67/70 (95.7%)

COMP0083

NMWX9, SN: 20114649

November 11, 2024

Contents

1 Feature spaces

1.1 Part 1

The dataset in the figure consists of two distinct classes: one is a cluster of points (red crosses) centered around the origin, and the other forms a ring (blue circles) surrounding the first class. This pattern is not linearly separable in the input space (x_1, x_2).

To transform this dataset into a space where it can be linearly separable, we can use a feature mapping that captures the radial symmetry of the ring. A simple feature space for this dataset can be defined using the following transformation:

$$\phi(x_1, x_2) = \begin{bmatrix} x_1^2 + x_2^2 \\ x_1 \\ x_2 \end{bmatrix}$$ ✓

Here's the reasoning behind this transformation:

- The first feature, $x_1^2 + x_2^2$, captures the squared Euclidean distance from the origin. This allows us to differentiate between points inside and outside the ring.
- The other two features, x_1 and x_2, retain the original coordinates to provide additional spatial information. ✓

In this new feature space, the points belonging to the inner cluster (red crosses) will have a smaller value for $x_1^2 + x_2^2$ compared to those on the outer ring (blue circles). Therefore, a linear classifier (e.g., a hyperplane) can separate the two classes error free based on the transformed features. ✓

good! 10/10

1.2 Part 2

Given a finite input space $X = \{x_1, x_2, \ldots, x_m\}$ and the corresponding kernel (inner product) matrix K where

$$K_{ij} = \Big\langle \phi(x_i), \phi(x_j) \Big\rangle_{\mathcal{H}},$$

we aim to derive the explicit feature space representations $\phi(x_i)$ for each $x_i \in X$.

Since K is symmetric and positive semidefinite (SPSD), it admits an eigendecomposition of the form ✓

$$K = U \Lambda U^\top,$$

where:

- $U = [u_1, u_2, \ldots, u_m]$ is an orthogonal matrix whose columns u_k are the eigenvectors of K,
- $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$ is a diagonal matrix containing the non-negative eigenvalues of K.

The feature map $\phi : X \to \mathcal{H}$ can be expressed in terms of the eigendecomposition as follows:

$$\phi(x_i) = \sum_{k=1}^{m} \sqrt{\lambda_k}\, u_k(i)\, e_k,$$

↳ good, but simplifies 19/20

where:

- $u_k(i)$ denotes the i-th component of the eigenvector u_k,
- $\{e_k\}_{k=1}^m$ is an orthonormal basis for the feature space $\mathcal{H}$.

Alternatively, in matrix form, the feature matrix Φ whose columns are the feature vectors $\phi(x_i)$ can be written as

$$\Phi = U \Lambda^{1/2},$$

where $\Lambda^{1/2} = \mathrm{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_m})$.

Q1 29/30

This representation ensures that for any pair $x_i, x_j \in X$,

$$\Big\langle \phi(x_i), \phi(x_j) \Big\rangle_{\mathcal{H}} = \phi(x_i)^\top \phi(x_j) = (U\Lambda^{1/2})_i^\top (U\Lambda^{1/2})_j = K_{ij},$$

thus satisfying the original kernel definition.

2 Kernel dependence detection

2.1 Part 1: Efficient COCO

Recall from the lecture notes that the solution to the COCO problem is given by:

$$\text{COCO} := \max_{f,g} \left\langle f, \hat{C}_{XY}\, g \right\rangle_{\mathcal{G}}$$

subject to:

$$\|f\|_{\mathcal{F}} = 1, \quad \|g\|_{\mathcal{G}} = 1$$

Here, $\hat{C}_{XY}$ is the empirical cross-covariance operator, and $\mathcal{F}$ and $\mathcal{G}$ are reproducing kernel Hilbert spaces (RKHS) associated with kernels K and L, respectively.

From the lecture notes, the problem reduces to solving the generalized eigenvalue problem:

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

where:

$$f = \sum_{i=1}^{n} \alpha_i [\phi(x_i) - \hat{\mu}_x] = \Phi H \alpha$$

$$g = \sum_{j=1}^{n} \beta_j [\psi(y_j) - \hat{\mu}_y] = \Psi H \beta$$

$$\tilde{K} = HKH, \quad \tilde{L} = HLH$$

and $H = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ is the centering matrix, with $\mathbf{1}_n \in \mathbb{R}^n$ being a vector of ones.

2.1.1 Approximation via Incomplete Cholesky Decomposition

To derive a more computationally efficient estimate of COCO, we approximate the kernel matrices K and L using incomplete Cholesky decomposition: ✓

$$K \approx RR^\top, \quad L \approx QQ^\top$$

where $R, Q \in \mathbb{R}^{n\times T}$ are the incomplete Cholesky factors, and $T \ll n$ is the number of pivots.

2.1.2 Centering After Decomposition

We apply centering to the approximated kernel matrices:

$$\tilde{K} = HKH \approx HRR^\top H = (HR)(HR)^\top = \tilde{R}\tilde{R}^\top$$
$$\tilde{L} = HLH \approx HQQ^\top H = (HQ)(HQ)^\top = \tilde{Q}\tilde{Q}^\top$$

where $\tilde{R} = HR \in \mathbb{R}^{n\times T}$ and $\tilde{Q} = HQ \in \mathbb{R}^{n\times T}$. ✓

2.1.3 Formulating the Approximate COCO Problem

Substituting the approximated centered kernel matrices into the generalized eigenvalue problem, we have:

$$\begin{bmatrix} 0 & \frac{1}{n}\tilde{R}\tilde{R}^\top\tilde{Q}\tilde{Q}^\top \\ \frac{1}{n}\tilde{Q}\tilde{Q}^\top\tilde{R}\tilde{R}^\top & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \gamma \begin{bmatrix} \tilde{R}\tilde{R}^\top & 0 \\ 0 & \tilde{Q}\tilde{Q}^\top \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

We can make this low rank approximation more efficient. To reduce dimensionality, we project the problem onto the subspace spanned by the columns of $\tilde{R}$ and $\tilde{Q}$.

2.1.4 Reducing the Dimensionality

Define new variables (similar solution as in Gretton et al (2005)[3]:

$$u = \tilde{R}^\top \alpha, \quad v = \tilde{Q}^\top \beta$$ ✓

4/3 (justify injectivity)

Solving for α and β in terms of u and v, we obtain:

$$\alpha = (\tilde{R}\tilde{R}^\top)^\dagger \tilde{R} u, \quad \beta = (\tilde{Q}\tilde{Q}^\top)^\dagger \tilde{Q} v$$ ✓

Here the $(\cdot)^\dagger$ denotes the Moore-Penrose pseudo-inverse.

2.1.5 Reformulating the Objective Function

The COCO objective aims to maximize the correlation between the projections f and g:

$$\text{COCO} := \max_{f,g} \left\langle f, \hat{C}_{XY}\, g \right\rangle_{\mathcal{G}} = \max_{\alpha,\beta} \frac{1}{n} \alpha^\top \tilde{K} \tilde{L} \beta$$

Using the low-rank approximations, the objective simplifies to:

$$\max_{\alpha,\beta} \quad \frac{1}{n} \alpha^\top \tilde{R}\tilde{R}^\top \tilde{Q}\tilde{Q}^\top \beta$$

Substituting α and β in terms of u and v, we got:

$$\max_{u,v} \quad \frac{1}{n} u^\top C_{RQ} v$$

where:

$$C_{RQ} = \tilde{R}^\top \tilde{Q}$$

2.1.6 Reformulating the Constraints

The constraints are derived from the norms of α and β:

$$\alpha^\top \tilde{K} \alpha = 1, \quad \beta^\top \tilde{L} \beta = 1$$

Substituting the low-rank approximations and expressions for α and β:

$$\alpha^\top \tilde{R}\tilde{R}^\top \alpha = u^\top C_{RR} u = 1$$

$$\beta^\top \tilde{Q}\tilde{Q}^\top \beta = v^\top C_{QQ} v = 1$$

where:

$$C_{RR} = \tilde{R}^\top \tilde{R}, \quad C_{QQ} = \tilde{Q}^\top \tilde{Q}$$

2.1.7 Reformulating COCO Optimization Problem

The optimization problem is thus:

$$\max_{u,v} \frac{1}{n} u^\top C_{RQ} v \quad \text{subject to} \quad u^\top C_{RR} u = 1, \quad v^\top C_{QQ} v = 1$$

2.1.8 The Lagrangian

To incorporate the constraints, we construct the Lagrangian:

$$\mathcal{L}(u, v, \gamma_u, \gamma_v) = \frac{1}{n} u^\top C_{RQ} v - \frac{\gamma}{2}(u^\top C_{RR} u - 1) - \frac{\lambda}{2}(v^\top C_{QQ} v - 1)$$

2.1.9 Deriving the Generalized Eigenvalue Problem

Taking partial derivatives of the Lagrangian with respect to u and v and setting them to zero yields:

$$\frac{\partial \mathcal{L}}{\partial u} = \frac{1}{n} C_{RQ} v - \gamma C_{RR} u = 0$$

$$\frac{\partial \mathcal{L}}{\partial v} = \frac{1}{n} C_{RQ}^\top u - \lambda C_{QQ} v = 0$$

Assuming $\gamma = \lambda$, we obtain:

$$\frac{1}{n} C_{RQ} v = \gamma C_{RR} u$$

$$\frac{1}{n} C_{RQ}^\top u = \gamma C_{QQ} v$$

✓

We have our GEP:

$$\begin{bmatrix} 0 & \frac{1}{n} C_{RQ} \\ \frac{1}{n} C_{RQ}^\top & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \gamma \begin{bmatrix} C_{RR} & 0 \\ 0 & C_{QQ} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

This constitutes a generalised eigenvalue problem of size $2T \times 2T$

We already got the solution we used in our implementation here but if we follow Gretton et al (2005)[3], we can simplify even further to a standrad eigenvalue problem of size $2T \times 2T$ by premultiply both sides by diag$[\tilde{Q}, \tilde{R}]$ (The matrix need not be square, however, and the diagonal is in this case defined in a manner consistent with $\tilde{Q}$ & $\tilde{R}$'s asymmetry) then eliminate diag$[\tilde{Q}\tilde{Q}^\top, \tilde{L}\tilde{L}^\top]$ (but we use the solution above for analysis of computation cost as this is what is implemented since this is just as efficient computationally):

2.1.10 Implications for the Witness Functions f and g

The incomplete Cholesky decomposition, interpreted as an incomplete Gram-Schmidt procedure, facilitates the projection of the feature maps onto a lower-dimensional subspace spanned by the pivot points. This projection make computing functions f and g more efficient.

Expressing f and g in the Reduced Subspace The witness functions are originally defined as:

$$f = \Phi H \alpha, \quad g = \Psi H \beta$$

Substituting the expressions for α and β in terms of the reduced variables u and v:

$$\alpha = (\tilde{R}\tilde{R}^\top)^\dagger \tilde{R} u, \quad \beta = (\tilde{Q}\tilde{Q}^\top)^\dagger \tilde{Q} v$$

we obtain:

$$f = \Phi (\tilde{R}\tilde{R}^\top)^\dagger \tilde{R} u = \Phi \tilde{R}^\dagger u$$
$$g = \Psi (\tilde{Q}\tilde{Q}^\top)^\dagger \tilde{Q} v = \Psi \tilde{Q}^\dagger v$$

Reconstructing the Witness Functions After solving the GEP for the reduced variables u and v, the corresponding witness functions f and g are reconstructed as: ✓ 5/5

$$f_k = \Phi \tilde{R}^\dagger u_k, \quad g_k = \Psi \tilde{Q}^\dagger v_k$$

for each canonical pair (f_k, g_k), where u_k and v_k are the eigenvectors associated with the eigenvalue γ_k.

2.1.11 Computational Cost Comparison

- **Exact Computation of COCO**
  - **Step 1: Compute the Kernel Matrices K and L**
    - **Operation**: Calculate the Gram matrices using the kernel functions $k(x_i, x_j)$ and $l(y_i, y_j)$ for all pairs of training points.
    - **Computational Cost**:

$$O(n^2) \quad \text{operations per kernel matrix}$$

  - **Step 2: Center the Kernel Matrices**
    - **Operation**: Apply the centering matrix H to center K and L:

$$\tilde{K} = HKH, \quad \tilde{L} = HLH$$

✓ 5/5

    - **Computational Cost**:

$$O(n^2) \quad \text{operations per centered kernel matrix}$$

  - **Step 3: Formulate the Generalized Eigenvalue Problem (GEP)**
    - **Operation**: Construct the GEP:

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

    - **Computational Cost**:

$$O(n^3) \quad \text{operations}$$

  - **Step 4: Solve the Generalized Eigenvalue Problem**
    - **Operation**: Compute the largest eigenvalues and corresponding eigenvectors of the GEP.
    - **Computational Cost**:

$$O(n^3) \quad \text{operations}$$

  - **Step 5: Compute the Witness Functions f and g**
    - **Operation**: Calculate the witness functions using:

$$f = \Phi H\alpha, \quad g = \Psi H\beta$$

    - **Computational Cost**:

$$O(n^2) \quad \text{operations per witness function}$$

  - **Total Computational Cost for Exact COCO**
    - **Dominant Cost**:

$$O(n^3) \quad \text{operations}$$

- **Approximate Computation via Incomplete Cholesky Decomposition**
  - **Interpretation of Incomplete Cholesky**
    - **Understanding**: The incomplete Cholesky decomposition acts as an incomplete Gram-Schmidt procedure, projecting the feature maps $\{\phi(x_i)\}_{i=1}^n$ onto a subspace spanned by a subset of pivots $\{\phi(x_j)\}_{j\in I}$, where $I \subset \{1, \ldots, n\}$ and $|I| = T \ll n$.
  - **Step 1: Perform Incomplete Cholesky Decomposition**
    - **Operation**: Approximate the kernel matrices K and L as low-rank matrices:

$$K \approx RR^\top, \quad L \approx QQ^\top$$

      where $R, Q \in \mathbb{R}^{n\times T}$.

    - **Computational Cost**:

$$O(nT^2) \quad \text{operations per kernel matrix}$$

  - **Step 2: Center the Low-Rank Approximations**
    - **Operation**: Apply the centering matrix H to the low-rank factors:

$$\tilde{R} = HR, \quad \tilde{Q} = HQ$$

    - **Computational Cost**:

$$O(nT) \quad \text{operations per centered matrix}$$

  - **Step 3: Compute Cross-Covariance and Covariance Matrices**
    - **Operation**: Calculate the following matrices:

$$C_{RQ} = \tilde{R}^\top \tilde{Q}, \quad C_{RR} = \tilde{R}^\top \tilde{R}, \quad C_{QQ} = \tilde{Q}^\top \tilde{Q}$$

    - **Computational Cost**:

$$O(nT^2) \quad \text{operations}$$

  - **Step 4: Formulate the Reduced Generalized Eigenvalue Problem (GEP)**
    - **Operation**: Construct the reduced GEP of size $2T \times 2T$:

$$\begin{bmatrix} 0 & \frac{1}{n}C_{RQ} \\ \frac{1}{n}C_{RQ}^\top & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \gamma \begin{bmatrix} C_{RR} & 0 \\ 0 & C_{QQ} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

✓ v. detailed!

    - **Computational Cost**:

$$O(T^2) \quad \text{operations}$$

  - **Step 5: Solve the Reduced Generalized Eigenvalue Problem**
    - **Operation**: Determine the eigenvalues and eigenvectors of the reduced GEP.
    - **Computational Cost**:

$$O(T^3) \quad \text{operations}$$

  - **Step 6: Compute the Witness Functions f and g**
    - **Operation**: Reconstruct the witness functions using the reduced variables:

$$f = \Phi \tilde{R}^\dagger u, \quad g = \Psi \tilde{Q}^\dagger v$$

      where $\tilde{R}^\dagger$ and $\tilde{Q}^\dagger$ are the Moore-Penrose pseudo-inverses of $\tilde{R}$ and $\tilde{Q}$, respectively.
    - **Computational Cost**:

$$O(nT^2) \quad \text{operations per witness function}$$

  - **Total Computational Cost for Approximate COCO**
    - **Dominant Costs**:

$$O(nT^2 + T^3) \quad \text{operations}$$

    - **Assumption**: $T \ll n$, making nT^2 and T^3 significantly smaller than n^3.

- **Summary of Computational Costs**
  - **Exact COCO**
    - **Total Cost**:

$$O(n^3) \quad \text{operations}$$

  - **Approximate COCO via Incomplete Cholesky**
    - **Total Cost**:

$$O(nT^2 + T^3) \quad \text{operations}$$

  - **Comparison and Advantage**
    - The approximate method offers a significant reduction in computational cost, especially when $T \ll n$.
    - This makes the approximate approach more scalable and feasible for large datasets.

2.1.12 Incomplete Cholesky Based COCO Implementation

In our implementations we set the cut-off $\eta = 1 \times 10^{-}6$, the shape of matrix R is 5×300 which is a significant reduction to the size of K which is 300×300, so COCO with incomplete Cholesky reduces the complexity of the problem.

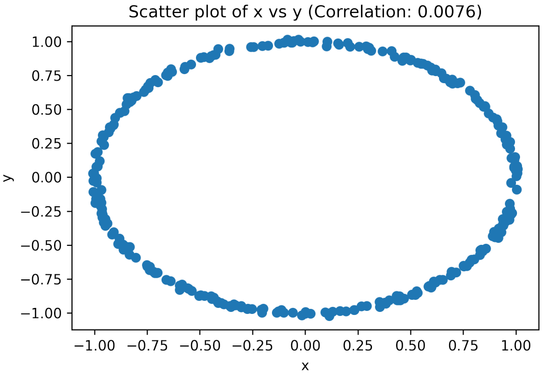
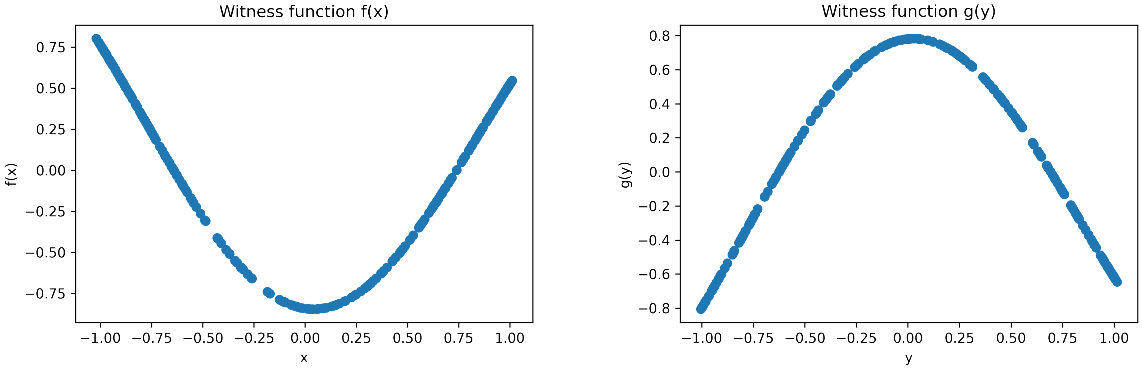


Figure 1: (x, y)



(a) (x, f(x))

(b) (y, g(y))

The map of f and g using incomplete Cholesky is displayed below with correlation we can see after the mapping our data is more dependent (0.0076 compares to 0.9504)

✓ 5/5 Implementation

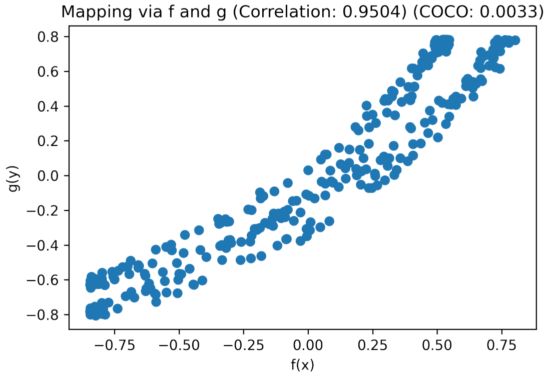


Figure 3: (f(x), g(y))

Q 2.1 19/20

2.2 Part 2: Regularized KCCA

2.2.1 Kernel CCA

We need to find:

$$\arg\max_{f,g} \left\langle f, \hat{C}_{XY} g \right\rangle_{\mathcal{G}},$$

Constrained to:

$$\left\langle f, \hat{C}_{XX} f \right\rangle_{\mathcal{F}} = 1 \text{ and } \left\langle g, \hat{C}_{YY} g \right\rangle_{\mathcal{G}} = 1$$

The Lagrangian:

$$\begin{aligned}
\mathcal{L}(f,g,\lambda,\gamma) &= \left\langle f, \hat{C}_{XY} g \right\rangle_{\mathcal{G}} - \frac{\lambda}{2}\left(\left\langle f, \hat{C}_{XX} f \right\rangle_{\mathcal{F}} - 1\right) - \frac{\gamma}{2}\left(\left\langle g, \hat{C}_{YY} g \right\rangle_{\mathcal{G}} - 1\right) \\
&= \frac{1}{n}\left\langle XH\alpha, XHY^\top YH\beta \right\rangle_{\mathcal{G}} - \frac{\lambda}{2}\left(\frac{1}{n}\left\langle XH\alpha, XHX^\top XH\alpha \right\rangle_{\mathcal{F}} - 1\right) - \frac{\gamma}{2}\left(\frac{1}{n}\left\langle YH\beta, YHY^\top YH\beta \right\rangle_{\mathcal{G}} - 1\right) \\
&= \frac{1}{n}(XH\alpha)^\top XHY^\top YH\beta - \frac{\lambda}{2}\left(\frac{1}{n}(XH\alpha)^\top XHX^\top XH\alpha - 1\right) - \frac{\gamma}{2}\left(\frac{1}{n}(YH\beta)^\top YHY^\top YH\beta - 1\right) \\
&= \frac{1}{n}\alpha^\top \tilde{K}\tilde{L}\beta - \frac{\lambda}{2}\left(\frac{1}{n}\alpha^\top \tilde{K}^2\alpha - 1\right) - \frac{\gamma}{2}\left(\frac{1}{n}\beta^\top \tilde{L}^2\beta - 1\right)
\end{aligned}$$

where: $H = H^\top(\text{Centering Matrix}), H = HH, \tilde{K} = HKH = HXX^\top H, \tilde{L} = HLH = HYY^\top H$

Taking derivatives with respect to α and β and setting them to zero:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \alpha} &= \tilde{K}\tilde{L}\beta - \lambda\tilde{K}^2\alpha = 0, \\
\frac{\partial \mathcal{L}}{\partial \beta} &= \tilde{L}\tilde{K}\alpha - \gamma\tilde{L}^2\beta = 0.
\end{aligned}$$

Solve the equations above we get $\lambda = \gamma$, hence the constrains become:

$$\begin{aligned}
\tilde{K}\tilde{L}\beta &= \lambda\tilde{K}^2\alpha = 0, \\
\tilde{L}\tilde{K}\alpha &= \lambda\tilde{L}^2\beta = 0.
\end{aligned}$$

which can we write equivalently as:

$$\begin{bmatrix} 0 & \tilde{K}\tilde{L} \\ \tilde{L}\tilde{K} & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \lambda \begin{bmatrix} \tilde{K}^2 & 0 \\ 0 & \tilde{L}^2 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \quad \checkmark$$

$$Ua = \lambda_i Va$$

Which is the generalised eigenvalue problem, let's set the largest eiginvalue to be λ^* the max CCA solution is obtained at $\frac{1}{n}\alpha^\top\tilde{K}\tilde{L}\beta = \frac{\lambda^*}{n}$:

$$\begin{aligned}
\operatorname{cov}[f(x), g(y)] &= \frac{1}{n}\alpha^\top\tilde{K}\tilde{L}\beta, \\
\operatorname{var}[f(x)] &= \frac{1}{n}\alpha^\top\tilde{K}^2\alpha, \\
\operatorname{var}[g(y)] &= \frac{1}{n}\beta^\top\tilde{L}^2\beta.
\end{aligned}$$

2.2.2 Regularizations

When points are also non-pathologically distributed so that K and L have full rank, which will always be the case with a Gaussian kernel, when we do not enforce regularizations, the non-zero solution of the our generalised eiginvalue problem becomes:

$$\lambda_i = \pm 1, \forall a_i$$

proof: Our problem is equivalent to:

✓ v. good, but can derive this!

$$\hat{\rho}_{\mathcal{F}} = \max_{\alpha,\beta} \frac{\alpha^\top \tilde{K}\tilde{L}\beta}{(\alpha^\top \tilde{K}^2\alpha)^{\frac{1}{2}}(\beta^\top \tilde{L}^2\beta)^{\frac{1}{2}}} = \max_{\alpha,\beta} \cos(\tilde{K}\alpha, \tilde{L}\beta)$$

4/5

This is because if K and L are full rank, then $\tilde{K} = HKH$ and $\tilde{L} = HLH$ are a both subspace orthogonal to the vector with all ones. Therefore the elements in the denominator $\alpha^\top \tilde{K}^2\alpha$ and $\beta^\top \tilde{L}^2\beta$ are the same. As a result regardless of α and β the value of the cosine above can only be ± 1. So for every α there exists β such that they are correlated perfectly. To prevent overfitting, we implement the regularizations. the updated constraints are:

$$\mathrm{var}[f(x)] = \left\langle f, \hat{C}_{XX} f \right\rangle_{\mathcal{F}} + \kappa \|f\|_{\mathcal{F}}^2 = \alpha^\top(\tilde{K}^2 + \kappa\tilde{K})\alpha = 1$$

$$\mathrm{var}[g(y)] = \left\langle g, \hat{C}_{YY} g \right\rangle_{\mathcal{G}} + \kappa \|g\|_{\mathcal{G}}^2 = \beta^\top(\tilde{L}^2 + \kappa\tilde{L})\beta = 1$$

The problem becomes:

$$\begin{aligned} \max_{\alpha,\beta} \quad & \alpha^\top \tilde{K}\tilde{L}\beta, \\ \text{subject to} \quad & \alpha^\top(\tilde{K}^2 + \kappa\tilde{K})\alpha - 1 = 0, \\ & \beta^\top(\tilde{L}^2 + \kappa\tilde{L})\beta - 1 = 0. \end{aligned}$$

To find the stationary points, we update the Lagrangian:

$$\mathcal{L}(\alpha,\beta,\lambda) = \alpha^\top \tilde{K}\tilde{L}\beta - \frac{\lambda}{2}\left(\alpha^\top(\tilde{K}^2 + \kappa\tilde{K})\alpha - 1\right) - \frac{\lambda}{2}\left(\beta^\top(\tilde{L}^2 + \kappa\tilde{L})\beta - 1\right).$$

Taking derivatives with respect to α and β and setting them to zero:

$$\frac{\partial \mathcal{L}}{\partial \alpha} = \tilde{K}\tilde{L}\beta - \lambda(\tilde{K}^2 + \kappa\tilde{K})\alpha = 0,$$
$$\frac{\partial \mathcal{L}}{\partial \beta} = \tilde{L}\tilde{K}\alpha - \lambda(\tilde{L}^2 + \kappa\tilde{L})\beta = 0.$$

✓ 5/5

These equations can be combined into a single generalized eigenvalue problem. Let $z = \begin{bmatrix}\alpha \\ \beta\end{bmatrix}$, then:

$$\begin{bmatrix} 0 & \frac{1}{n}\tilde{K}\tilde{L} \\ \frac{1}{n}\tilde{L}\tilde{K} & 0 \end{bmatrix} \begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \lambda \begin{bmatrix} \tilde{K}^2 + \kappa\tilde{K} & 0 \\ 0 & \tilde{L}^2 + \kappa\tilde{L} \end{bmatrix} \begin{bmatrix}\alpha \\ \beta\end{bmatrix}.$$

We have our CCA solution. Note that this regularsation inherits the independence characterization property of the origin problem. To estimate it from a finite sample, we approximate $\tilde{K}^2 + \kappa\tilde{K} \text{and} \tilde{L}^2 + \kappa\tilde{L}$ by the following [1] (We diverge from the Tikhonov regularization used in Standard KCCA in favor of the regularization used in Bach and Jordan (2002)):

$$\tilde{K}^2 + \kappa\tilde{K} \approx (\tilde{K} + \frac{n\kappa}{2}I)^2$$

$$\tilde{L}^2 + \kappa\tilde{L} \approx (\tilde{L} + \frac{n\kappa}{2}I)^2$$

So the actual generalized eigenvalue problem we solve is (it is well behaved computationally):

$$\begin{bmatrix} 0 & \frac{1}{n}\tilde{K}\tilde{L} \\ \frac{1}{n}\tilde{L}\tilde{K} & 0 \end{bmatrix} \begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \lambda \begin{bmatrix} (\tilde{K} + \frac{n\kappa}{2}I)^2 & 0 \\ 0 & (\tilde{L} + \frac{n\kappa}{2}I)^2 \end{bmatrix} \begin{bmatrix}\alpha \\ \beta\end{bmatrix}.$$

This regularsation have another neat feature, that is as $n \to \infty$ the estimate converge to the probability to the population quantity

2.2.3 KCCA Implementation

The map of f and g using Kernel CCA is displayed below with correlation we can see after the mapping our data is more dependent (0.0076 compares to 0.9975) We are using the same data as in the COCO with incomplete Cholesky

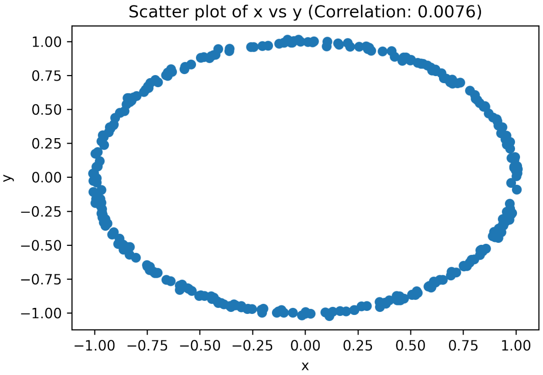
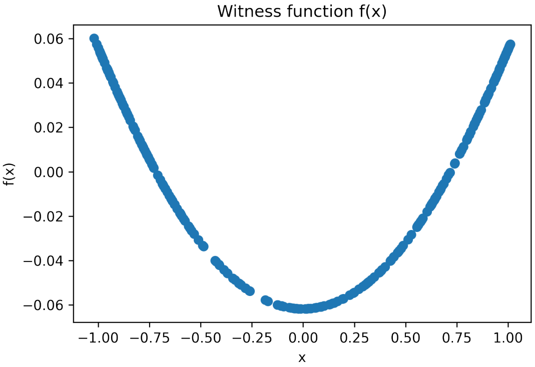
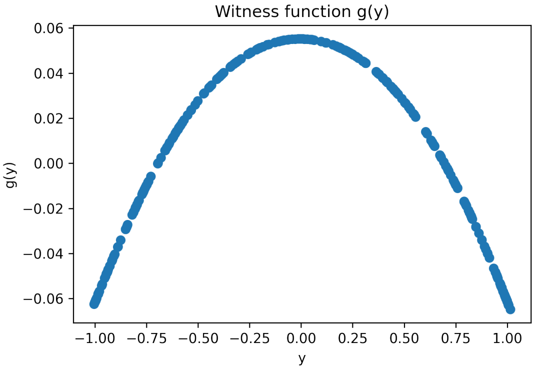


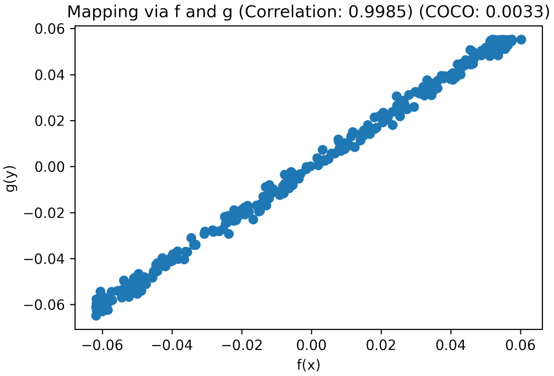
Figure 4: (x, y)



(a) (x, f(x))



(b) (y, g(y))



✓ 5/5

Figure 6: (f(x), g(y))

Q2.2

19/20

✓

2.2.4 Comparison: Regularized KCCA vs. Incomplete Cholesky Based COCO

We compare the incomplete Cholesky based COCO and KCCA. we can see that the range of functions generated by COCO is a bit wider than the one generated by KCCA. This is the because the constraints are different in COCO and KCCA and our KCCA is regularized, the low-rank approximation inherent in the incomplete Cholesky decomposition restricts the feature space and, consequently,the expressiveness of the witness functions. More over, COCO tends to find functions with large variance for f(X) and g(Y), which in many case may not be the most correlated features[2]. While KCCA focuses on finding linear correlations in the kernel-transformed feature space, Resulting in KCCA being more accurate. But the computational cost for running KCCA is higher than COCO making it harder to scale and slow to run in higher dimensions. The functions generated by COCO and KCCA have very similar shapes. The witness functions in KCCA is smoother and show more consistently high correlations across different parts of the data range than incomplete Cholesky based COCO. The withness funtions in incomplete Cholesky based COCO is slightly more different in shapes than in KCCA. We can see that in Fig 6. KCCA gives linearly correlated feature vectors while COCO's is less linear. Although the COCO also give the features that contain the sufficient information on the dependency between X and Y

References

[1] Francis R. Bach and Michael I. Jordan. “Kernel independent component analysis”. In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP ’03).* 4 (2003), pp. IV–876. URL: `https://jmlr.org/papers/v3/bach02a.html`.

[2] Kenji Fukumizu, Francis R. Bach, and Arthur Gretton. “Statistical Consistency of Kernel Canonical Correlation Analysis”. In: *Journal of Machine Learning Research* 8.14 (2007), pp. 361–383. URL: `http://jmlr.org/papers/v8/fukumizu07a.html`.

[3] Arthur Gretton et al. “Kernel Methods for Measuring Independence”. In: *Journal of Machine Learning Research* 6.70 (2005), pp. 2075–2129. URL: `http://jmlr.org/papers/v6/gretton05a.html`.