Probabilistic Canonical Correlation Analysis

(PCCA)

Nisreen Sabir (0801CS171050)

Farida Fakhri (0801CS171024)

Contents

1. Introduction

1.1 Canonical Correlation Analysis (CCA):

Canonical correlation analysis is used to identify and measure the associations among two sets of variables. Canonical correlation is appropriate in the same situations where multiple regression would be, but where are there are multiple inter-correlated outcome variables. Canonical correlation analysis determines a set of canonical variates, orthogonal linear combinations of the variables within each set that best explain the variability both within and between sets.

Canonical correlation analysis (CCA), first proposed by Hotelling [9] in 1936, is a typical subspace learning approach. Its main idea is to find pairs of projections for different views so that the correlations between them are maximized. Since CCA takes the relationship between different feature sets into account, which is consistent with the idea of MVL, it is widely used in MVL [12]–[14], including multi-view dimensionality reduction [15], multi-view clustering [16], [17], multi-view regression [18], and so on.

1.2 Probabilistic Canonical Correlation Analysis (PCCA):

Probabilistic CCA is a reinterpretation of CCA as a latent variable model that motivates classical CCA as a generative model. One advantage of this CCA variant is that it has a more principled definition of the variation to be expected in the data and so has more opportunity to produce synthetic but plausible observations once the model has been fit. Additionally, because PCCA allows for the introduction of prior knowledge into the model specification, an advantageous aspect of many Bayesian models, this approach has been shown to yield more convincing results in small biomedical datasets which would otherwise be challenging to handle ordinary CCA (eg. Fujiwara et al., 2009, Huopaniemi et al., 2009).

2. Mathematical Formulation

2.1 Formulation:

Let X_a and X_b be two datasets of n observations of two random variables x_a and x_b and dimensionality p and q, respectively, that are generated by a shared latent variable z.

Latent variable Interpretation:

Rather than using linear algebra to set up an objective and then solve for two linear projections w_a and w_b, we instead write down a model that captures these probabilistic relationships and use maximum likelihood estimates to update its parameters.

The model is:

$$\begin{aligned} \mathbf{z} &\sim \mathcal{N}(0, I_r), \quad r \leq \min(p, q) \\ \mathbf{x}_a \mid \mathbf{z} &\sim \mathcal{N}(W_a \mathbf{z} + \boldsymbol{\mu}_a, \Psi_a) \\ \mathbf{x}_b \mid \mathbf{z} &\sim \mathcal{N}(W_b \mathbf{z} + \boldsymbol{\mu}_b, \Psi_b) \end{aligned} \tag{1}$$

Where W_a and W_b are two arbitrary matrices, and Ψ_a and Ψ_b are both positive semi-definite.

(Bach & Jordan, 2005) proved that the resulting maximum likelihood estimates are equivalent, up to rotation and scaling, to CCA.

In the probabilistic model, Equations (1) is a function of random variables. If we marginalize out z for either x_a or x_b, we get the following generative model:

$$x = Wz + u \tag{2}$$

where u is canonical direction vector and $u \sim N(\mu, \Psi)$.

If we assume our data is mean-centered, meaning $\mu = 0$ and rename W to Λ, we can see that PCCA is just group factor analysis with two groups (Klami et al., 2015):

$$\begin{aligned} \mathbf{x}_a &= \Lambda_a \mathbf{z} + \mathbf{u}_a \\ \mathbf{x}_b &= \Lambda_b \mathbf{z} + \mathbf{u}_b \end{aligned} \tag{3}$$

Now, the maximum likelihood estimates computed for factor analysis for PCCA can be used.

To see this, let's use block matrices to represent our data and parameters:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix}, \qquad \Lambda = \begin{bmatrix} \Lambda_a \\ \Lambda_b \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} \mathbf{u}_a \\ \mathbf{u}_b \end{bmatrix}$$

Where $x \in R^{p+q}$, $\Lambda \in R^{(p+q)\times k}$, and $u \in R^{p+q}$ and,

$$\mathbf{u} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Psi_a & 0 \\ 0 & \Psi_b \end{bmatrix} \right)$$

Then the PCCA updates are identical to the updates for factor analysis:

$$\Lambda^\star = \left(\sum_{i=1}^{n} \mathbf{x} \mathbb{E}_{\mathbf{z}|\mathbf{x}} \left[\mathbf{z} \mid \mathbf{x}_i \right]^\top \right) \left(\sum_{i=1}^{n} \mathbb{E}_{\mathbf{z}|\mathbf{x}} \left[\mathbf{z}\mathbf{z}^\top \mid \mathbf{x}_i \right] \right)^{-1}$$

$$\Psi^\star = \frac{1}{n} \operatorname{diag} \left(\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top - \Lambda^\star \mathbb{E}_{\mathbf{z}|\mathbf{x}} \left[\mathbf{z} \mid \mathbf{x}_i \right] \mathbf{x}_i^\top \right)$$

Futhermore, this definition gives us a joint density for x that should look similar to the density p(x) in factor analysis:

$$p(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Lambda_a \Lambda_a^\top + \Psi_a & \Lambda_a \Lambda_b^\top \\ \Lambda_b \Lambda_a^\top & \Lambda_b \Lambda_b^\top + \Psi_b \end{bmatrix} \right)$$

3. Algorithm

3.1 PCCA Algorithm:

Algorithm for Probabilistic Canonical Correlation Analysis

Input: Two datasets X_a and X_b of dimensions p and q

Output: Latent variable Z matrix with reduced dimensions

Step1: Apply the latent variable interpretation as described above in the mathematical formulation.

Step2 : Use the EM algorithm shown below to calculate the cross-covarience matrix with reduced dimensions.

The EM algorithm provides a general framework for fitting the parameters of latent variable models (Dempster et al., 1977).

In particular, our latent variable formulation of CCA readily yields the following EM update equations:

$$\begin{aligned} W_{t+1} &= \tilde{\Sigma}\Psi_t^{-1}W_tM_t(M_t + M_tW_t^\top\Psi_t^{-1}\tilde{\Sigma}\Psi_t^{-1}W_tM_t)^{-1} \\ \Psi_{t+1} &= \begin{pmatrix} (\tilde{\Sigma} - \tilde{\Sigma}\Psi_t^{-1}W_tM_tW_{t+1}^\top)_{11} & 0 \\ 0 & (\tilde{\Sigma} - \tilde{\Sigma}\Psi_t^{-1}W_tM_tW_{t+1}^\top)_{22} \end{pmatrix} \end{aligned}$$

where $M_t = (I + W_t^\top\Psi_t^{-1}W_t)^{-1}$.

The EM algorithm always converges to a solution of the form described above, where the specific solution that is found among solutions of this form depends on the initialization.

Step3: The equation in Step2 would be used to derive the Z matrix which is our reduced dimension matrix.

4. Documentation of API

4.1 Package Organization:

Required library: Numpy

4.2 Methods:

expectation_maximize(params): To implement the EM algorithm described above.

fit(params): Fits the model to data and return psi matrix.

transform(params): Transforms the fitted model and returns Z (latent variable used) matrix

fit_transform(params): Reduces the dimensionality of the trained data.

get_projections(params, n_samples): Returns the projected vectors X1 and X2.

pcca(inmats, nvecs, iters = 500, nprojs = 100)

Parameters:

inmats: input matrices [X1,X2]

nvecs: number of reduced dimensions

iters: number of iterations for optimization

nprojs: number of projections to generate

Returns the transformed X_1 and X_2 datasets along with dimensionally reduced Z matrix.

5. Examples

5.1 Example 1:

N = 10 # number of instances

NVECS = 2 # number of reduced dimensions

X = np.random.rand(N, 4)

Y = np.random.rand(N, 5)

Z, X1, X2 = pcca([X, Y], nvecs = NVECS, nprojs = N)

```
print(Z)
```

```
[[-0.92685876 -1.24807945]
 [-0.08401839 -1.00570488]
 [ 1.62996732 -0.30235556]
 [-0.59345041  0.6316667 ]
 [ 0.47387167  0.08871249]
 [ 0.23720424 -0.97758628]
 [ 0.74136739  0.8026159 ]
 [-0.74737633  0.80332649]
 [-0.10402118  0.73764799]
 [-0.62668555  0.46975659]]
```

5.2 Example 2:

N = 10 # number of instances

NVECS = 3 # number of reduced dimensions

Generate random data

X = np.random.rand(N, 6)

Y = np.random.rand(N, 8)

Z, X1, X2 = pcca([X, Y], nvecs = NVECS, nprojs = N)

```
print(Z)
```

```
[[ 0.03758889 -1.02736095  0.52083724]
 [-1.02062077  0.62835255 -1.38952221]
 [ 0.36994658 -1.06419378 -0.05873589]
 [ 1.47276287  0.20190599 -0.94186114]
 [-0.22459376 -0.00332402 -0.20935278]
 [ 0.33850169 -0.1029961   0.36483614]
 [-0.59834667 -0.65666066  0.98148924]
 [-1.20945419 -0.11301478 -0.4685088 ]
 [ 0.9024395   0.43949108 -0.09977397]
 [-0.06822414  1.69780066  1.30059217]]
```

6. Learning Outcomes

1. Explored the concept of Canonical Covarience Analysis (CCA).
2. Learnt the limitations of traditional CCA and eplored various extensions of CCA.
3. Learnt to use PCCA as probabilistic interpretation of CCA and how it can be used as a generative model.

A. References

Research paper:

F. R. Bach and M. I. Jordan, “A probabilistic interpretation of canonical correlation analysis,” Department of Statistics, University of California, Berkeley, CA, Tech. Rep. 688, 2005.