Discriminative Extended Canonical Correlation Analysis

DE-CCA

Kuldeep Sharma

0801CS171036

Chapter 1

Introduction

Abstract

In this paper we have trouble matching sets of vectors embedded in the same input space. We propose a method driven by the analysis of canonical correlation (CCA), a mathematical process that has proven to be effective in a variety of pattern detection problems. As a CCA when used for match comparisons, our canonical correlation analysis (E-CCA) aims to produce very similar variance methods between the two sets. Our first major contribution is the formulation of a framework for the dynamic dynamics of these methods from data where there is uncertainty associated with sound and random sampling. The E-CCA maintains the efficiency and security of the CCA form, but in contrast, it does not have free limits that can be added directly from the data (data size, and the number of canonical combinations used for match setup). Our second major contribution is to show that unlike the CCA, the E-CCA is easily adapted to match sets in a discriminatory learning program we call the canonical correlation analysis (DE-CCA) analysis. The theoretical contributions of this paper are followed by an examination of the strength of its structures with visual activity from the sets of visual images. The results show that our approach, E-CCA, has already surpassed both the CCA and its discriminatory CCA (C-CCA) counterparts, in all their free-range boundaries. The greatest improvement is found in the discriminatory variance, the DE-CCA.

1.1 Canonical Correlation Analysis (CCA)

Canonical Correlation Analysis (CCA) is a fundamental statistical technique for characterizing the linear relationships between two multidimensional variables. First introduced in 1936 by Hotelling, it has found numerous applications. For the machine learning community, more familiar applications include learning with privileged information, semi-supervised learning, monolingual and multilingual word representation learning, locality sensitive hashing and clustering. Because these applications involve unlabeled or partially labeled data, the amount of data available for analysis can be vast, motivating the need for scalable approaches.

1.2 Discriminative extended canonical component analysis (DE-CCA)

In CCA we always try to find latent Canonical Variants. The inevitable issues of cca were

1. it is example hard partitioning .

2. As the very first step of CCA is applying PCA and It was not efficient to determine all parameters efficiently.

DE-CCA relies upon discriminative learning framework.

Chapter 2

Mathematical Formulation

1.1 Mathematical Formulation

we will be performing space transformation using the covariance matrix.

$$\boldsymbol{\Upsilon}_X = (\boldsymbol{\Sigma}_X)^{1/2} = \mathbf{V}_X \begin{bmatrix} \sqrt{\lambda_X^{(1)}} & 0 & 0 & 0 \\ 0 & \sqrt{\lambda_X^{(2)}} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sqrt{\lambda_X^{(D)}} \end{bmatrix} \mathbf{V}_X{}^T = \mathbf{V}_X (\boldsymbol{\Lambda}_X)^{1/2} \mathbf{V}_X{}^T$$

Where, V_x is Eigen vector of Matrix $X.X^T$

$$\psi_1 = \max_{\hat{\mathbf{w}}_1} \left\{ (\boldsymbol{\Upsilon}_X \hat{\mathbf{w}}_1)^T (\boldsymbol{\Upsilon}_Y \hat{\mathbf{w}}_1) \right\} = \max_{\hat{\mathbf{w}}_1} \left\{ \hat{\mathbf{w}}_1^T \boldsymbol{\Upsilon}_X{}^T \boldsymbol{\Upsilon}_Y \hat{\mathbf{w}}_1 \right\}$$
$$= \max_{\hat{\mathbf{w}}_1} \left\{ \hat{\mathbf{w}}_1^T \boldsymbol{\Upsilon}_X{}^T \boldsymbol{\Upsilon}_Y \hat{\mathbf{w}}_1 \right\} = \max_{\hat{\mathbf{w}}_1} \left\{ \hat{\mathbf{w}}_1^T \boldsymbol{\Phi}_{XY} \hat{\mathbf{w}}_1 \right\},$$

We are replacing $\boldsymbol{\Upsilon}_X{}^T \boldsymbol{\Upsilon}_Y$ with $\boldsymbol{\Phi}_{XY}$

The Total degree of Agreement/Similarity will be

$$\hat{\mu}_{XY} = \frac{\sum_{i=1}^{D} \lambda_\phi^{(i)}}{\sum_{i=1}^{D} \sqrt{\lambda_X^{(i)} \lambda_Y^{(i)}}} = \frac{\mathrm{Tr}\ [\hat{\boldsymbol{\Phi}}_{XY}]}{\sum_{i=1}^{D} \sqrt{\lambda_X^{(i)} \lambda_Y^{(i)}}}$$

where $\lambda_\Phi^{(1)} \geq \lambda_\Phi^{(2)} \geq \ldots \lambda_\Phi^{(D)} \geq 0$ are the eigenvalues of $\boldsymbol{\Phi}_{XY}$.

$\lambda_X^{(i)} \lambda_Y^{(i)}$ are Eigen values of X.XT and Y.YT respectively

Now Finding Singular Vector Decomposition of res,

$$\hat{\mathbf{\Phi}}_{XY} = \mathbf{\Upsilon}_X \mathbf{P} \mathbf{\Upsilon}_Y$$

where,

$$\mathbf{P} = (\mathbf{\Sigma}_W)^{1/2} (\mathbf{\Sigma}_B)^{-1} (\mathbf{\Sigma}_W)^{1/2}$$

Chapter 3

Algorithm

3.1 Algorithm

Step 1->Find dot product of feature matrix **X** and $\mathbf{X^T}$ and named it as **matrix_x**.

Step 2->Find dot product of feature matrix **Y** and $\mathbf{Y^T}$, and named it as **matrix_x**.

Step 3-> Find Finding **Eigen values** and **Eigen Vectors** of both the matrices.

Step 4->Find **Deviation** for both the matrix and named as **dev_x** and **dev_y**.

Step 5-> Find product of deviation and named it as **res**.

Step 6-> Find **Degress of agreement** now.

Step 7-> Find **Signgular Vector Decomposition** of **res. SVD** will have best parameters of correlation, which will used for future reference.

Chapter 4

Documentation of API

4.1 Package organization

from DECCA import DescriminativeExtendedcca

model=DescriminativeExtendedcca()

Parameters:

X : ndarry, [n, p] Matrix of one feature space. Where n is no of sample and p is no of feature.

Y : ndarry, [n, q] Matrix of second feature space. Where n is no of sample and q is no of feature.

fit_tranform(X,Y):

Fit model to data.

Parameters:

X : ndarray, (n*p) where p is a feature and n is a number of samples.

Y : ndarray, (n*q) where q is a feature and n is a number of samples.

Return

It will return degree of agreement between two feature matrices

get_final_coeffcient()

It will get final DE-CCA Coefficients.

Return

It will return coefficients

Chapter 5

Example

5.1 Example

```
In [34]: from DECCA import DescriminativeExtendecca
         model=DescriminativeExtendecca()
```

```
In [33]: model.fit_tranform(a,b)
Out[33]: 7.77100890049928
```

```
In [32]: model.get_final_coeffcient()
Out[32]: (array([[-0.20027386,  0.42315011,  0.05119975, ...,  0.002039  ,
                  -0.04137332,  0.06569134],
                 [-0.39260532, -0.08396303,  0.59306166, ...,  0.00886796,
                   0.05208261, -0.06218481],
                 [-0.23742024,  0.51448906, -0.17624518, ...,  0.0105266 ,
                  -0.00216013, -0.01110619],
                 ...,
                 [-0.0902915 , -0.00349087, -0.04059546, ..., -0.0055114 ,
                  -0.18475038, -0.02222837],
                 [-0.09089729,  0.04439858, -0.01035985, ..., -0.20215883,
                  -0.15295671,  0.11204781],
                 [-0.08687654, -0.02341012, -0.06923417, ...,  0.10098122,
                  -0.01107934, -0.02622606]]),
          [None, None])
```

Chapter 6

Learning Outcomes

- Studied thoroughly CCA and multi-view analysis.
- Studied How to find most efficient canonical covariants from the set of views.
- Studied that accuracy get higher when we divide subspace into distcriminative and non discriminative.

Reference

Discriminative extended canonical correlation analysis for pattern set matching ***Discriminative extended canonical correlation analysis for pattern set matching | DeepAI***