Canonical correlation Analysis

Qiuping Xu

Department of Math, FSU

1 Introduction

Principal component analysis and independent component analysis involve only one data set. Canonical correlation analysis (CCA) is a standard statistical technique for finding linear projections of two data sets that are maximally correlated.

Proposed by Hotelling in 1936 [1], Canonical correlation analysis can be seen as the problem of finding basis vectors for two sets of variables such that the correlation between the projections of the variables onto these basis vectors is mutually maximized. Correlation analysis is dependent on the coordinate system in which the variables are described, so even if there is a very strong linear relationship between two sets of multidimensional variables, depending on the coordinate system used, this relationship might not be visible as a correlation. Canonical correlation analysis seeks a pair of linear transformations, one for each of the sets of variables, such that when the set of variables is transformed, the corresponding coordinates are maximally correlated.

Let $(X_1, X_2) \in R^{n_1} \times R^{n_2}$ denote random vectors with covariance $(\Sigma_{11}, \Sigma_{22})$ and corss-covariance Σ_{12}. CCA finds paris of linear projections of two vectors, $(w_1^{'} X_1, w_2^{'} X_2)$ that are maximally correlated:

$$\begin{aligned}(w_1^*, w_2^*) &= \underset{w_1, w_2}{\operatorname{argmax}} \operatorname{corr}(w_1^{'} X_1, w_2^{'} X_2), \\ &= \underset{w_1, w_2}{\operatorname{argmax}} \sqrt{\frac{w_1^{'} \Sigma_{12} w_2}{w_1^{'} \Sigma_{11} w_1 w_2^{'} \Sigma_{22} w_2}}\end{aligned}$$

Since the objective is invariant to scaling of w_1 and w_2, the projections are constrained to have unit variance:

$$(w_1^*, w_2^*) = \underset{w_1^{'} \Sigma_{11} w_1 = w_2^{'} \Sigma_{22} w_2 = 1}{\operatorname{argmax}} w_1^{'} \Sigma_{12} w_2 \tag{1}$$

When finding multiple paris of vectors (w_1^i, w_2^i), subsequent projections are also constrained to be uncorrelated with previous ones, that is $w_1^i \Sigma_{11} w_1^j = w_2^i \Sigma_{22} w_2^j = 0$ for $i < j$. Assmebling the top k projection vector w_1^i into the columns of a matrix $A_1 \in \mathbb{R}^{n_1 \times k}$, and similarly placing w_2^i into $A_2 \in \mathbb{R}^{n_2 \times k}$, we obtain the following formulation to identify the top $k \leq \min(n_1, n_2)$ projections:

$$\begin{aligned}&\text{maximize:} \quad tr(A_1^T \Sigma_{12} A_2) \\ &\text{subject to:} \quad A_1^T \Sigma_{11} A_1 = A_2^T \Sigma_{22} A_2 = I\end{aligned}$$

There are several ways to express the solution to this objective. We follow the one shows in [2]. Define T as $\Sigma_{11}^{-\frac{1}{2}}\Sigma_{12}\Sigma_{22}^{-\frac{1}{2}}$, and let U_k and V_k be the matrices of the first k left and right singular vectors of T. Then the optimal objective value is the sum of the top k singular value of T and the optimum is attained at $(A_1^*, A_2^*) = (\Sigma_{11}^{-\frac{1}{2}}U_k, \Sigma_{22}^{-\frac{1}{2}}V_k)$. Note that this solution assumes that the covariance matrices Σ_{11} and Σ_{22} are nonsingular, which is satisfied in practice because they are estimated from data with regularization: given cneted data matrices $\bar{H}_1 \in \mathbb{R}^{n_1 \times m}$, $\bar{H}_2 \in \mathbb{R}^{n_2 \times m}$, one can estimate, e.g.

$$\hat{\Sigma_{11}} = \frac{1}{m-1}\bar{H}_1\bar{H}_1{}' + r_1 I \tag{2}$$

wher $r_1 > 0$ is a regularization parameter. Estimating the covariance matrices with regularization also reduces the detection of spurious correlation in the training data.

2 Singular value decomposition

Since in both PCA and CCA, singular value decomposition is used at various point to solve for the optimal solution. Here I would like to spend some place to explain the concept of singular value decomposition.

Singular value decomposition (SVD) is based on a theorem from linear algebra which says that a rectangular matrix A can be broken down into the product of three matrices - an orthogonal matrix U, a diagonal matrix S, and the transpose of an orthogonal matrix V. The theorem is usually presented something like this:

$$A_{mn} = U_{mn}S_{mn}V_{nn}^T \tag{3}$$

The columns of U are orthonormal eigenvectors of AA^T, the columns of V are orthonormal eigenvectors of A^TA and S is a diagonal matrix containing the square roots of eigenvalues from U or V (they share the same eigenvalues) in descending order.

3 Matlab Function

3.1 Syntax

There is a simple matlab function for CCA. It is called **canoncorr**. The standard syntax is

```
[w1 w2 r U V] = canoncorr(X, Y);
```

Here the X, Y are the two data sets. They are ordered as observation $\times$ # of variables. The **columns of** $w1$ and $w2$ contain the canonical coefficients, i.e., the linear combination of variables making up canonical variable for X and Y, respectively. Columns of U and V are the canonical variables. r shows the correlation coefficient between each pair of the canonical variables.

$$U = (X - repmat(mean(X), N, 1)) * w1$$
$$V = (Y - repmat(mean(Y), N, 1)) * w2$$

N is the number of observations (samples).

One important thing to notice here is that **Columns of $w1$ and $w2$ are scaled to make the covariance matrices of the canonical variables the identity matrix**

3.2 Example

Now, let us look at an example of CCA

- Let us first construct some synthetic data

```
%construct a data set satisfy 4*a+3*b=2*c+d
a=rand(100,1);
b=rand(100,1);
c=rand(100,1);
d=4*a+3*b-2*c+0.1*[rand(100,1)-0.5];
% form two set of data [a, b] and [a,d]
X=[a,b];
Y=[c,d];
clear a b c d
[w1 w2 r U V] = canoncorr(X, Y);
```

- We exam the results. We manually construct 4∗column1 of X+3∗column2 of X= 2∗column1 of y+1∗column2 of y. The first item of r is 0.99 because of the random noise we added for d. And the first column of $w1$ and $w2$ define the canonical coefficient. We got the first column of A and B are $[2.7976, 2.0950]^{'}$ and $[1.4023, 0.6984]^{T}$. They are not exactly give us the same coefficient as we used in generating the data. But the ratio are the same as what we used (e.g. $2.7976/2.0950 \sim 4/3$, $1.4023/0.6984 \sim 2/1$)
  There are couple of reasons why this happens, first, the correlation coefficient is invariant if we scale both sides of equation, second, matlab automaticlly scale columns of $w1$ and $w2$ to make the covariance matrices of the canonical variables the identity matrix.
  If we run

```
corr(U)
corr(V)
```

We will have an identity matrix. The visualization of the canonical variable can be visualized by scatter plots.

4 Alternative Implementation

There is another implementation available at `http://www.imt.liu.se/~magnus/cca/`. This is more similar to what we described in the introduction part. The canonical coefficients are scaled into unit vector.

```
function [Wx, Wy, r] = cca(X,Y)

% CCA calculate canonical correlations
%
% [Wx Wy r] = cca(X,Y) where Wx and Wy contains the canonical
% correlation vectors as columns and r is a vector with
% corresponding canonical correlations. The correlations are
% sorted in descending order. X and Y are matrices where
% each column is a sample. Hence, X and Y must have
% the same number of columns.
%
% Example: If X is M*K and Y is N*K there are L=MIN(M,N) solutions.
% Wx is then M*L, Wy is N*L and r is L*1.
%
%
%  2000 Magnus Borga, Linkpings universitet

% --- Calculate covariance matrices ---

z = [X;Y];
C = cov(z.');
sx = size(X,1);
sy = size(Y,1);
Cxx = C(1:sx, 1:sx) + 10^(-8)*eye(sx);
Cxy = C(1:sx, sx+1:sx+sy);
Cyx = Cxy';
Cyy = C(sx+1:sx+sy, sx+1:sx+sy) + 10^(-8)*eye(sy);
invCyy = inv(Cyy);

% --- Calcualte Wx and r ---

[Wx,r] = eig(inv(Cxx)*Cxy*invCyy*Cyx); % Basis in X
r = sqrt(real(r));      % Canonical correlations

% --- Sort correlations ---

V = fliplr(Wx); % reverse order of eigenvectors
r = flipud(diag(r)); % extract eigenvalues anr reverse their orrer
[r,I]= sort((real(r))); % sort reversed eigenvalues in ascending order
r = flipud(r); % restore sorted eigenvalues into descending order
```

```
for j = 1:length(I)
  Wx(:,j) = V(:,I(j));  % sort reversed eigenvectors in ascending order
end
Wx = fliplr(Wx); % restore sorted eigenvectors into descending order

% --- Calcualte Wy  ---

Wy = invCyy*Cyx*Wx;     % Basis in Y
Wy = Wy./repmat(sqrt(sum(abs(Wy).^2)),sy,1); % Normalize Wy
```

References

1. Hotelling, H., *Relations between two sets of variants.* . Biometrika, 28(321-377), 1936.
2. Mardia, K. V. and Kent, J. T. and Bibby, J. M., *Multivariate Analysis.*. Academic Press. 1979