Feature Extraction: Extract two different set of feature vectors from the original views.
- Canonical Projection Pair: Based on the extracted features, extract canonical projection pairs such that which maximizes the correlation between these views.
- Feature Fusion: Projecting the original features onto these canonical pairs and fusing them either through serial or parallel fusion.[3]

1.3. Orthogonal Canonical Correlation Analysis:

Considering the traditional method or classical CCA, the generated canonical covariates are independent of any transformation and are less sensitive to data distribution and noises in the data. These inherit features of the canonical projection pairs due to the orthogonality of the covariates. The canonical covariates obtained from CCA are ensured to follow conjugate orthogonality property,i.e., pairwise orthogonality of each of the covariates, as they are obtained from the eigenvalue decomposition of maximization condition of the canonical correlation coefficient of the two views subject to the constraints that the maximum correlation is contained between each pair. Since the projection pairs are a result of the eigenvalue decomposition, so by default they will be orthogonal in nature.

But if the same maximization method is applied on a view set with a small number of sata samples, then the canonical covariates obtained from the eigen decomposition tends to violate the conjugated orthogonality property. This problem is known as the Small Sample Size (SSS) problem. The reason of this violation of orthogonality in a small sample data adheres to the fact that since the formulation of the covariance matrix either between the same view or different, captures the information of patterns or relation between those two data and if in a similar manner covariance matrix is computed between such views having less number of data samples, then the variance or the relation between the various features of those dataset cannot be truly captured by the covariance matrix. In simple terms, the covariance matrix cannot represent all the variance between the features. So if eigen decomposition is done on such such a covariance matrix, then the eigenvectors obtained do not corresponds to the true direction of maximum variance and are not necessarily independent of each other, resulting in non-orthogonal projection vectors which are affected by various data transformations and noises. Therefore in SSS problem cases, the projection vectors or canonical covariates obtained are not optimal.

In contrast to classical CCA, OCCA(Orthogonal Canonical Correlation Analysis) always produces orthogonal projection pairs by imposing an additional orthogonality constraint on the maximization function of the canonical correlation coefficient along with the prior ones.
This variant of classical CCA was first proposed by Xiao-Bo Shen, Quan-Sen Sun, Yun-Hao Yuan in 2013[1]. Similar to classic CCA, OCCA is also a linear multivariate statistical analysis and dimensionality reduction technique with the same maximization condition as the CCA and imposed an additional constraint to ensure that all the new generated canonical covariates are perpendicular to the older ones and thus ensuring conjugated orthogonality property.
Instead of postprocessing and orthogonalizing the canonical covariates obtained from the CCA, the OCCA tries to maintain this pairwise orthogonality constraint by performing twin eigen decomposition of the optimization condition instead of a single step of normal eigenvalue decomposition. Similar to the classical CCA where the maximum number of projection pairs or bases that could be generated are less than or equal to the minimum of the rank of the two views. Orthogonality constraint is implemented by checking the orthogonality of a current projection pair to the older ones. This means that for the first canonical projection pair there will

be no orthogonality check and for the *ith* the orthogonality will be checked with each of the *jth* pair, where *j* varies from *{0,1,2,.....,i-1}.*

The whole process of Orthogonal Canonical Correlation could be summarized below in the form of 2 models[3]:

- Model 1: Finding the first canonical projection pair from the classical CCA approach by just imposing maximum correlation constraint on the canonical correlation coefficient maximization function.

- Model 2: Iteratively finding the remaining canonical projection pair by ensuring orthogonality check with all the previously generated canonical vectors.

A drawback of OCCA is that in most of the cases that the generated eigenvectors for large eigenvalues are complex in nature.

Notations:

Before moving ahead with the mathematical formulation of the, below are the notations which will be carried on throughout this report:

X , Y	2 different views of the same information whose multivariate analysis has to be made both containing *n* number of samples in each of them.
p	Number of features in X view such that X is *n x p*
q	Number of features in Y view such that Y is *n x q*
α	*p* dimensional projection direction/canonical variate of X view
β	*q* dimensional projection direction/canonical variate of X view
Sxx	Covariance matrix of X and X
Syy	Covariance matrix of Y and Y
Sxy	Covariance matrix of X and Y
$\mathcal{P}$	Canonical Correlation Coefficient
Wx, Wy	Orthogonal canonical projection matrix of X and Y

Chapter 2

Mathematical Formulation

2.1 Formulation

The OCCA is aimed at finding a set of projection vectors or directions for each of the views such that the correlation between projections of each of the views on these vectors maximizes.
Let X $\varepsilon R^{n x p}$ and Y $\varepsilon R^{n x q}$ be the two different views of the same information each with p and q features and having n samples in each. Let α and β be the respective projection vectors or canonical variate pairs for each of the view such that projection of these views Z1 = X.α and Z2 = X.β on a common latent space have maximum correlation between them. The set of such projection vectors α and β for each of the views X and Y is known as canonical correlation projection matrix Wx and Wy each of which are represented as[3]:

$$Wx = \{ \alpha_1 , \alpha_2 , \alpha_3 , \ldots\ldots , \alpha_{k-2} , \alpha_{k-1} , \alpha_k \} \quad (2.1)$$

$$Wy = \{ \beta_1 , \beta_2 , \beta_3 , \ldots\ldots , \beta_{k-2} , \beta_{k-1} , \beta_k \} \quad (2.2)$$

where , *k* is the number of canonical covariate pairs that are needed to be extracted.

Let, Sxx, Syy and Sxy be the respective covariance matrices given by[3]:

$$S_{xy} = X X^T /N \quad (2.3) \qquad S_{xy} = X Y^T /N \quad (2.4) \qquad S_{xyy} = Y Y^T /N \quad (2.5)$$

Each of the dimensions (p x p), (q x q) and (p x q) respectively.
The following maximization problem defines the OCCA objective functions[3]:

$$\arg\max_{\alpha,\beta} \rho = \frac{\alpha^T S_{xy} \beta}{\sqrt{\alpha^T S_{xx} \alpha} \times \sqrt{\beta^T S_{yy} \beta}} \quad (2.6)$$

The set of α and β that maximizes the above equation corresponds to Wx and Wy. The maximization of above equation in two steps[3]:

- Obtaining the first canonical pair, similar to CCA, by maximizing the above optimization problem in (2.6) subject to the constraint of maximum correlation between the two views.

$$\text{model 1} \quad \begin{cases} \max \rho(\alpha, \beta) \\ \alpha^T S_{xx} \alpha = 1, \beta^T S_{yy} \beta = 1 \\ \alpha \in \Re^p, \beta \in \Re^q \end{cases} \tag{2.7}$$

- For the remaining canonical pair,2 to k, the could be found iteratively by maximizing the (2.6) subject to constraint of maximum correlation and orthogonality.

$$\text{model 2} \quad \begin{cases} \max \rho(\alpha, \beta) \\ \alpha^T S_{xx} \alpha = 1, \beta^T S_{yy} \beta = 1 \\ \alpha^T \alpha_j = 0, \beta^T \beta_j = 0 \, (j = 1, 2, \cdots k-1) \\ \alpha \in \Re^p, \beta \in \Re^q \end{cases} \tag{2.8}$$

Model 1 could easily be solved using lagrange duality and reducing to a standard eigen decomposition where (α, β).T will be the desired canonical pair[4].

Maximization of model 2 for 2 to kth is reduced to just find the largest eigenvalue and its corresponding vector of the below eigenvalue problem at each iteration[3]

$$\begin{pmatrix} 0 & (I - G_x) H_x \\ (I - G_y) H_y & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \lambda \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

where

$$\alpha^{(k-1)} = (\alpha_1, \alpha_2, \cdots, \alpha_{k-1})$$
$$\beta^{(k-1)} = (\beta_1, \beta_2, \cdots, \beta_{k-1})$$
$$H_x = S_{xx}^{-1} S_{xy}$$
$$G_x = S_{xx}^{-1} \alpha^{(k-1)} \left(\left[\alpha^{(k-1)} \right]^T S_{xx}^{-1} \alpha^{(k-1)} \right)^{-1} \left[\left(\alpha^{(k-1)} \right) \right]^T$$
$$H_y = S_{yy}^{-1} S_{yx}$$
$$G_y = S_{yy}^{-1} \beta^{(k-1)} \left(\left[\beta^{(k-1)} \right]^T S_{yy}^{-1} \beta^{(k-1)} \right)^{-1} \left[\left(\beta^{(k-1)} \right) \right]^T \tag{2.9}$$

So, the transformed views is obtained by, Z1=X.Wx and Z2=Y.Wy where Wx and Wy are computed using (2.1) and (2.2) after obtaining α and β from the above optimization function.

2.2 Universe Subspace

OCCA is a linear statistical analysis method which tries to compute a latent space(or, common subspace) on which if the views are projected have maximum correlation among them.
This common subspace/latent space between these views can be computed using either serial feature fusion or parallel feature fusion[1][2][3].

- Serial Feature Fusion:
  The fused feature can be computed as:

  $$Z = (X.Wx, Y.Wy).T$$

  In general for any m number of feature sets, the fused feature will be given by:

  $$Z = (X1.Wx1, X2.Wx2, \ldots\ldots, Xm\text{-}1.Wxm\text{-}1, Xm.Wxm).T$$

  where Z is the fused feature vector of dimension mk for each sample where k is the number of canonical covariates extracted from OCCA.

- Parallel Feature Fusion:
  The fused feature can be computed as:

  $$Z = X.Wx + Y.Wy$$

  In general for any m number of feature sets, the fused feature will be given by:

  $$Z = \sum Xi.Wxi \quad \text{for i from1 to m}$$

  where Z is the fused feature vector of dimension k for each sample, where k is the number of canonical covariates extracted from OCCA.

Among these 2 approaches for feature fusion, the serial fusion has better accuracy when a classifier is trained on it as instead of adding just 2 feature sets, it concat them and thus it retains more discriminant information in it about the fused feature sets.

Chapter 3

Algorithm

3.1 OCCA Algorithm:

Input: Multiview training datasets X and Y, each of dimensions (n x p) and (n x q) where p and q are their respective feature dimensionalities and n is number of samples in each of them.
Output: Wx and Wy

---

1. Construct covariance matrices Sxx, Sxy and Syy as defined in (2.3), (2.4) and (2.5)
2. Construct the largest eigenvalue problem matrix P using lagrange multiplier which is given by P= $Sxx^{-1/2} Sxy\ Syy^{-1/2}$
3. Perform SVD on matrix *P: P=UDV.T*
4. Obtain the first canonical projection pair given by:
   4.1: α_1 = $Sxx^{-1/2}$ U[:,0]
   4.2: $_1$ = $Syy^{-1/2}$ V[:,0]
5. For the ith canonical projection pair from 2 to k follow the below mentioned steps:
   5.1 Wx = { α_1 , α_2 , α_3 , ... , α_{i-2} , α_{i-1} , α_i } (2.10)
   5.2 Wy = { $_1$, $_2$, $_3$, ... , $_{i-2}$, $_{i-1}$, $_i$ } (2.11)
   5.3. Compute Gx, Gy, Hx and Hy using (2.9), (2.10) and (2.11)
   5.4. Compute $A = (I - Gx)Hx$ and $B = (I - Gy)Hy$.

   Solving the largest eigenvalue problem in (2.9) in the form of a simultaneous pair of equations.
   5.5. Find eigenvalue decomposition of $\sqrt{}\ A.B$: $\sqrt{}A.B = Ualpha.D.V\,alpha.T$
   5.6. Find eigenvalue decomposition of $\sqrt{}\ B.A$: $\sqrt{}B.A = Ubeta.D.V\,beta.T$
   5.7 Eigenvector $alpha$ for the ith pair is computed by: $V\ alpha[:,0]^2$
   5.7 Eigenvector $beta$ for the ith pair is computed by: $V\ beta[:,0]^2$
   5.8. Compute ith canonical projection pair using:
   Wx_i = $Sxx^{-1/2}\ alpha$
   Wyi = $Syy^{-1/2}\ beta$
   5.9 Append these canonical projection pairs to Wx and Wy respectively.
   If i is less than k go back to step 5.

Chapter 4

Documentation of API

4.1 Package Organization:

class OCCA(n_*components=2, normalize=True, complex*_=False)
Parameters:

- **n_components:** int, (default=2)
  number of components to keep,maximum allowed value is equal to min of rank of input views
- **normalize:** boolean, (default=2)
  whether to normalize the data or not? If set to true it will transform data to mean 0 and standard deviation 1.
- **complex_:** boolean, (default=False)
  whether to allow complex values or not? If set to false then all the negative values while taking square root are converted to 0. This parameter is required to be True is the input views contain signals or require Fourier Transform.If set to True, most of the largest eigenvalue canonical pair generated by OCCA will be complex to establish conjugated orthogonality rule

Attributes:

p-->corresponds to the feature dimension of first view

q-->corresponds to the feature dimension of second view

n-->corresponds to the number of samples in X and Y

- **x_*weights*_**: array, [p, n_components]
  X canonical covariate or projection directions in decreasing order of eigenvalues.
- **y_*weights*_**: array, [q, n_components]
  Y canonical covariate or projection directions in decreasing order of eigenvalues.
- **eigval_**: array, [1,n_components]
  1D array that stores the eigenvalues of canonical covariate pairs in descending order.
- **x_*transform*_**: array, [n, n_components]
  2D array that stores the transformed X view by applying the fitted model on it.
- **y_*transform*_**: array, [n, n_components]
  2D array that stores the transformed Y view by applying the fitted model on it.
- **covariance_xx**: array, [n_components, n_components]
  2D array that stores the covariance of the transformed X view with itself.

- **covariance_xy**: array, [n_components, n_components]
  2D array that stores the covariance of the transformed X view with transformed Y view. It is defined only when 2 views are passed for transformation.
- **covariance_xx**: array, [n_components, n_components]
  2D array that stores the covariance of the transformed Y view with itself.It is defined only when alongwith the X view Y view is also passed for transformation.

4.2. Methods:

- **fit(X,Y)**:
  Fit the OCCA model to the passed data
- **transform(X,[,Y])**:
  Apply the learned model/ dimensionality reduction to the trained data.
- **fit_transform(X,Y)**:
  Fit the OCCA model on the passed data arguments and apply learned dimensionality reduction on the passed data(on which model is learned)

__init__(n_components=2, normalize=True, complex_=False)
Initialize self for OCCA class and set parameters for instantiating objects.

fit(X,Y): Fit the model to the data
Parameters: X: array-like of shape (n_samples,n_features)

Training vector to learn the model where n_samples is number of data samples and n_features is dimensionality of each sample.

Y: array-like of shape (n_samples,n_features1)

Training vector(in case of OCCA) or target vector where n_samples is number of data samples and n_features1 id dimensionality of each sample.

For OCCA the number of samples for both X and Y must be the same.

fit_transform(X,Y): Fit the model to the data and transform the fitted data on the basis of the learned model.
Parameters: X: array-like of shape (n_samples,n_features)

Training vector to learn the model where n_samples is number of data samples and n_features is dimensionality of each sample.

Y: array-like of shape (n_samples,n_features1)

Training vector(in case of OCCA) or target vector where n_samples is number of data samples and n_features1 id dimensionality of each sample.

For OCCA the number of samples for both X and Y must be the same.

transform(X,Y=None): Apply the learned model to the data and transform it.
Parameters: X: array-like of shape (n_samples,n_features)

Training vector to learn the model where n_samples is number of data samples and n_features is dimensionality of each sample.

Y: array-like of shape (n_samples,n_features1)

Training vector(in case of OCCA) or target vector where n_samples is number of data samples and n_features1 id dimensionality of each sample.

For OCCA the number of samples for both X and Y may vary in case of transform function. For transforming any other matrix on which the model is not learned/fitted, the feature dimension of X and feature dimension of Y(if passed) must be equal to feature dimensions of the data on which model is fitted respectively.

Chapter 5

Examples

5.1. Example 1:

```
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]]

occa=OCCA()

occa.fit(X,Y)

X_c,Y_c=occa.transform(X,Y)
```

5.2 Example 2:

```
X = [[0.1, -0.2], [0.9, 1.1], [6.2, 5.9], [11.9, 12.3]]

Y = [[0.1, -0.9], [0.9, 8.1], [6.2, 5.9], [11.9, 12.3]]

occa=OCCA(complex_=False)

X_c,Y_c=occa.fit_transform(X,Y)
```

Chapter 6:

Learning Outcomes:

- Ability to analyze, understand and interpret high scientific and mathematical concepts and implement those concepts from scratch.

- Ability to analyze and recognize patterns from the raw data and from the different aspects of the same information at a shared subspace level and retrieve relative insights from those different aspects.

- Ability to merge multiple views of the same data on a common or shared basis so that a better understanding, analysis and evaluation of information can be made.

- Understanding the concepts of latent space and how Canonical Correlation Analysis (CCA) could be used to get maximum insights from different views of the same data by considering them on a shared latent space and eliminating the drawbacks of the process with the use of OCCA (Orthogonal Canonical Correlation Analysis) as a improvement over the latter.

Appendix A

References

[1] Guo Chenfeng and Wu Dongrui.: Canonical correlation analysis (cca) based multi-view learning: An overview, arXiv preprint arXiv:1907.01693, July 2019.doi:abs/1907.01693.

[2] Jing Zhao, Xijiong Xie, Xin Xu, and Shiliang Sun.: Multiview learning overview: Recent progress and new challenges, Information Fusion, vol. 38, pp. 43-54, Nov. 2017. doi:10. 1016/j.inffus.2017.02.007

[3] X. Shen, Q. Sun, and Y. Yuan, Orthogonal canonical correlation analysis and its application in
feature fusion, in Proceedings of the 16th International Conference on Information Fusion, IEEE, Istanbul, 2013, pp. 151–157.

[4] Li Wang, Lei-hong Zhang, Zhojun Bai and Ren-Cang Li. Orthogonal Canonical Correlation Analysis and its Applications.,2020

[5]H. Hotelling, "Relations between two sets of variates," Biometrika,vol. 28, pp. 321-377, 1936.

[6] Z. Wen and W. Yin, A feasible method for optimization with orthogonality constraints, Math. Program. 142(1-2) (2013), pp. 397–434.

[7] Y. Fu, L. Cao, G. Guo, and T. S. Huang, "Multiple feature fusion by subspace learning," in Proceedings of the 2008 international conference on Content-based image and video retrieval, 2008, pp.127-134.

[8] Rajendra Prasad Regmi: Finding of principle square root of a real number by using interpolation
Method, Janapriya Journal of Interdisciplinary Research, Vol. VII, 2018