EXERCISE SESSION kCCA: ADVANCED MACHINE LEARNING COURSE – EPFL – Lecturer A. Billard

Illustrations in the solution of the exercise can be generated from the matlab code provided in annexes (see class's website).

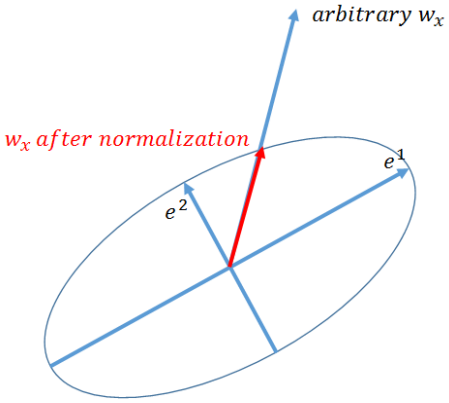
Exercise 1: Canonical Correlation Analysis

Recall that CCA looks for two vectors w_x, w_y for the datasets X and Y, respectively, which maximize the correlation between their respective projections, i.e.

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

a) Give a geometrical interpretation to the constraints $\mathrm{w_x^T} C_{xx} w_x = \mathrm{w_y^T} C_{yy} w_y = 1.$

Solution: The constraints bound the norm of w_x and w_y. The two vectors are on an ellipse whose axes are aligned with the eigenvectors of the covariance matrices C_{xx} and C_{yy}. The lengths of the ellipse's axes are given by the inverse of the square root of the eigenvalue of the associated eigenvectors. Applying this constraint amounts to scaling the norm of the vectors w_x and w_y so that they touch the ellipsoid isoline of value 1. It does not change the direction of the vectors w_x and w_y and hence does not change the relative correlation across pairs of vectors.



Proof:

Since C_{xx} and C_{yy} are square and symmetric, one can perform an eigenvalue decomposition: $w_x^T C_{xx} w_x = w_x^T U \Lambda U^T w_x = a^T \Lambda a = 1$

with $U = [e^1 \; e^2 ..]$ matrix of eigenvectors, $a = U^T w_x$ and ($a_1 = (e^1)^T w_x$).

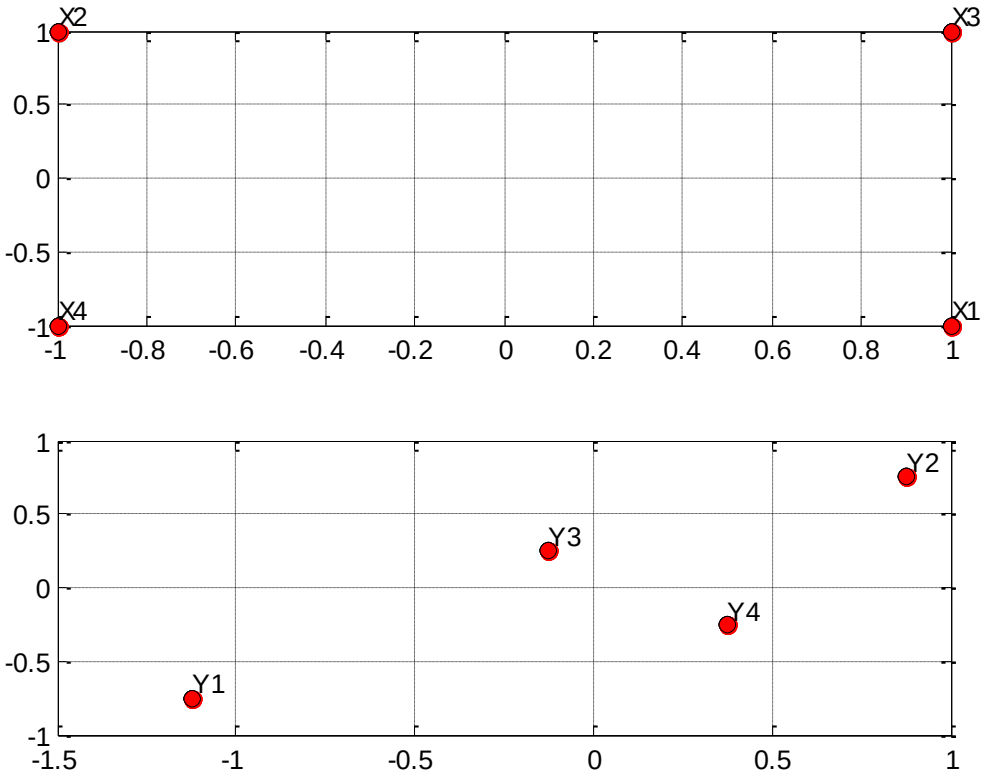
→This gives the equation of an ellipse: $a_1^2 \Lambda_1 + a_2^2 \Lambda_2 = 1,$, whose axes are aligned with the eigenvectors.

Besides, we also have a result on the variance of the projected data $\left\|\mathrm{w}_{\mathrm{x}}^{\mathrm{T}} X\right\|$:

$$\left\|\mathrm{w}_{\mathrm{x}}^{\mathrm{T}} X\right\|=(\mathrm{w}_{\mathrm{x}}^{\mathrm{T}} X)\cdot(X^{T} w_{x})=\mathrm{w}_{\mathrm{x}}^{\mathrm{T}} X X^{T} w_{x}=\mathrm{w}_{\mathrm{x}}^{\mathrm{T}} C_{xx} w_{x}=1.$$

In other words, projecting onto a vector that is solution of $\mathrm{w}_{\mathrm{x}}^{\mathrm{T}} C_{xx} w_{x}=1$ is called *whitening* the data, i.e. in the projection the data have unit variance. This step is interesting in that it normalizes data whose distribution may be larger in some dimensions.

b) Consider the example below of a dataset of 4 points with 2-dimensional coordinates in both X and Y.
   i) Determine by hand the directions found by CCA in each space.
   ii) Contrast to the directions found by PCA.



Solutions:

i) Since we can ignore the scaling factor given by the constraints, we focus on determining the directions of the vectors w_{x}, w_{y}. Each pair of vectors maximizes the correlation, i.e. $\max_{w^{x},w^{y}} corr\left(w_{x}^{T} X, w_{y}^{T} Y\right)$. In the example, we have a dataset composed of 4 datapoints that are two dimensional. They

compose the columns of the matrix $X = \left[x^1\ x^2\ x^3\ x^4 \right]$, idem for Y. If we define z_x, z_y as the vectors of coordinates of X and Y when projected onto w_x, w_y respectively. Since the projection in our 2-dimensional space is 1-dimensional, z_x, z_y are vectors of dimension 4. $\max_{w^x, w^y} corr\left(z_x, z_y\right)$ is maximal when the two vectors **are collinear**. The co-linearity is in dimension 4, i.e. the dimension of the datapoints! Hence, points in X must project *the same way* onto w_x as they do onto w_y. In other words, CCA will try to find projections in each space such that the datapoints have equal relative distance (i.e. relative coordinates) in each projection.

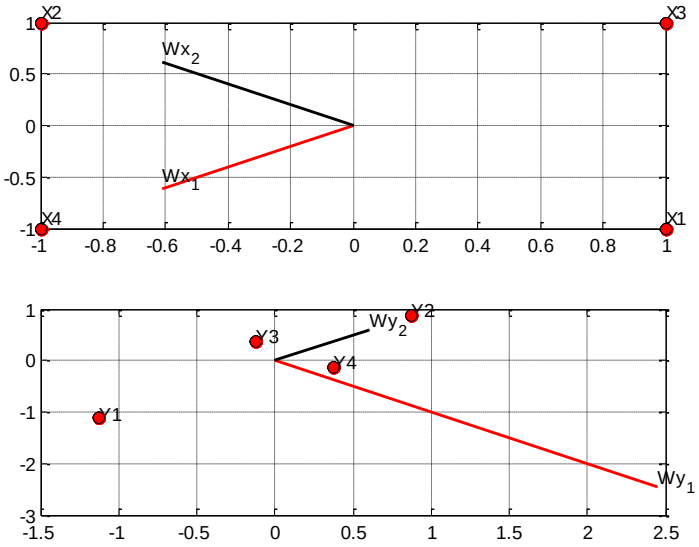
$z_x = \begin{bmatrix} z_x^1 \\ z_x^2 \\ z_x^3 \\ z_x^4 \end{bmatrix}$ where $z_x^i = w_x^T x^i$ is the coordinate of point x^i projected onto w_x.

$z_y = \begin{bmatrix} z_y^1 \\ z_y^2 \\ z_y^3 \\ z_y^4 \end{bmatrix}$ where $z_y^i = w_y^T y^i$ is the coordinate of point y^i projected onto w_y.

z_x, z_y are colinear implies that similar ratio acros the coordinates in each space,

e.g. $\frac{z_x^1}{z_x^2} = \delta \frac{z_y^1}{z_y^2}$, $\delta \in \mathbb{R}$.

Hence, whenever possible, CCA will hence find one pair of vectors that passes exactly through a set of datapoints. We plot below the solution for the example above:



The first projection w_x^1 splits the space along one of the two symmetry directions across the 4 datapoints (idem in y). In this case the two vectors are exactly identical, i.e. $corr(z_x, z_x) = 1$. This is due to the normalization of the norms in projected space. The correlation is then one when the vectors are exactly co-linear. The second projection maximizes the correlation along different directions. Observe that w_x^2 is not orthogonal to w_x^1 and passes through the second axes of symmetry.

iii) The major differences between CCA and PCA are that:

- PCA finds direction that are orthogonal and with norm 1, CCA instead finds directions that are not necessarily orthogonal and whose norm is scaled to fit the ellipsoid (see answer to question a).
- PCA finds directions in each space (X) and (Y) separately and the directions are hence not informed by the labels of the datapoints. CCA instead will make sure to find pairs of vectors in each space that yields some features common to groups of points.
- In the example above, PCA would find as first direction in *x* a line going through either x^1, x^2 or x^3, x^4 and the other directions will be orthogonal to the first one, and in *y* it will find a line that passes almost through y^1, y^2 while the second line will be orthogonal to this.

Exercise 2: Kernel Canonical Correlation Analysis

Recall that kernel CCA looks for the projection vectors α_x and α_y, solutions of:

$$\max_{\alpha_x,\alpha_y} \rho = \max_{\alpha_x,\alpha_y} \alpha_x K_x K_y \alpha_y$$

$$\text{u.c } \alpha_x K_x^2 \alpha_x = \alpha_y K_y^2 \alpha_y = 1$$

Consider that you are performing kernel CCA and that you obtain 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 we have and what is the original dimension of *X* and *Y?*

b) Assuming that the same RBF kernel (same kernel width) has been used for constructing Kx and Ky, draw the distribution of points that may have produced these matrices and give the values for the entries of the first set of dual projection vectors α_x and α_y. Assume now that a different RBF kernel has been used for constructing Kx and Ky, what is the effect on the resulting correlations across the two original spaces, X and Y.

c) Answer to a and b by assuming now a homogeneous polynomial kernel and consider that $x \in \mathbb{R}^2$ and $y \in \mathbb{R}^3$,

Solutions:

a) We have 3 points. CCA and kernel CCA require the same number of points to build X and Y. Points are paired. The dimension of the points in X and Y may be arbitrary.

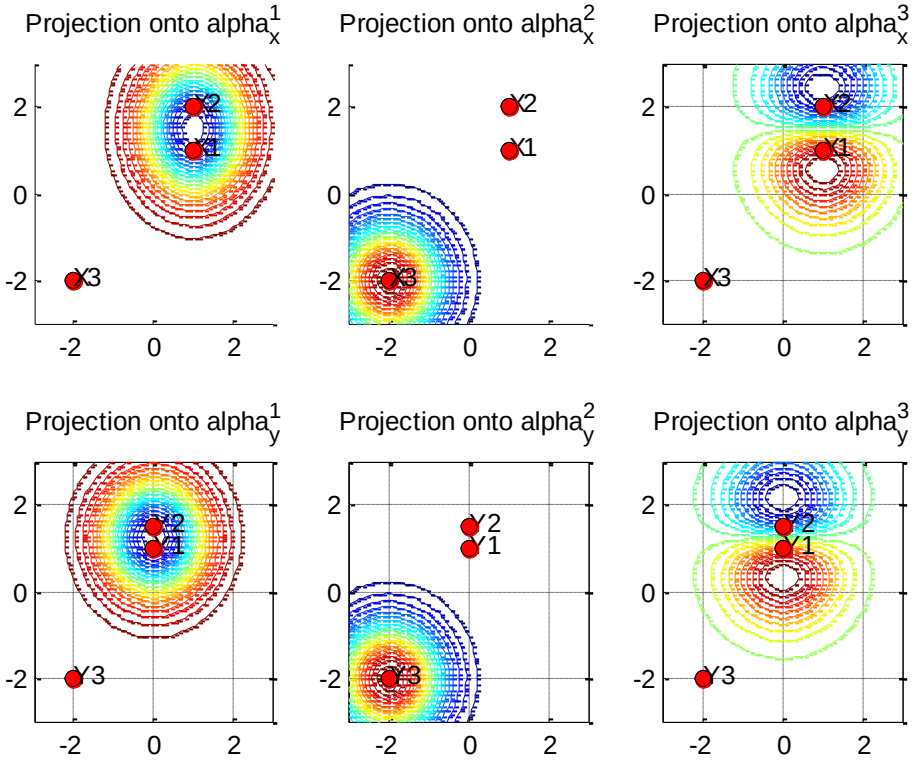
b) In the original space, the coordinates of the two first points in *X* are close to one another according to the kernel width (i.e. within 1 sigma). The third point is very far from the two first points. The same situation happens with coordinates in *Y*, with the two first points closer to each other and the third one even further away.

Following the same reasoning as for the first exercise, we can see that $\alpha_x K_x K_y \alpha_y$ is maximal when the vectors $z_x = \alpha_x K_x$ and $z_y = K_y \alpha_y$ are collinear. In the example above, it turns out that the Gram matrices K_x and K_y have three pairs

of collinear eigenvectors: $\alpha_x^1 = \alpha_y^1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$, $\alpha_y^2 = \alpha_y^2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ and $\alpha_y^3 = \alpha_y^3 = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$ with same eigenvalues (value 1).

Projecting onto the first set of dual vectors removes the effect of the third datapoint in both x and y. The resulting projection in x and y is the same. The isolines are ellipses around the first two datapoints. Each ellipse in X is correlated to a corresponding ellipse in Y.
Similarly projecting onto the second set of dual vectors removes the effect of the first two datapoints in both x and y. The isolines are ellipses around the last datapoint. Each ellipse in X is correlated to a corresponding ellipse in Y.

Given the distribution of points described previously, changing the kernel width does not affect the correlation as this is just a scaling of the values in the entries on the off-diagonal of each Gram matrix, unless the kernel is so large that the entries on the 2rd lines of the Gram matrices are no longer zero or so small that the Gram matrices become diagonal. Below is an example of such distribution of points in 2D and their associated isolines on the first two pair of eigenvectors.



c) If we assume that the entries on the Gram matrices remain the same (i.e. solution of inner product!), this would mean that distribution of the points is different. In *X* and *Y*, the two first point live in a plane orthogonal to the third point. All three point

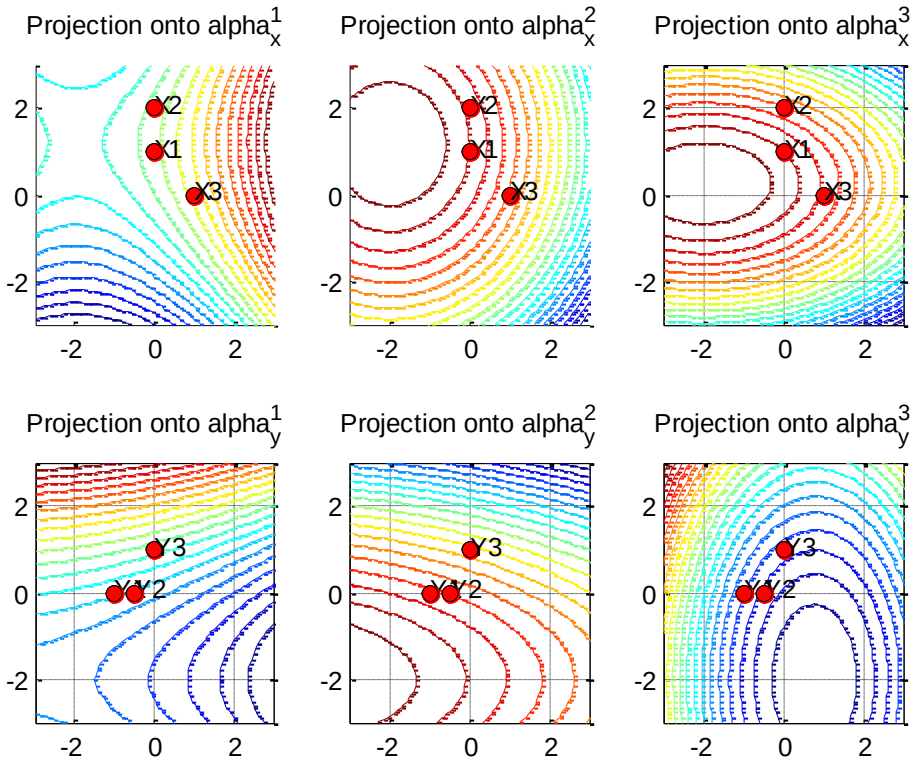
vectors have norm 1. The angle between the first two vectors is wider in x than in y. We get the same projections vectors α_x and α_y as for the RBF case. However, the isolines are different, depending on whether the kernel has odd or even order. If we consider that we do not have the same Gram matrices, but the same point distribution as in the (b) example above, then, we obtain a different decomposition of the dual eigenvectors. The Gram matrices become:

$$K_x = \begin{bmatrix} 9 & 16 & 0 \\ 16 & 36 & 4 \\ 0 & 4 & 100 \end{bmatrix}, \; K_y = \begin{bmatrix} 9 & 12 & 0 \\ 12 & 18 & 1 \\ 0 & 1 & 100 \end{bmatrix}$$

The dual eigenvectors are:

$$\alpha_x = \begin{bmatrix} 0.3 & 0.6 & -0.2 \\ -0.1 & -0.1 & 0.1 \\ 0.6 & 0.0 & -0.0 \end{bmatrix}, \; \alpha_y = \begin{bmatrix} -0.3 & -0.3 & -0.7 \\ 0.1 & 0.6 & 0.5 \\ 0.6 & -0.0 & -0.0 \end{bmatrix}$$

With an inhomogeneous polynomial of power p=2 and offset c=2, we obtain on the first dual eigenvectors a hyperbola and an ellipse for the other two. Notice that in the first two projections, the group of points x^1, x^2 and y^1, y^2 lie on isoline of same value, whereas the third projection groups the pair of points x^1, x^3 and y^1, y^3.



Exercise 3: Kernel Canonical Correlation Analysis

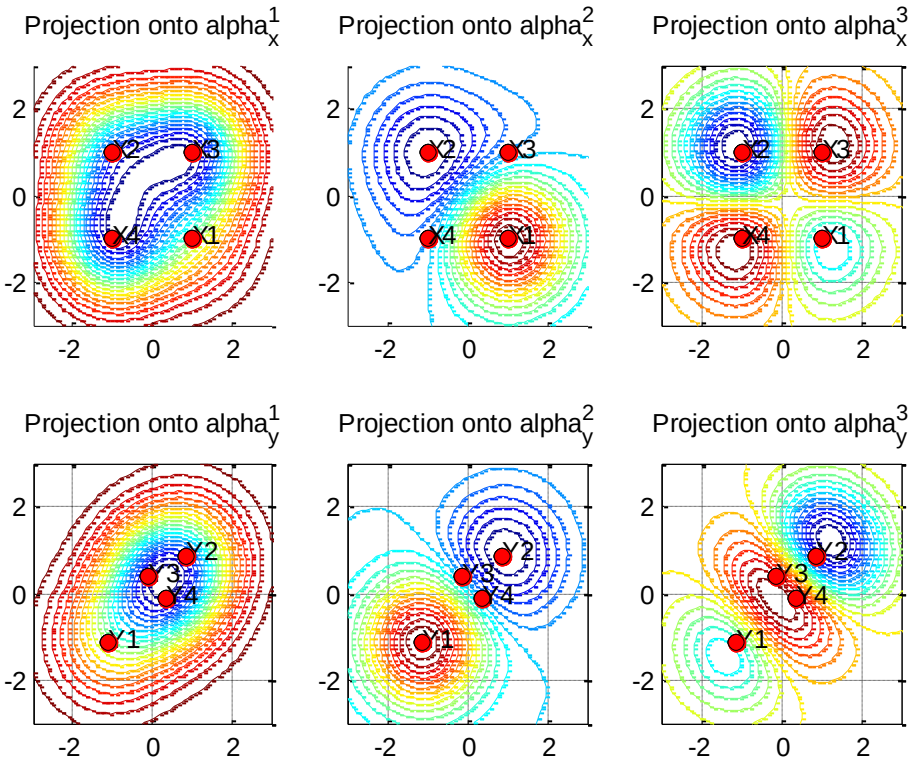
Consider again the example shown in exercise 1 (i.e. dataset of 4 points with 2-dimensional coordinates in both X and Y)

- What is the shape of the kernel matrices and dual eigenvectors?
- Draw the isolines when considering a RBF kernel.
- Do the same with a polynomial kernel.

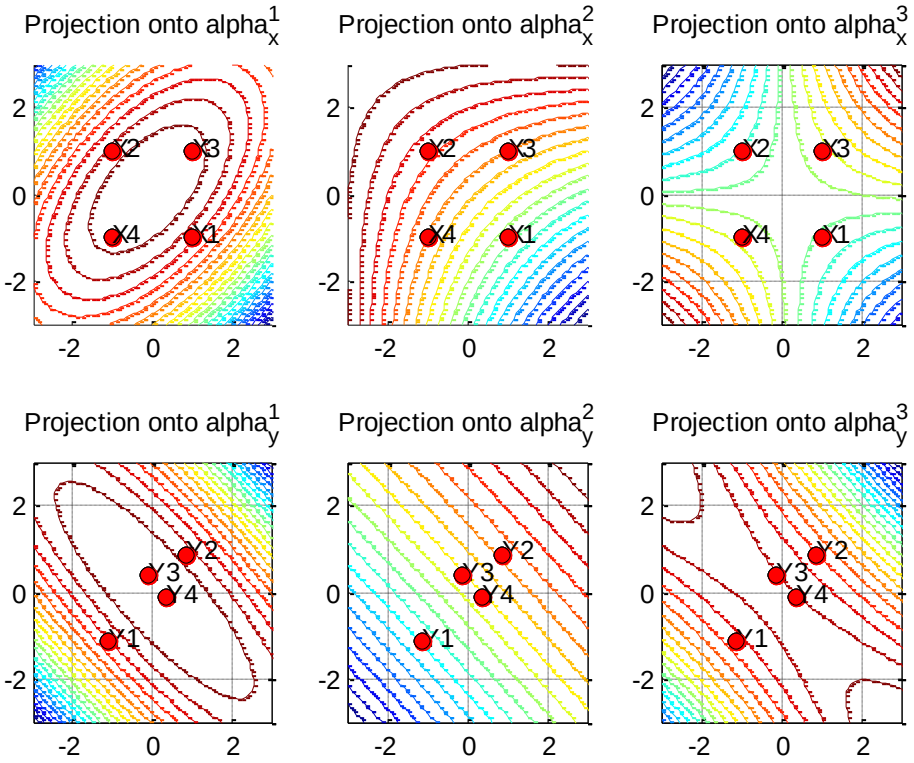
Solution:

For an average kernel width, the Gram matrix K_x has small values for all off-diagonal elements, whereas K_y has high values for all crossings between row/column 3 and 4.

The dual eigenvectors extract correctly the various groupings. For instance, the first projections (see below) groups the first closest three points, i.e. x^1, x^2, x^3 and y^1, y^2, y^3. The second projection remove the two middle points x^3, x^4 and y^3, y^4 and distinguish the two remaining points by giving them opposite values (very positive and very negative) on the isolines. The third projection highlights yet another axis of symmetry comparing pairwise the first and last two points. Notice again that the same group of datapoints lie on the same isolines in both X and Y projections.



With the polynomial kernel (again inhomogeneous with p=3 and c=2), we obtain the following projections:



The Gram matrices are different, and hence the dual eigenvectors are also slightly different in their absolute coordinates, however they retain a similar shape, as they embed the same notion of symmetry in the data (see matlab code for the values of the Gram matrices and dual eigenvectors). Hence, similarly to what we obtained with the RBF kernel, we get a grouping of the three first closest points on the first projection, then a clear distinction between the first and second point with opposite values on the isolines on the second projection and finally a separate grouping for pairs of points on the 3rd projection.

Conclusion:

This exercise illustrated how kernel CCA can be used to group datapoints according to the value they obtain in their respective projections. It is in this sense similar to kernel PCA. While kPCA performs this grouping using all dimensions of the datapoint at once, kCCA generates projections for each modality that groups set of datapoints the same way in each modality.

Supplement: Canonical Correlation Analysis Derivation

Show that CCA for two multivariate random variables (X, Y) can be rephrased as a generalized eigenproblem of the form $Ax = \lambda Bx$.

Hint: Formulate CCA as an optimization under constraint problems. To recall, CCA aims at determining a set of projection vectors w_x and w_y for X and Y such that the correlation ρ between the projections $X' = w_x^T \mathrm{X}$ and $\mathrm{Y}' = w_y^T Y$ is maximized.

Solution:

Maximizing the correlation across the two projection vectors is given by:

$$\max_{w_x, w_y} \operatorname{corr}(\mathrm{X}', \mathrm{Y}') = \max_{w_x, w_y} \frac{\mathrm{w}_\mathrm{x}^\mathrm{T} E\left\{xy^T\right\} \mathrm{w}_\mathrm{y}}{\left\|\mathrm{w}_\mathrm{x}^\mathrm{T} X\right\| \left\|\mathrm{w}_\mathrm{y}^T Y\right\|} = \max_{w_x, w_y} \frac{\mathrm{w}_\mathrm{x}^\mathrm{T} C_{xy} \mathrm{w}_\mathrm{y}}{\sqrt{\mathrm{w}_\mathrm{x}^\mathrm{T} C_{xx} w_x \mathrm{w}_\mathrm{y}^\mathrm{T} C_{yy} w_y}} \qquad (0.1)$$

Observe that the solution to the above problem is not affected by rescaling w_x or w_y either together or independently (e.g. replace in the above w_x by αw_x with α a scalar). Since the choice of rescaling is therefore arbitrary, the CCA optimization problem is equivalent to maximizing the numerator of Eq. 1.1 above subject to the constraint that:
$\mathrm{w}_\mathrm{x}^\mathrm{T} C_{xx} w_x = \mathrm{w}_\mathrm{y}^\mathrm{T} C_{yy} w_y = 1$

The corresponding Lagrangian is:

$$L(\lambda_x, \lambda_y, \mathrm{w}_x, \mathrm{w}_\mathrm{y}) = \mathrm{w}_\mathrm{x}^\mathrm{T} C_{xy} \mathrm{w}_\mathrm{y} - \lambda_x \left(\mathrm{w}_\mathrm{x}^\mathrm{T} C_{xx} \mathrm{w}_x - 1\right) - \lambda_y \left(\mathrm{w}_\mathrm{y}^\mathrm{T} C_{yy} \mathrm{w}_y - 1\right)$$

Taking the partial derivatives

$$\frac{\partial L}{\partial w_x} = C_{xy} \mathrm{w}_\mathrm{y} - 2\lambda_x C_{xx} \mathrm{w}_x \qquad (0.2)$$

$$\frac{\partial L}{\partial w_y} = C_{yx} \mathrm{w}_x - 2\lambda_y C_{yy} \mathrm{w}_y \qquad (0.3)$$

Multiplying 1.2 and 1.3 with w_x^T and w_y^T respectively**,** and then subtracting 0.2 from 0.3, we have:

$$0 = \mathrm{w}_x^T C_{xy} \mathrm{w}_\mathrm{y} - 2\lambda_x \mathrm{w}_x^T C_{xx} \mathrm{w}_x - \mathrm{w}_\mathrm{y}^T C_{yx} \mathrm{w}_x - 2\lambda_y \mathrm{w}_\mathrm{y}^T C_{yy} \mathrm{w}_y$$

$$0 = -2\lambda_x \mathrm{w}_x^T C_{xx} \mathrm{w}_x - 2\lambda_y \mathrm{w}_\mathrm{y}^T C_{yy} \mathrm{w}_y$$

which together with the constraints implies that $\lambda_x - \lambda_y = 0$. Let $\lambda = \lambda_x = \lambda_y$ and assuming that C_{yy} is invertible (can you tell when this may not be true?), we have:

$$\mathrm{w}_y = \frac{1}{2\lambda} C_{yy}^{-1} C_{yx} w_x$$

And so substituting in Eq. 0.2 gives:

$$C_{xy} C_{yy}^{-1} C_{yx} w_x = 4\lambda^2 C_{xx} w_x$$

Since λ is an arbitrary scalar, we can rescale it by dividing it by two. If C_{xx} is invertible, we obtain the following eigenvalue problem: $C_{xx}^{-1} C_{xy} C_{yy}^{-1} C_{yx} w_x = \lambda^2 w_x$.