MACHINE LEARNING

Kernel Canonical Correlation Analysis
Structure of today’s and next week’s class

1) Briefly go through one extension of principal component analysis, namely Canonical Correlation Analysis (CCA).

2) Derive the non-linear version of CCA, kernel CCA (kCCA).

3) Make an exercise to understand the modulation of the space generated by CCA and kCCA.
Canonical Correlation Analysis (CCA)

\[ x \in \mathbb{R}^{N_x} \quad \quad \quad y \in \mathbb{R}^{N_y} \]

Determine features in two (or more) separate descriptions of the dataset that best explain each datapoint.

Extract hidden structure that maximize correlation across two different projections.
Canonical Correlation Analysis (CCA)

Pair of multidimensional zero mean variables

We have $M$ instances of the pairs.

Search two projections $w_x$ and $w_y$:

$$z_x = w_x^T X \quad \text{and} \quad z_y = w_y^T Y$$

solutions of:

$$\max \rho = \max_{w_x, w_y} \ \text{corr} \left( z_x, z_y \right)$$

$$X = \left\{ x^i \in \mathbb{R}^{N_x} \right\}_{i=1}^M, Y \in \left\{ y^i \in \mathbb{R}^{N_y} \right\}_{i=1}^M$$

$$\max \ corr \left( w_x^T x, w_y^T y \right)$$
Canonical Correlation Analysis (CCA)

Search two projections $w_x$ and $w_y$:

$$z_x = w_x^T X \quad \text{and} \quad z_y = w_y^T Y$$

solutions of:

$$\max \rho = \max_{w_x,w_y} \text{corr}(z_x, z_y)$$

$$= \max_{w_x,w_y} \frac{w_x^T E\{XY^T\} w_y}{\|w_x^T X\| \|w_y^T Y\|} = \max_{w_x,w_y} \frac{w_x^T C_{xy} w_y}{\sqrt{w_x^T C_{xx} w_x w_y^T C_{yy} w_y}}$$

With $X$ and $Y$ zero mean, i.e.

$$E\{X\} = E\{Y\} = 0$$
Canonical Correlation Analysis (CCA)

Crosscovariance matrix

$C_{xy}$ is $N_x \times N_y$

Measure crosscorrelation between $X$ and $Y$.

Solutions of:

$$\max \rho = \max_{w_x,w_y} \text{corr}(z_x,z_y)$$

$$= \max_{w_x,w_y} \frac{w_x^T E\{XY^T\} w_y}{\|w_x^T X\| \|w_y^T Y\|}$$

$$= \max_{w_x,w_y} \frac{w_x^T C_{xy} w_y}{\sqrt{w_x^T C_{xx} w_x w_y^T C_{yy} w_y}}$$

With $X$ and $Y$ zero mean, i.e.

$E\{X\} = E\{Y\} = 0$

Covariance matrices

$C_{xx} = E\{XX^T\} : N_x \times N_x$

$C_{yy} = E\{YY^T\} : N_y \times N_y$
Canonical Correlation Analysis (CCA)

Correlation not affected by rescaling the norm of the vectors, 

⇒ we can ask that \( w_x^T C_{xx} w_x = w_y^T C_{yy} w_y = 1 \)

\[
\max \rho = \max_{w_x, w_y} \frac{w_x^T E \{ X Y^T \} w_y}{\| w_x^T X \| \| w_y^T Y \|} = \max_{w_x, w_y} \sqrt{\frac{w_x^T C_{xy} w_y}{w_x^T C_{xx} w_x w_y^T C_{yy} w_y}}
\]

solutions of: 

\[
\max \rho = \max_{w_x, w_y} \text{corr}(z_x, z_y)
\]
Canonical Correlation Analysis (CCA)

Correlation not affected by rescaling the norm of the vectors,
\[ \Rightarrow \text{we can ask that } w_x^T C_{xx} w_x = w_y^T C_{yy} w_y = 1 \]

\[ \max \rho = \max_{w_x, w_y} \ w_x^T C_{xy} w_y \]
\[ \text{u. c. } w_x^T C_{xx} w_x = w_y^T C_{yy} w_y = 1 \]

To determine the optimum (maximum) of \( \rho \), solve by Lagrange:
\[ L(w_x, w_y, \lambda_x, \lambda_y) = w_x^T C_{xy} w_y - \lambda_x (w_x^T C_{xx} w_x - 1) - \lambda_y (w_y^T C_{yy} w_y - 1) \]

Taking the partial derivatives over \( w_x, w_y \)
\[ \Rightarrow \lambda_x = \lambda_y := \lambda / 2 \]
Canonical Correlation Analysis (CCA)

Replacing $\lambda$ and write the set of equations gives:

\[
\begin{pmatrix}
0 & C_{xy} \\
C_{yx} & 0
\end{pmatrix}
\begin{pmatrix}
w_x \\
w_y
\end{pmatrix}
= \lambda
\begin{pmatrix}
C_{xx} & 0 \\
0 & C_{yy}
\end{pmatrix}
\begin{pmatrix}
w_x \\
w_y
\end{pmatrix}
\]

$\Rightarrow$ Which can be rewritten as

\[
C_{xy}C_{yy}^{-1}C_{yx}w_x = \lambda^2 C_{xx}w_x
\]

Solving for $w_y$ gives:

\[
C_{yx}C_{xx}^{-1}C_{xy}w_y = \lambda^2 C_{yy}w_y
\]

If $C_{yy}$ is invertible, it becomes an eigenvalue problem as for $w_y$.

These two eigenvalue problems yield a pair of $q$ vectors $\{w_x^i, w_y^i\}_{i=1..q}$, where $q = \min(N_x, N_y)$

\[
w_x^i \in \mathbb{R}^{N_x}, w_y^i \in \mathbb{R}^{N_y}
\]
CCA: Exercise I

Consider the example below of a dataset of 4 points with 2-dimensional coordinates in both X and Y.

- Determine by hand the directions found by CCA in each space.
- Contrast to the directions found by PCA.
Kernel Canonical Correlation Analysis

CCA finds basis vectors, s.t. the correlation between the projections (of all datapoints in $X$ and $Y$) is mutually maximized.

CCA is a generalized version of PCA for two or more multi-dimensional datasets, but unlike PCA it does have the constraint to find orthogonal vectors.

Assumes a linear correlation. If correlation non-linear $\rightarrow$ Kernel CCA.
Kernel Canonical Correlation Analysis (kCCA)

Assume two transformations

\[ \{x^1, y^1\} \]

\[ \{x^2, y^2\} \]

And then perform correlation analysis in feature space across the two feature spaces.

\[ \text{max } corr \left( w^T_x \phi_x(x), w^T_y \phi_y(y) \right) \]
**From CCA to Kernel CCA**

\[ X = \{ x^i \in \mathbb{R}^{N_x} \}_{i=1}^M, Y = \{ y^i \in \mathbb{R}^{N_y} \}_{i=1}^M \]

Send into two separate feature spaces for data in \( X \) and in \( Y \).

\[ \left\{ \phi_x \left( x^i \right) \right\}_{i=1}^M \quad \text{and} \quad \left\{ \phi_y \left( y^i \right) \right\}_{i=1}^M, \quad \text{with} \quad \sum_{i=1}^M \phi_x \left( x^i \right) = 0 \quad \text{and} \quad \sum_{i=1}^M \phi_y \left( y^i \right) = 0 \]

Construct associated kernel matrices:

\[ K_x = F_x^T F_x, \quad K_y = F_y^T F_y, \quad \text{columns of } F_x, F_y \text{ are } \phi_x \left( x^i \right), \quad \phi_y \left( y^i \right) \]
From CCA to Kernel CCA

In Linear CCA, we were solving for:

\[ \max_{w_x, w_y} w_x^T C_{xy} w_y \]

u.c. \( w_x^T C_{xx} w_x = w_y^T C_{yy} w_y = 1 \)

Express the projection vectors as a linear combination of images of datapoints in feature space (as in kPCA):

\[ w_x = F_x \alpha_x \text{ and } w_y = F_y \alpha_y \]

\[ \Rightarrow w_x = \sum_{i=1}^M \alpha_{x,i} \phi_x \left( x^i \right) \text{ and } w_x = \sum_{i=1}^M \alpha_{y,i} \phi_y \left( y^i \right) \]

In kernel CCA, we solve for:

\[ \max_{\alpha_x, \alpha_y} \alpha_x^T F_x^T F_x F_y F_y \alpha_y \]

u.c. \( \alpha_x^T F_x^T F_x F_y F_y \alpha_y = 1 \)

Replace the covariance and crosscovariance matrices by the product of the projection vectors in feature space (as in kPCA):

\[ C_{xx} = F_x F_x^T \]

\[ C_{yy} = F_y F_y^T \]

\[ C_{xy} = F_x F_y^T \]
Kernel CCA

In summary, in kernel CCA, we search the projection vectors $w_x, w_y$ (that live in feature space) so as to maximize:

$$\max_{w_x, w_y} \rho = \max_{w_x, w_y} corr\left(w_x \phi_x(x), w_y \phi_y(y)\right)$$

This is again a generalized eigenvalue problem with $\alpha_x, \alpha_y$ the dual eigenvectors (as dual vectors in kPCA), see documentation in annexes for derivation.

Generalized eigenvalue problem:

$$\begin{pmatrix} 0 & K_x K_y \\ K_y K_x & 0 \end{pmatrix} \begin{pmatrix} \alpha_x \\ \alpha_y \end{pmatrix} = \lambda \begin{pmatrix} K_x^2 & 0 \\ 0 & K_y^2 \end{pmatrix} \begin{pmatrix} \alpha_x \\ \alpha_y \end{pmatrix}$$
Kernel CCA

If the intersection between the spaces spanned by $K_x \alpha_x$, $K_y \alpha_y$ is non-zero, then the problem has a trivial solution, as $\rho \sim \cos(K_x \alpha_x, K_y \alpha_y) = 1$ (see solution to the exercises).

$$\max_{w_x, w_y} \rho = \max_{\alpha_x, \alpha_y} \alpha_x^T K_x K_y \alpha_y$$

u.c. $(\alpha_x^T K_x^2 \alpha_x) = (\alpha_y^T K_y^2 \alpha_y) = 1$

Generalized eigenvalue problem:

$$\begin{pmatrix}
0 & K_x K_y \\
K_y K_x & 0
\end{pmatrix}
\begin{pmatrix}
\alpha_x \\
\alpha_y
\end{pmatrix}
= \lambda
\begin{pmatrix}
K_x^2 & 0 \\
0 & K_y^2
\end{pmatrix}
\begin{pmatrix}
\alpha_x \\
\alpha_y
\end{pmatrix}$$
Kernel CCA

Add a regularization term to increase the rank of the matrix and make it invertible (to avoid the trivial solution)

\[ K_x^2 \rightarrow \left( K_x + \frac{M \kappa}{2} I \right)^2, \quad \kappa > 0 \]

Generalized eigenvalue problem:

\[
\begin{pmatrix}
0 & K_x K_y \\
K_y K_x & 0
\end{pmatrix}
\begin{pmatrix}
\alpha_x \\
\alpha_y
\end{pmatrix}
= \lambda
\begin{pmatrix}
K_x^2 & 0 \\
0 & K_y^2
\end{pmatrix}
\begin{pmatrix}
\alpha_x \\
\alpha_y
\end{pmatrix}
\]

Several methods have been proposed to choose carefully the regularization term so as to get projections that are as close as possible to the “true” projections.
Kernel CCA

\[
\begin{pmatrix}
0 & K_x K_y \\
K_y K_x & 0
\end{pmatrix}
\begin{pmatrix}
\alpha_x \\
\alpha_y
\end{pmatrix}
= \lambda
\begin{pmatrix}
\left( K_x + \frac{M \kappa}{2} I \right)^2 & 0 \\
0 & \left( K_y + \frac{M \kappa}{2} I \right)^2
\end{pmatrix}
\begin{pmatrix}
\alpha_x \\
\alpha_y
\end{pmatrix}
\]

Set: \( B = C^T C \) and \( \beta = C \alpha \)

Becomes a classical eigenvalue problem \( \Rightarrow \) \( \left(C^T\right)^{-1} A C^{-1} \beta = \lambda \beta \)
**Kernel CCA**

\[
X = \left\{ x^i \in \mathbb{R}^{N_x} \right\}_{i=1}^M, \quad Y = \left\{ y^i \in \mathbb{R}^{N_y} \right\}_{i=1}^M
\]

Two datasets case

Can be extended to multiple datasets:

\[L \text{ datasets: } X_1, \ldots, X_L \text{ with } M \text{ observations each}\]

Dimensions \(N_1, \ldots, N_L\); i.e. \(X_i : N_i \times M\)

Applying non-linear transformation \(\phi_1\), to \(X_1, \ldots, X_L\)

→ construct \(L\) Gram matrices: \(K_1, \ldots, K_L\)
**Kernel CCA**

\[
X = \left\{ x^i \in \mathbb{R}^{N_x} \right\}_{i=1}^M, \quad Y = \left\{ y^i \in \mathbb{R}^{N_y} \right\}_{i=1}^M
\]

Two datasets case

Can be extended to multiple datasets:

\[L \text{ datasets: } X_1, \ldots, X_L \text{ with } M \text{ observations each}\]

Dimensions \(N_1, \ldots, N_L\); i.e. \(X_i : N_i \times M\)

\[
\begin{pmatrix}
0 & K_1 K_2 & \cdots & K_1 K_L \\
K_2 K_1 & 0 & \cdots & K_2 K_L \\
\vdots & & \ddots & \vdots \\
K_L K_1 & K_L K_2 & \cdots & 0
\end{pmatrix}
\begin{pmatrix}
\alpha_1 \\
\alpha_2 \\
\vdots \\
\alpha_L
\end{pmatrix}
= \lambda
\begin{pmatrix}
\left( K_1 + \frac{M \kappa}{2} I \right)^2 & 0 \\
0 & \left( K_L + \frac{M \kappa}{2} I \right)^2
\end{pmatrix}
\begin{pmatrix}
\alpha_1 \\
\alpha_2 \\
\vdots \\
\alpha_L
\end{pmatrix}
\]
Kernel CCA

Two datasets case

had as solution the following generalized eigenvalue problem:

\[
\begin{pmatrix}
0 & K_xK_y \\
K_yK_x & 0
\end{pmatrix}
\begin{pmatrix}
\alpha_x \\
\alpha_y
\end{pmatrix}
= \lambda
\begin{pmatrix}
\left( K_x + \frac{M \kappa}{2} I \right)^2 & 0 \\
0 & \left( K_y + \frac{M \kappa}{2} I \right)^2
\end{pmatrix}
\begin{pmatrix}
\alpha_x \\
\alpha_y
\end{pmatrix}
\]

Can be extended to multiple datasets (MKCCA)

\[
\begin{pmatrix}
0 & K_1K_2 & \ldots & K_1K_L \\
K_2K_1 & 0 & \ldots & K_2K_L \\
\vdots & \vdots & \ddots & \vdots \\
K_LK_1 & K_LK_2 & \ldots & 0
\end{pmatrix}
\begin{pmatrix}
\alpha_1 \\
\alpha_2 \\
\vdots \\
\alpha_L
\end{pmatrix}
= \lambda
\begin{pmatrix}
\left( K_1 + \frac{M \kappa}{2} I \right)^2 & 0 \\
0 & \left( K_L + \frac{M \kappa}{2} I \right)^2
\end{pmatrix}
\begin{pmatrix}
\alpha_1 \\
\alpha_2 \\
\vdots \\
\alpha_L
\end{pmatrix}
\]

\[
X = \left\{ x^i \in \mathbb{R}^{N_x} \right\}_{i=1}^M, Y = \left\{ y^i \in \mathbb{R}^{N_y} \right\}_{i=1}^M
\]
Consider the following kernel matrices:

\[ K_x = \begin{bmatrix} 1 & 0.5 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad K_y = \begin{bmatrix} 1 & 0.8 & 0 \\ 0.8 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

a) How many datapoints do you have? What are the dimensions of the two feature spaces?

b) Assume a RBF kernel with same kernel width for \( K_x \) and \( K_y \), draw the distribution of points and give the shape of the dual vectors \( \alpha_x \) and \( \alpha_y \), solutions of \( \max_{\alpha_x,\alpha_y} \rho = \max_{\alpha_x,\alpha_y} \alpha_x^T K_x K_y \alpha_y \) with \( \alpha_x^T K_x^2 \alpha_x = \alpha_y^T K_y^2 \alpha_y = 1 \).

c) What is the effect of changing the kernel width on \( K_x \) and \( K_y \) and on \( \alpha_x \) and \( \alpha_y \)?

d) Do (b) when considering a polynomial kernel. Assume then same distribution of points as for RBF kernel. What are the Gram matrices?
CCA: Exercise III

Consider the example below of a dataset of 4 points with 2-dimensional coordinates in both X and Y.

- What is the shape of the kernel matrices and dual eigenvectors and draw the isolines when considering a RBF kernel.
- Do the same with a polynomial kernel.
Applications of Kernel CCA

Goal: To measure correlation between heterogeneous datasets and to extract sets of genes which share similarities with respect to multiple biological attributes.

Kernel matrices $K_1$, $K_2$, and $K_3$ correspond to gene-gene similarities in pathways, genome position, and microarray expression data resp. Use RBF kernel with fixed kernel width.

Correlation scores in MKCCA: pathway vs. genome vs. expression.

Applications of Kernel CCA

Goal: To measure correlation between heterogeneous datasets and to extract sets of genes which share similarities with respect to multiple biological attributes.

Correlation scores in MKCCA: pathway vs. genome vs. expression.

Two clusters correspond to genes close to each other with respect to their positions in the pathways, in the genome, and to their expression.

A readout of the entries with equal projection onto the first canonical vectors $\alpha$ give the genes which belong to each cluster.

Gives pairwise correlation between K1, K2

Gives pairwise correlation between K1, K3

Applications of Kernel CCA

Goal: To construct appearance models for estimating an object’s pose from raw brightness images.

X: Set of images

Y: Pose parameters (pan and tilt angle of the object w.r.t. the camera in degrees)

Example of two image datapoints with different poses

Method: used linear kernel on X and RBF kernel on Y and compared performance to applying PCA on the (X, Y) dataset directly.

Applications of Kernel CCA

Goal: To construct appearance models for estimating an object’s pose from raw brightness images

Kernel-CCA performs better than kPCA, especially for small testing/training ratio (i.e., for larger training sets).

The kernel-CCA estimators tend to produce less outliers, i.e., gross errors, and consequently yield a smaller standard deviation of the pose estimation error than their PCA-based counterparts.

For very small training sets, the performance of both approaches becomes similar.

Summary

• CCA is an excellent means to discover appropriate projections when your data is multi-modal.

• In each modality (separately), CCA finds projections that highlight features common to the datapoints as a whole.

• It generates projections that are different from performing PCA on each modality separately.

• The non-linear version of CCA, kernel CCA, generates sets of projections different from linear CCA and from kPCA.

• These projections highlight sets of modalities that are common to groups of datapoints. It is a good pre-processing method before clustering.