Kernel assignment: advanced topics in machine learning

Arthur Gretton, Bariscan Bozkurt, Zhe Wang

September 27, 2025

COMP0083 student instructions: Sections 1 and 2 are mandatory - please follow the Moodle instructions for assignment submission. Section 3 is optional, and will not count toward the course mark, but if you choose to participate we will provide feedback. If you wish to participate in the optional Section 3, please email Zhe Wang by November 21 2025. Section 3 will not make use of Moodle.

Gatsby PhD student instructions: All sections are mandatory. Please submit all sections by email to Zhe Wang. The code must be emailed to Zhe in a text file; the proofs and plots must be submitted electronically (if written by hand, they may be scanned in).

Sectrions 1 and 2 are due on **Friday November 21 2025**.

Section 3 of the assignment involves teaming up with either one or two additional students, generating a dataset, and running your software on the dataset generated by another team (Section 3). Your datasets for kernel CCA must be emailed to Zhe by **11:59pm on Friday November 28 2025** (i.e. Zhe needs to receive by email Python or Matlab code to generate the data, and the plots of the canonical projection functions). Please submit this dataset on time, since it will be used by other students in the second part of the assignment. Your assessment of the dataset from another team is due **by 11:59pm Friday 12th December 2025**.

The UCL CS policy will apply to late submissions.

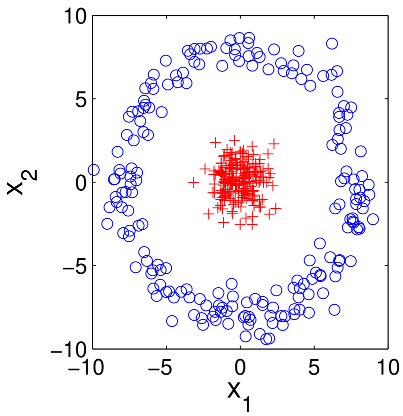
Please contact Bariscan Bozkurt with any questions on the assignment.

1 Feature spaces (30%)

1. Describe a *simple* (finite dimensional) feature space that allows error-free linear classification for the datasets in Figure 1 (the feature space coordinates will be functions of the input space coordinates x_1 and x_2).

2. Consider the case in which the input space $\mathcal{X}$ contains a finite number m of elements. You are given the inner product matrix K between the feature space mapping of every pair of elements x_i, x_j in $\mathcal{X}$, where the i,jth entry in K is written $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}} = (K)_{ij}$. Derive the feature

Figure 1: Ring dataset



space representation of each element $x_i \in \mathcal{X}$, $i \in \{1 \ldots m\}$. Hint: K is positive semidefinite and symmetric - what is its eigendecomposition?

2 Kernel dependence detection

2.1 Incomplete Cholesky for efficient COCO (20%)

We observe pairs (x_i, y_i) which we arrange in the matrices

$$X = \left[\begin{array}{ccc} \phi(x_1) & \ldots & \phi(x_n) \end{array} \right] \qquad Y = \left[\begin{array}{ccc} \psi(y_1) & \ldots & \psi(y_n) \end{array} \right],$$

where $x_i \in \mathcal{X}$, $\phi(x) \in \mathcal{F}$, $\mathcal{F}$ is an RKHS with kernel $k(x, x')$; and $y_i \in \mathcal{Y}$, $\psi(y) \in \mathcal{G}$, $\mathcal{G}$ is an RKHS with kernel $l(x, x')$. Define the Gram matrices K and L such that $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{F}} = (K)_{ij}$ and $l(y_i, y_j) = \langle \psi(y_i), \psi(y_j) \rangle_{\mathcal{G}} = (L)_{ij}$. The empirical covariance in feature space is

$$\begin{aligned} \widehat{C}_{XY} &= \frac{1}{n} \sum_{i=1}^{n} (\phi(x_i) - \hat{\mu}_x) \otimes (\psi(y_i) - \hat{\mu}_y) \\ &= \frac{1}{n} X H Y^\top, \end{aligned} \tag{1}$$

where

$$\hat{\mu}_x = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i) \qquad \hat{\mu}_y = \frac{1}{n} \sum_{i=1}^{n} \psi(y_i).$$

Recall from the lecture notes that the solution to

$$\begin{aligned}
\mathrm{COCO} := \max_{f,g} \left\langle f, \widehat{C}_{XY} g \right\rangle_{\mathcal{G}} & \\
\text{subject to} \quad \|f\|_{\mathcal{F}} = 1 & \quad (2) \\
\|g\|_{\mathcal{G}} = 1, & \quad (3)
\end{aligned}$$

is written

$$\begin{bmatrix} 0 & \frac{1}{n}\widetilde{K}\widetilde{L} \\ \frac{1}{n}\widetilde{L}\widetilde{K} & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \gamma \begin{bmatrix} \widetilde{K} & 0 \\ 0 & \widetilde{L} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

Here

$$f = \sum_{i=1}^{n} \alpha_i \left[\phi(x_i) - \hat{\mu}_x\right] = XH\alpha \qquad g = \sum_{j=1}^{n} \beta_i \left[\psi(y_i) - \hat{\mu}_y\right] = YH\beta,$$

and

$$\widetilde{K} = HKH \qquad \widetilde{L} = HLH, \tag{4}$$

where $H = I - n^{-1}\mathbf{1}_n$, and $\mathbf{1}_n$ is an $n \times n$ matrix of ones.[1]

Using the accompanying extract on incomplete Cholesky, taken from [1, Section 5.2], derive (i.e., show your working) and implement a more computationally efficient estimate of COCO (the estimate will not be exact). Compare the computational cost of COCO computed exactly, and approximated via incomplete Cholesky (give the number of operations, *not* just runtimes). **Hint:** do not just use incomplete Cholesky to factorize the kernel matrices! Rather, use the understanding that incomplete Cholesky is in fact an incomplete Gram Schmidt procedure, where feature maps of the training points $\{\phi(x_i)\}_{i\in\{1,\ldots,n\}}$ are projected onto the subspace spanned by the pivots $\{\phi(x_j)\}_{j\in\mathcal{I}}$, where $\mathcal{I} \subset \{1,\ldots,n\}$. What does this mean for the witness functions f, g?

Implement the incomplete Cholesky-based COCO in Python or Matlab using Gaussian kernels, and test it on the following data (see Figure 2).

$$\begin{aligned}
x &= \sin(t) + n_1 \\
y &= \cos(t) + n_2 \\
n_1, n_2 &\sim \mathcal{N}(0, 0.01^2) \\
t &\sim \mathcal{U}([0, 2\pi])
\end{aligned}$$

where $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian random variable with mean μ and variance σ^2, and $\mathcal{U}([a,b])$ is a uniform random variable on the interval $[a,b]$. The random variables t, n_1, n_2 are to be mutually independent. Plot f and g when a Gaussian kernel is used. Plot the mapping of (x,y) via these projections, and compute the correlation of the mapped variables.

[1] You can use that $H = HH$, and that XH is a matrix from which each column has had its mean subtracted: these are simple results, so you do not need to show working for them.

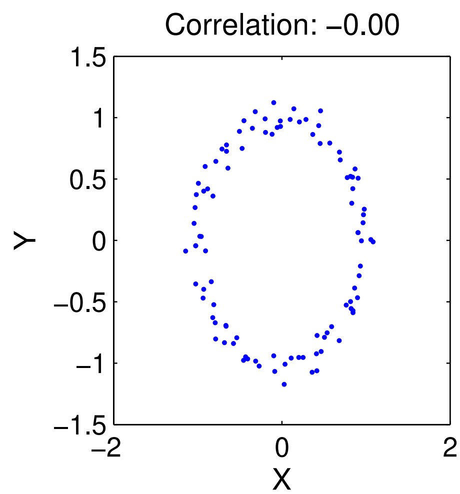


Figure 2: Data to be used for kernel CCA.

2.2 Kernel CCA (20%)

The canonical correlation is defined as

$$\arg\max_{f,g} \left(\operatorname{cov}[f(x), g(y)]\right) = \left\langle f, \widehat{C}_{XY} g \right\rangle_{\mathcal{G}}, \tag{5}$$

subject to the constraints

$$\begin{aligned} \operatorname{var}(f(x)) = \left\langle f, \widehat{C}_{XX} f \right\rangle_{\mathcal{F}} &= 1, & (6)\\ \operatorname{var}(g(y)) = \left\langle g, \widehat{C}_{YY} g \right\rangle_{\mathcal{G}} &= 1, & (7) \end{aligned}$$

where $\widehat{C}_{XY}$ is given in (1), and

$$\widehat{C}_{XX} = n^{-1} X H X^\top \qquad \widehat{C}_{YY} = n^{-1} Y H Y^\top.$$

Write a kernelized solution to the canonical correlation problem (5) in terms of the Gram matrices $\widetilde{K}$ and $\widetilde{L}$ defined in (4), as a generalized eigenvalue problem $U a_i = \lambda_i V a_i$ (U and V are matrices, a_i is the eigenvector, λ_i is the eigenvalue). Hints: **(1)** you may assume that

$$f = \sum_{i=1}^{n} \alpha_i \left[\phi(x_i) - \hat{\mu}_x\right] = X H \alpha \qquad g = \sum_{j=1}^{n} \beta_i \left[\psi(y_i) - \hat{\mu}_y\right] = Y H \beta.$$

(2) don't forget to keep track of the centring matrices H. Assume a Gaussian kernel, and that the points are also non-pathologically distributed so that K

and L have full rank. What went wrong? By adding suitable regularizing terms to (6) and (7), show you can obtain the solution

$$\begin{bmatrix} 0 & \frac{1}{n}\widetilde{K}\widetilde{L} \\ \frac{1}{n}\widetilde{L}\widetilde{K} & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \lambda \begin{bmatrix} \widetilde{K}^2 + \varkappa\widetilde{K} & 0 \\ 0 & \widetilde{L}^2 + \varkappa\widetilde{L} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \tag{8}$$

where $\widetilde{K}$ and $\widetilde{L}$ are defined in (4). Implement kernel CCA as above in Python or Matlab, using Gaussian kernels, and test it on the dataset in Figure 2. Compare functions f and g to those you got with COCO.

3 Dataset design for kernel CCA (15% for your dataset, 15% for results on other dataset)

Teaming up with either one or two other students, design a dataset for kernel CCA. Create variables (perhaps in more than one dimension) which have a nonlinear relationship, and plot the largest kernel canonical projections. You are encouraged to be creative in the choice of domain, even if this means that one of the canonical correlation functions f, g can't be plotted (though at least one projection function must be plottable, hence defined on $\Re$ or $\Re^2$). For instance, one of the domains might contain strings, or vectors of dimensionality greater than two. **Due 11:59pm on Friday 28 November 2025.**

Finally, we will assign your team a dataset generated by another team of students. Find and plot the largest canonical projection directions in this case. **Due 11:59pm Friday 12th December 2025.**

References

[1] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, Cambridge, UK, 2004.