

Canonical Correlation Analysis With $L_{2,1}$ -Norm for Multiview Data Representation

Gunjan Pandey (0801CS171025)
gs0801cs171025@sgsitsindore.in
December 13, 2020

Contents

1. Introduction	2
1.1 Multiview Learning	2
1.2 Multivariate Techniques	2
1.3 Limitations of CCA	3
2. Mathematical Formulation	4
2.1 Multiview Learning	4
2.2 $L_{2,1}$ CCA	4
3. Algorithm	6
3.1 Notation Definitions	6
3.2 $L_{2,1}$ CCA algorithm	7
4. Documentation of API	8
4.1 Package organization	8
4.2 Methods	9
5. Examples	11
5.1 Example 1	11
5.2 Example 2	12
6. Learning Outcome	13
6.1 Multiview Learning	13
6.2 $L_{2,1}$ CCA	13
A. References	14

Chapter 1

Introduction

1.1 Multiview Learning

Multi-view learning in machine learning considers learning with multiple-views to improve the generalization performance. Many data are often collected from different measuring methods as particular single-view data cannot comprehensively describe the information of all examples. The so-called multi-view data, in the community of machine learning, are technically referred to as the data collected or generated from different domains, sources, media, or diverse extractors.

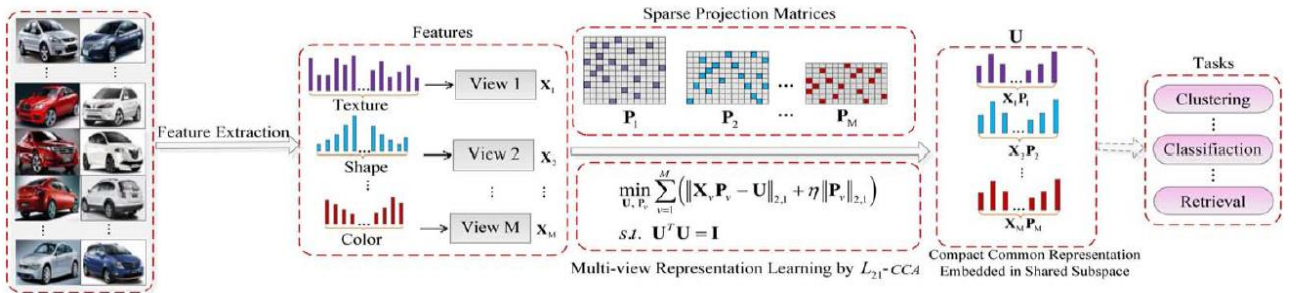


Fig. 1. Framework of L2,1-CCA for multi-view representation learning. $X_v \in \mathbb{R}^{(N \times d_v)}$ ($v = 1, \dots, M$) denote M views of the object, U is the common representation learned from multiple X_v 's by the proposed L2,1-CCA.

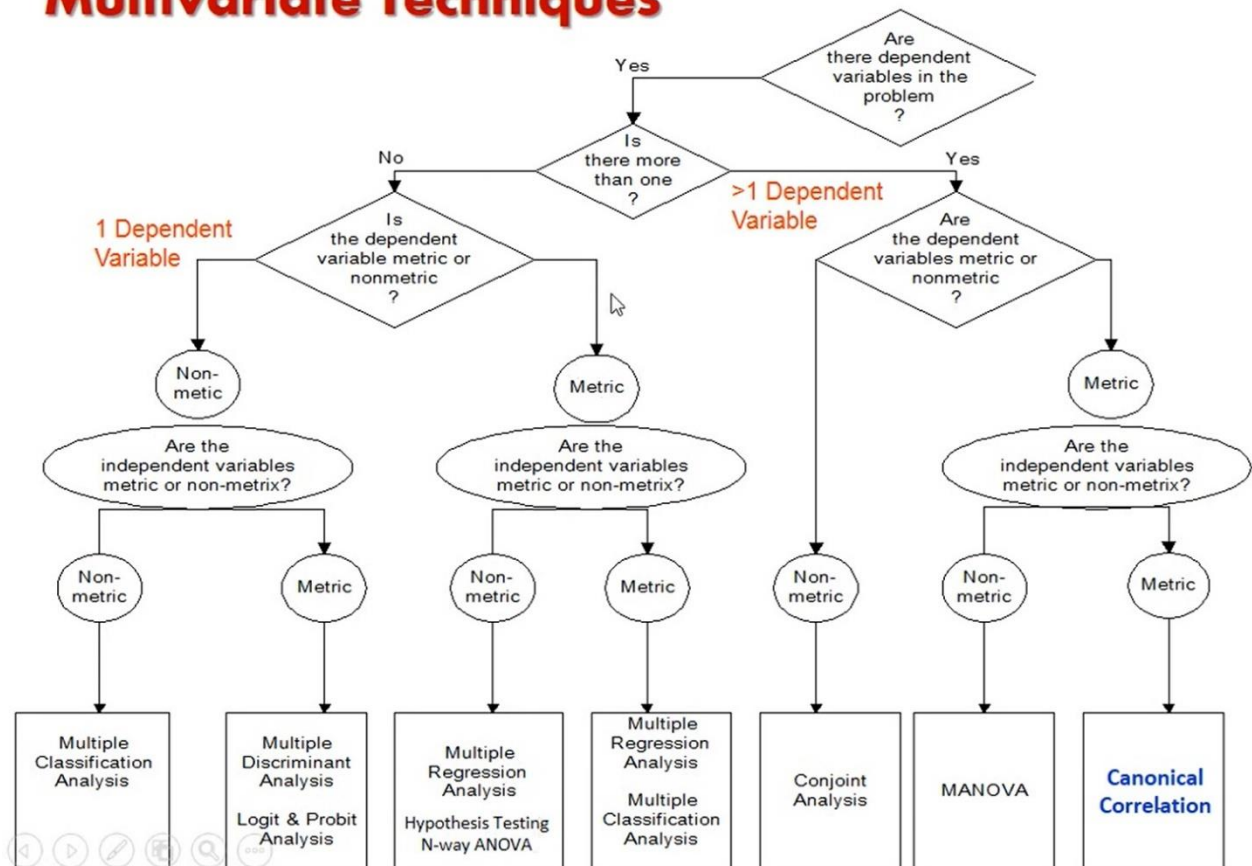
Taking the object car as an example, we can extract its heterogeneous multiple types of feature like shape, texture, color, etc. Naturally, each type of such heterogeneous features can be regarded as a view X_v . Then for the multiple views of representations, in virtue of the power of different sparse projection matrices, they can be projected into the shared subspace by L2,1-CCA. Using the learned compact common representation U , some tasks such as clustering, classification, and retrieval can be implemented.

1.2 Multivariate Techniques

Usually, multiple views associated to data objects possess complementary and coherent knowledge to each other, and multi-view learning is competent to make good use of such knowledge to potentially learn more expressive representation than single-view learning, which has been shown through three mainstream techniques, that is, co-training, multi-kernel learning, and shared subspace-based representation learning. Following either the consensus principal or the

complementary principal, a large number of multi-view representation learning algorithms have been developed, of which the most popular ones include canonical correlation analysis (CCA), bilinear model (BLM), multi-output regularized feature projection (MORP), partial least square (PLS), and deep neural networks-based (DNNs-based) approaches.

Multivariate Techniques



1.3 Limitations of CCA

First, most of the techniques are merely applicable to the case of two views, while in most cases multi-view data are associated to three or more views. Second, for multi-view data, each individual view may be high-dimensional and redundant. In other words, not all variables in the original variable set are identically informative for interpreting the canonical variables. In this case, if the learned canonical loadings are not potentially sparse or selective, it will be difficult to obtain a consistent common representation with promising discriminability. Third, they are unable to capture deep-level correlations between multi-view data.

Chapter 2

Mathematical Formulation

2.1 Formulation

To hopefully incorporate complementary and coherent information from multiple views, this report mainly focuses on exploring the power of correlation analysis for discovering a discriminative common representation for multi-view data.

$$\max_{\mathbf{p}_1, \mathbf{p}_2} \rho = \frac{\mathbf{p}_1^T \mathbf{X}_1^T \mathbf{X}_2 \mathbf{p}_2}{\sqrt{(\mathbf{p}_1^T \mathbf{X}_1^T \mathbf{X}_1 \mathbf{p}_1)(\mathbf{p}_2^T \mathbf{X}_2^T \mathbf{X}_2 \mathbf{p}_2)}}$$

$$\begin{aligned} \min_{\mathbf{P}_1, \mathbf{P}_2} & \|\mathbf{X}_1 \mathbf{P}_1 - \mathbf{X}_2 \mathbf{P}_2\|_F^2 \\ \text{s.t. } & \mathbf{P}_1^T \mathbf{X}_1^T \mathbf{X}_1 \mathbf{P}_1 = \mathbf{I}, \mathbf{P}_2^T \mathbf{X}_2^T \mathbf{X}_2 \mathbf{P}_2 = \mathbf{I} \end{aligned}$$

$$\mathbf{U} = \frac{\mathbf{X}_1 \mathbf{P}_1 + \mathbf{X}_2 \mathbf{P}_2}{2}$$

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{P}_v} & \sum_{v=1}^M \|\mathbf{U} - \mathbf{X}_v \mathbf{P}_v\|_F^2 \\ \text{s.t. } & \mathbf{U}^T \mathbf{U} = \mathbf{I} \end{aligned}$$

2.2 $L_{2,1}$ -CCA

The proposed $L_{2,1}$ -CCA model in this paper aims at discovering a discriminative common representation for multi-view data objects by incorporating more useful information from their associated multiple views with correlation analysis. To target this goal, twofold issues are considered, that is, how to facilitate the interpretability of the learned canonical variables and how to keep the learned

canonical common representation highly consistent with the canonical variables from each view of the data. To address these two issues, in $L_{2,1}$ -CCA, we impose the $L_{2,1}$ -norm constraint on the canonical loadings as a regularization term, which is inspired by the recent sparsity induced learning algorithms, and simultaneously we use the rotational invariant L_1 -norm to measure the correlation loss term

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{P}_v} & \sum_{v=1}^M \left(\underbrace{\sum_{i=1}^N \|\mathbf{X}_v^i \mathbf{P}_v - \mathbf{U}^{\cdots i}\|_2}_{\text{CorrelationLossTerm}} + \underbrace{\eta \|\mathbf{P}_v\|_{2,1}}_{\text{RegularizationTerm}} \right) \\ \text{s.t. } & \mathbf{U}^T \mathbf{U} = \mathbf{I}_d \\ & \mathbf{Q}_v = \mathbf{D}_v^{-1} \mathbf{A}_v^T (\mathbf{A}_v \mathbf{D}_v^{-1} \mathbf{A}_v^T)^{-1} \end{aligned}$$

$$\begin{aligned} \min_{\mathbf{U}} & \sum_{v=1}^M \text{tr}(\mathbf{U}^T \mathbf{Q}_v^T \mathbf{D}_v \mathbf{Q}_v \mathbf{U}) \\ \text{s.t. } & \mathbf{U}^T \mathbf{U} = \mathbf{I}_d. \end{aligned} \quad (14)$$

Clearly, when \mathbf{D}_v is fixed, the minimization optimization given by (14) is a classical and well-studied Stiefel manifold problem [43], [44]. Let $\mathbf{B} = \sum_{v=1}^M (\mathbf{Q}_v^T \mathbf{D}_v \mathbf{Q}_v)$, then $\mathbf{U} \in \mathbb{R}^{N \times d}$ is composed of d eigenvectors corresponding to the d smallest eigenvalues of \mathbf{B} .

To be better understood, the detailed procedure for solving the proposed $\ell_{2,1}$ -norm constrained CCA model is summarized as Algorithm 1.

C. Convergence Analysis

To show the convergence behavior of $L_{2,1}$ -CCA, we summarize it as Theorem 1.

Theorem 1: The objective function $\Phi(\mathbf{F}_v) = \sum_{v=1}^M \|\mathbf{F}_v\|_{2,1}$ in (9) is nonincreasing when iteratively optimizing \mathbf{U} and \mathbf{F}_v by the updating rules in Algorithm 1.

Chapter 3

Algorithm

3.1 Notation Definitions

A. Notation Definitions

Throughout this paper, $\mathbf{A}^{\cdot i}$ and \mathbf{A}^j are used to represent the i th row and j th column of the matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$, respectively. For the vector $\mathbf{a} \in \mathbb{R}^n$, its ℓ_p -norm is defined as $\|\mathbf{a}\|_p = (\sum_{i=1}^n |a_i|^p)^{(1/p)}$. $\|\mathbf{A}\|_F$ and $\|\mathbf{A}\|_{2,1}$ represent the Frobenius norm and $\ell_{2,1}$ -norm of the matrix \mathbf{A} , which are defined as $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m \mathbf{A}_{i,j}^2}$ and $\|\mathbf{A}\|_{2,1} = \sum_{i=1}^n \sqrt{\sum_{j=1}^m \mathbf{A}_{i,j}^2} = \sum_{i=1}^n \|\mathbf{A}^{\cdot i}\|_2$, respectively. We denote $\text{tr}(\cdot)$ as the trace of a symmetric matrix and \mathbf{I}_k for the $k \times k$ identity matrix. In the scenario of multiview representation learning, we shall refer to the data matrices $\mathbf{X}_v \in \mathbb{R}^{N \times d_v}$, with $v = 1, \dots, M$, as observations from M views, and each row $\mathbf{X}_v^{\cdot i} \in \mathbb{R}^{1 \times d_v}$ of these observation matrices, $i = 1, \dots, N$, corresponds to the same object or data and the columns of the observation matrix \mathbf{X}_v correspond to the attributes from the v th view. Also, lower-case letters and uppercase letters in boldface denote column vectors and matrices, respectively.

3.2 $L_{2,1}$ CCA algorithm

Algorithm 1 $L_{2,1}$ -CCA Algorithm

Input:

Multi-view data $\mathbf{X}_v \in \mathbb{R}^{N \times d_v}$, $v = 1, \dots, M$; the dimensionality d of the common representation; the regularization parameter η ; the maximum iteration numbers $maxIter_1$ and $maxIter_2$;

Output:

The common representation $\mathbf{U} \in \mathbb{R}^{N \times d}$;

- 1: **Preprocessing:** $\mathbf{A}_v = \begin{bmatrix} \frac{1}{\eta} \mathbf{X}_v & -\mathbf{I} \end{bmatrix}$;
 - 2: **Initializing:** $\mathbf{D}_v^{(0)} = \mathbf{I}$, $v = 1, \dots, M$;
 - 3: Compute $\mathbf{U}^{(0)}$ by solving the eigen-decomposition of $\mathbf{B} = \sum_{v=1}^M (\mathbf{Q}_v^T \mathbf{D}_v^{(0)} \mathbf{Q}_v)$
 - 4: **for** $p = 1 : maxIter_1$ **do**
 - 5: **for** $q = 1 : maxIter_2$ **do**
 - 6: Compute $\mathbf{F}_v^{(q)}$ according to Eq. (12):

$$\mathbf{F}_v^{(q)} = \left(\mathbf{D}_v^{(q-1)} \right)^{-1} \mathbf{A}_v^T \left(\mathbf{A}_v \left(\mathbf{D}_v^{(q-1)} \right)^{-1} \mathbf{A}_v^T \right)^{-1} \mathbf{U}^{(p-1)}$$
 - 7: Update $\mathbf{D}_v^{(q)}$ by virtue of $\mathbf{D}_v^{(q)} = \text{diag} \left(\frac{1}{2 \|\mathbf{F}_v^{(q)}\|_2} \right)$;
 - 8: **if** \mathbf{F}_v converge **then**
 - 9: break;
 - 10: **end if**
 - 11: **end for**
 - 12: Compute $\mathbf{U}^{(p)}$ according to Eq. (11) by seeking the d eigen-vectors corresponding to the d smallest eigen-values of matrix $\mathbf{B} = \sum_{v=1}^M (\mathbf{Q}_v^T \mathbf{D}_v \mathbf{Q}_v)$;
 - 13: **if** the objective function converges **then**
 - 14: break;
 - 15: **end if**
 - 16: **end for**
-

Chapter 4

Documentation of API

4.1 Package Organization

Required Module: NumPy

Package Name: l21cca

4.2 Methods

Methods:

`inverse(A)`

Parameters

A: 2D Matrix

Returns

B: Inverse of a matrix

`eigen_decom(A)`

Parameters

A: 2D Matrix

Returns

w(..., M) array : The eigenvalues, each repeated according to its multiplicity.

v(..., M, M) array
The normalized (unit "length") eigenvectors, such that the column `v[:,i]` is the eigenvector corresponding to the eigenvalue `w[i]`.

`norml21(A)`

Parameters

A: 2D Matrix

Returns

l: Column wise Frobenius normalization of a matrix

`converge(A, B) :`

Parameters

A: 3D Matrix

B: 3D Matrix

Returns

l: Boolean value, representing whether matrices are same or not.

l2l_cca(R, d=1, neta=1, maxIter1=1, maxIter2=1)

Parameters:

R: 3D array

R is a 3D list where each matrix in R represents observations from different view

Column of each matrix represents attributes of vth view and Row of each matrix represents different objects.

d: int

Dimension of the common representation

neta: float

Regularization parameter

maxIter1: int

Maximum iteration number

maxIter2: int

Maximum iteration number

Returns

U: Common representation in d dimension

Chapter 5

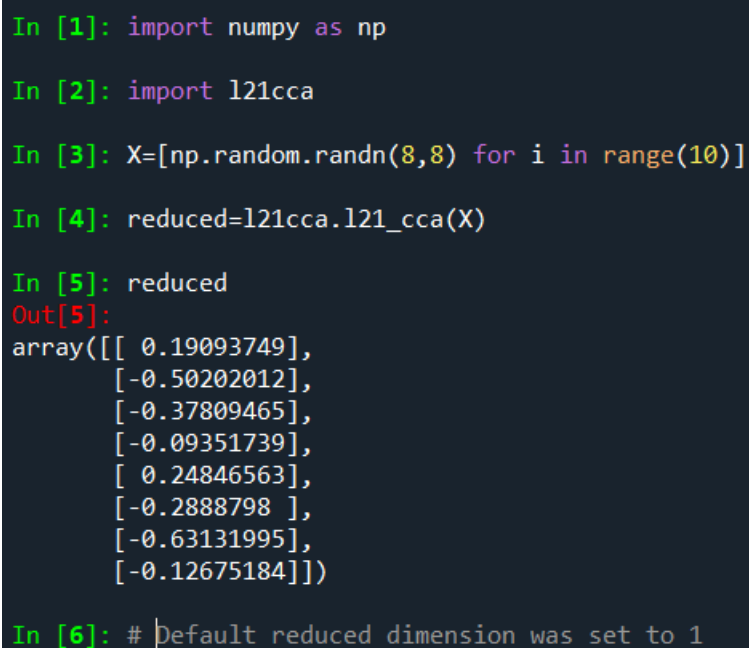
Examples

5.1 Example 1

Code:

```
import numpy as np
import l21cca
X=[np.random.randn(8,8) for i in range(10)]
reduced=l21cca.l21_cca(X)
```

Screenshot:



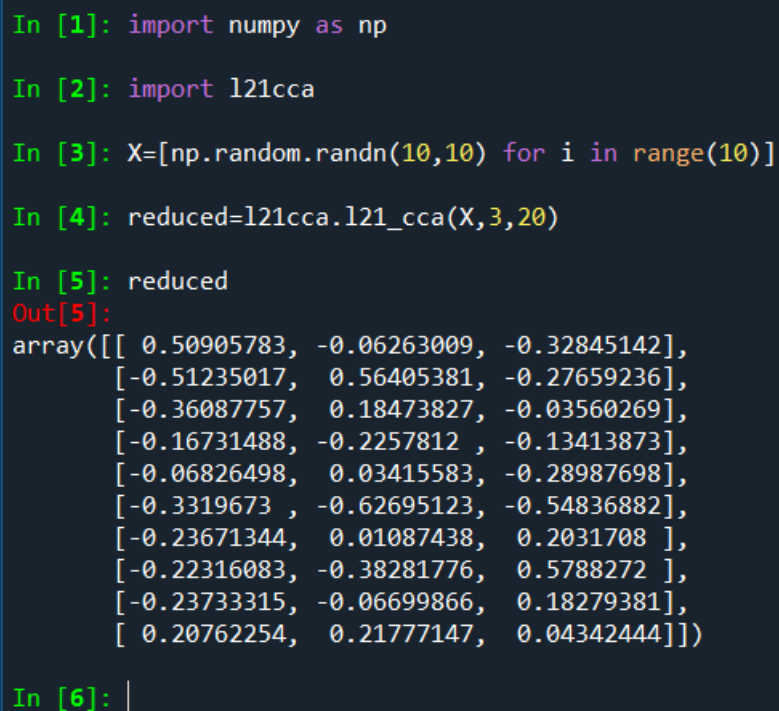
```
In [1]: import numpy as np
In [2]: import l21cca
In [3]: X=[np.random.randn(8,8) for i in range(10)]
In [4]: reduced=l21cca.l21_cca(X)
In [5]: reduced
Out[5]:
array([[ 0.19093749],
       [-0.50202012],
       [-0.37809465],
       [-0.09351739],
       [ 0.24846563],
       [-0.2888798 ],
       [-0.63131995],
       [-0.12675184]])
In [6]: # Default reduced dimension was set to 1
```

5.2 Example 2

Code:

```
import numpy as np
import l21cca
X=[np.random.randn(10,10) for i in range(10)]
reduced=l21cca.l21_cca(X,3,20)
```

Screenshot:



```
In [1]: import numpy as np
In [2]: import l21cca
In [3]: X=[np.random.randn(10,10) for i in range(10)]
In [4]: reduced=l21cca.l21_cca(X,3,20)
In [5]: reduced
Out[5]:
array([[ 0.50905783, -0.06263009, -0.32845142],
       [-0.51235017,  0.56405381, -0.27659236],
       [-0.36087757,  0.18473827, -0.03560269],
       [-0.16731488, -0.2257812 , -0.13413873],
       [-0.06826498,  0.03415583, -0.28987698],
       [-0.3319673 , -0.62695123, -0.54836882],
       [-0.23671344,  0.01087438,  0.2031708 ],
       [-0.22316083, -0.38281776,  0.5788272 ],
       [-0.23733315, -0.06699866,  0.18279381],
       [ 0.20762254,  0.21777147,  0.04342444]])
In [6]: |
```

Chapter 6

Learning Outcomes

- How to use multiple views associated to data objects?
- To well exploit the complementary and coherent information across multiple views
- How to learn a compact and consistent representation by aggregating the variables from each view.
- How to apply multi-view learning for more than two views?
- How to obtain a consistent common representation with promising Discriminability?

Appendix A

References

- [1] C. Xu, D. Tao, and C. Xu, “A survey on multi-view learning,” CoRR, vol. abs/1304.5634, 2013.
- [2] A. Blum and T. M. Mitchell, “Combining labeled and unlabeled data with co-training,” in Proc. Int. Conf. Learn. Theory, 1998, pp. 92–100.
- [3] A. Kumar and H. Daume, III, “A co-training approach for multi-view spectral clustering,” in Proc. Int. Conf. Mach. Learn. (ICML), 2011, pp. 393–400.
- [4] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, “Multiple kernel learning, conic duality, and the SMO algorithm,” in Proc. Int. Conf. Mach. Learn. (ICML), 2004, pp. 41–48.
- [5] B. McFee and G. R. G. Lanckriet, “Learning multi-modal similarity,” J. Mach. Learn. Res., vol. 12, no. 8, pp. 491–523, 2011.
- [6] N. Chen, J. Zhu, F. Sun, and E. P. Xing, “Large-margin predictive latent subspace learning for multiview data analysis,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 12, pp. 2365–2378, Dec. 2012.
- [7] Meixiang Xu, Zhenfeng Zhu, Xingxing Zhang, Yao Zhao, Senior Member, IEEE, and Xuelong Li, Fellow, IEEE Canonical Correlation Analysis With L2,1-Norm For Multiview Data Representation