1 Feature spaces

1. In the 2 dimensional space, an error-free non-linear classifier is displayed as the black circle below:

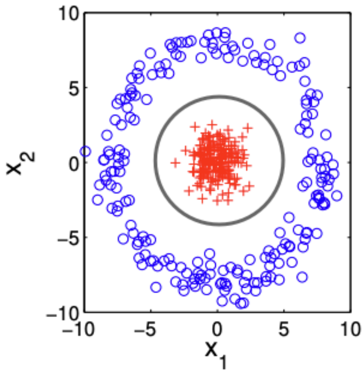

Figure 1: Ring dataset in 2D space with a non-linear classifier

In order to get a linear classifier, we need to add a dimension, which is the distance between the origin and the data points, expressed as $x_1^2 + x_2^2$.

Thus $\phi(x) = [x_1 \quad x_2 \quad x_1^2 + x_2^2]^\top$.

2. By eigendecomposition, $K = PDP^{-1} = PDP^\top$ since K is symmetric, where $P = [\mathbf{v}_1, \cdots, \mathbf{v}_m]$ is the eigenvector matrix and D is the diagonal matrix containing eigenvalues λ_i of K with $i \in \{1, \cdots, m\}$. Since K is positive semidefinite, $D_{ii} = \lambda_i \geq 0$. Thus we can take square root to D:

$$K = PDP^\top = P\sqrt{D}\sqrt{D}^\top P^\top = P\sqrt{D}(P\sqrt{D})^\top = QQ^\top\,, \tag{1}$$

where $Q = P\sqrt{D} = [\sqrt{\lambda_1}\mathbf{v}_1, \cdots, \sqrt{\lambda_m}\mathbf{v}_m]$.

By definition:

$$\begin{aligned} K(x_i, x_j) &= \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}} \\ &= (QQ^\top)_{ij} \\ &= [\sqrt{\lambda_1}\mathbf{v}_1^{(i)}, \cdots, \sqrt{\lambda_m}\mathbf{v}_m^{(i)}][\sqrt{\lambda_1}\mathbf{v}_1^{(j)}, \cdots, \sqrt{\lambda_m}\mathbf{v}_m^{(j)}]^\top\,. \end{aligned} \tag{2}$$

Thus, we find the feature space representation of x_i: $\phi(x_i) = [\sqrt{\lambda_1}\mathbf{v}_1^{(i)}, \cdots, \sqrt{\lambda_m}\mathbf{v}_m^{(i)}]$

2 Kernel dependence detection

1. **Incomplete Cholesky for efficient COCO**

Define:

$$A = \begin{bmatrix} 0 & \frac{1}{n}\widetilde{K}\widetilde{L} \\ \frac{1}{n}\widetilde{K}\widetilde{L} & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} \widetilde{K} & 0 \\ 0 & \widetilde{L} \end{bmatrix}.$$

The computational cost of COCO mainly comes from matrices A and B.

The computational cost for calculating COCO exactly:

As we know, $\widetilde{K} = HKH$, $\widetilde{L} = HLH$, where all the matrices (K, L, H, $\widetilde{K}$, $\widetilde{L}$) are $n \times n$ matrices. The naive matrix multiplication of two $n \times n$ matrices involves n^3 times multiplications (n rows $\times$ n columns $\times$ n terms) and $n^2 \times (n-1)$ times addition (n rows $\times$ n columns $\times$ $n-1$ addition operations), thus it has complexity $O(n^3 + n^2(n-1)) = O(2n^3 - n^2)$.

For both $\widetilde{K}$ and $\widetilde{L}$, the complexity is $O(2(2n^3 - n^2)) = O(4n^3 - 2n^2)$, thus for the matrix B, the complexity is $O(2(4n^3 - 2n^2)) = O(8n^3 - 4n^2)$.

For both $\frac{1}{n}\widetilde{K}\widetilde{L}$ and $\frac{1}{n}\widetilde{L}\widetilde{K}$, there are $n^3 + n^2$ multiplications and $n^2(n-1)$ additions, thus the complexity is $O((n^3 + n^2) + n^2(n-1)) = O(2n^3)$. Thus for matrix A, the complexity is $O(2(2n^3)) = O(4n^3)$.

In total, the complexity of calculating both A and B is $O(12n^3 - 4n^2)$.

The computational cost for approximated COCO via incomplete Cholesky:

By incomplete Cholesky decomposition, $\widetilde{K} = HKH = H(RR^\top)H = H^\top R^\top RH = (RH)^\top RH$, where R is $t \times n$, H is $n \times n$. The complexity of RH is $O(tn^2 + tn(n-1)) = O(2tn^2 - tn)$. RH is a $t \times n$ matrix, thus the complexity computing $\widetilde{K}$ is $O(tn^2 + n^2(t-1) + 2tn^2 - tn) = O(4tn^2 - n^2 - tn)$. The complexity of computing $\widetilde{L}$ is the same as that of $\widetilde{K}$. Hence the complexity of matrix B is $O(2(4tn^2 - n^2 - tn)) = O(8tn^2 - 2n^2 - 2tn)$.

The complexity of $\frac{1}{n}\widetilde{K}\widetilde{L}$ is $O(n^3 + n^2 + n^2(n-1)) = O(2n^3)$. The complexity of $\frac{1}{n}\widetilde{L}\widetilde{K}$ is exactly the same. Thus for matrix A, the complexity is $O(2(2n^3)) = O(4n^3)$.

In total, the complexity of calculating both A and B is $O(8tn^2 - 2n^2 - 2tn + 4n^3)$.

We conclude that when $t < n$, the approximated COCO is more efficient than the exactly computed COCO with simpler complexity.

The plotted f and g are displayed below:

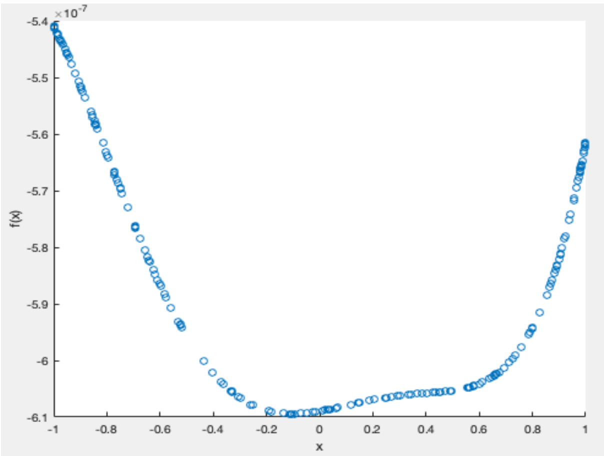

Figure 2: X vs f(x)

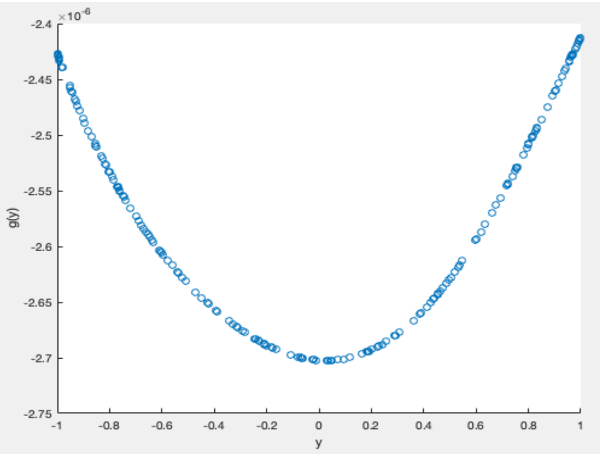

Figure 3: Y vs g(y)

The mapping of (x, y) pairs is displayed below, with correlation -0.87098:

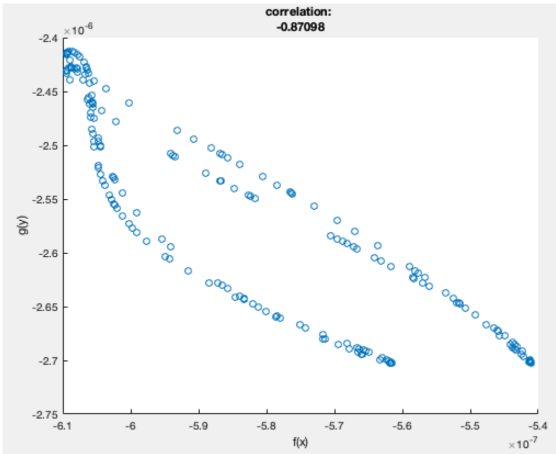

Figure 4: f(x) vs g(y)

The maps make our data more dependent.

2. **kernel CCA**

kernelized solution to CCA problem:

To

$$\arg\max_{f,g} \left\langle f, \hat{C}_{XY} g \right\rangle_{\mathcal{G}} , \tag{3}$$

subject to

$$\left\langle f, \hat{C}_{XX} f \right\rangle_{\mathcal{F}} = 1 \text{ and } \left\langle g, \hat{C}_{YY} g \right\rangle_{\mathcal{G}} = 1 \,, \tag{4}$$

we build Lagrangian as follow:

$$\begin{aligned}
\mathcal{L}(f,g,\lambda,\gamma) &= \left\langle f, \hat{C}_{XY} g \right\rangle_{\mathcal{G}} - \frac{\lambda}{2}\left(\left\langle f, \hat{C}_{XX} f \right\rangle_{\mathcal{F}} - 1\right) - \frac{\gamma}{2}\left(\left\langle g, \hat{C}_{YY} g \right\rangle_{\mathcal{G}} - 1\right) \\
&= \frac{1}{n}\left\langle XH\alpha, XHY^\top YH\beta \right\rangle_{\mathcal{G}} - \frac{\lambda}{2}\left(\frac{1}{n}\left\langle XH\alpha, XHX^\top XH\alpha \right\rangle_{\mathcal{F}} - 1\right) \\
&- \frac{\gamma}{2}\left(\frac{1}{n}\left\langle YH\beta, YHY^\top YH\beta \right\rangle_{\mathcal{G}} - 1\right) \\
&= \frac{1}{n}(XH\alpha)^\top XHY^\top YH\beta - \frac{\lambda}{2}\left(\frac{1}{n}(XH\alpha)^\top XHX^\top XH\alpha - 1\right) \\
&- \frac{\gamma}{2}\left(\frac{1}{n}(YH\beta)^\top YHY^\top YH\beta - 1\right) \\
&= \frac{1}{n}\alpha^\top \widetilde{K}\widetilde{L}\beta - \frac{\lambda}{2}\left(\frac{1}{n}\alpha^\top \widetilde{K}^2\alpha - 1\right) - \frac{\gamma}{2}\left(\frac{1}{n}\beta^\top \widetilde{L}^2\beta - 1\right)
\end{aligned} \tag{5}$$

since $\widetilde{K} = HKH = HX^\top XH$, $\widetilde{L} = HLH = HY^\top YH$, $H = HH$.

Differentiating wrt α and β and setting to 0, we get:

$$\begin{aligned}
\frac{1}{n}\widetilde{K}\widetilde{L}\beta - \frac{1}{n}\frac{\lambda}{2}2\widetilde{K}^2\alpha = 0 \rightarrow \widetilde{K}\widetilde{L}\beta = \lambda\widetilde{K}^2\alpha \\
\frac{1}{n}\widetilde{L}\widetilde{K}\alpha - \frac{1}{n}\frac{\gamma}{2}2\widetilde{L}^2\beta = 0 \rightarrow \widetilde{L}\widetilde{K}\alpha = \gamma\widetilde{L}^2\beta
\end{aligned} \tag{6}$$

By solving the above 2 equations, we get: $\lambda = \gamma$. Thus our above 2 equations become:

$$\begin{aligned}
\widetilde{K}\widetilde{L}\beta = \lambda\widetilde{K}^2\alpha \\
\widetilde{L}\widetilde{K}\alpha = \lambda\widetilde{L}^2\beta \,,
\end{aligned} \tag{7}$$

The linear system above is equivalent to:

$$\begin{aligned}
\begin{bmatrix} 0 & \widetilde{K}\widetilde{L} \\ \widetilde{L}\widetilde{K} & 0 \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} = \lambda_i \begin{bmatrix} \widetilde{K}^2 & 0 \\ 0 & \widetilde{L}^2 \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \\
Ua_i = \lambda_i V a_i \,,
\end{aligned} \tag{8}$$

which is the generalised eigenvalue problem. The CCA solution is obtained at $\max \lambda_i$.

When the points are non-pathologically distributed so that K and L have full rank, without regularisation, the non-zero solutions to the generalised eigenvalue problem then becomes $\lambda_i = \pm 1$, regardless of a_i.

brief proof:

Our problem is equivalent to:

$$\arg\max_{f,g} \frac{\mathrm{cov}([f(x), g(y)])}{\mathrm{var}(f(x))^{\frac{1}{2}}\mathrm{var}(g(y))^{\frac{1}{2}}} = \arg\max_{\alpha,\beta} \frac{\alpha^\top \widetilde{K}\widetilde{L}\beta}{(\alpha^\top \widetilde{K}^2\alpha)^{\frac{1}{2}}(\beta^\top \widetilde{L}^2\beta)^{\frac{1}{2}}} = \arg\max_{\alpha,\beta} \cos(\widetilde{K}\alpha, \widetilde{L}\beta) \tag{9}$$

When K and L have full rank, $\widetilde{K} = HKH$ and $\widetilde{L} = HLH$ are a both subspace orthogonal to the vector with all ones. Thus $\alpha^\top \widetilde{K}^2\alpha$ and $\beta^\top \widetilde{L}^2\beta$ in the denominator are the same, and thus cosine is either 1 or -1, regardless α, β. $\square$

By adding regularisation terms to $\mathrm{var}(f(x))$ and $\mathrm{var}(g(y)$, we have updated constraints as follow:

$$\begin{aligned} \left\langle f, \hat{C}_{XX} f \right\rangle_{\mathcal{F}} + \kappa \|f\|_{\mathcal{F}}^2 = 1 \\ \left\langle g, \hat{C}_{YY} g \right\rangle_{\mathcal{G}} + \kappa \|g\|_{\mathcal{G}}^2 = 1\,. \end{aligned} \tag{10}$$

We know that $\|f\|_{\mathcal{F}}^2 = \langle f, f\rangle_{\mathcal{F}} = \alpha^\top \widetilde{K}\alpha$ and $\|g\|_{\mathcal{G}}^2 = \langle g, g\rangle_{\mathcal{G}} = \beta^\top \widetilde{L}\beta$, thus our updated Lagrangian becomes:

$$\mathcal{L}(f, g, \lambda, \gamma, \kappa) = \frac{1}{n}\alpha^\top \widetilde{K}\widetilde{L}\beta - \frac{\lambda}{2}\left(\alpha^\top \widetilde{K}^2\alpha + \kappa\alpha^\top \widetilde{K}\alpha - 1\right) - \frac{\gamma}{2}\left(\beta^\top \widetilde{L}^2\beta + \kappa\beta^\top \widetilde{L}\beta - 1\right) \tag{11}$$

Differentiating wrt α, β and setting to 0, we get:

$$\begin{aligned} \frac{1}{n}\widetilde{K}\widetilde{L}\beta - \lambda\left(\widetilde{K}^2\alpha + \kappa\widetilde{K}\alpha\right) = 0 \\ \frac{1}{n}\widetilde{L}\widetilde{K}\alpha - \gamma\left(\widetilde{L}^2\beta + \kappa\widetilde{L}\beta\right) = 0 \end{aligned} \tag{12}$$

By solving the above 2 equations, we get $\lambda = \gamma$, thus our linear system becomes:

$$\begin{aligned} \frac{1}{n}\widetilde{K}\widetilde{L}\beta = \lambda\left(\widetilde{K}^2\alpha + \kappa\widetilde{K}\alpha\right) \\ \frac{1}{n}\widetilde{L}\widetilde{K}\alpha = \lambda\left(\widetilde{L}^2\beta + \kappa\widetilde{L}\beta\right)\,, \end{aligned} \tag{13}$$

which is equivalent to:

$$\begin{bmatrix} 0 & \frac{1}{n}\widetilde{K}\widetilde{L} \\ \frac{1}{n}\widetilde{L}\widetilde{K} & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \lambda \begin{bmatrix} \widetilde{K}^2 + \kappa\widetilde{K} & 0 \\ 0 & \widetilde{L}^2 + \kappa\widetilde{L} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{14}$$

The f and g functions calculated by CCA are displayed below:

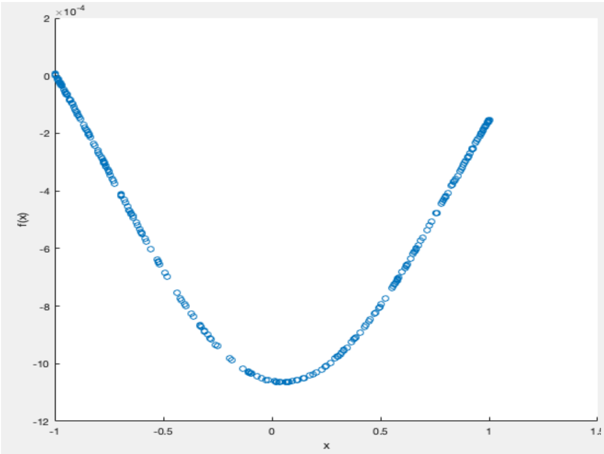

Figure 5: X vs f(x)

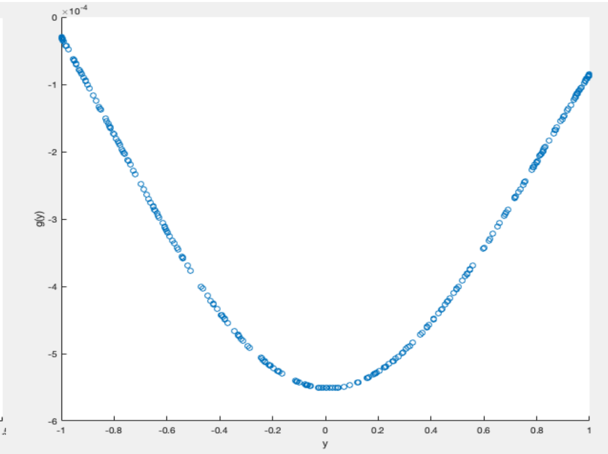

Figure 6: Y vs g(y)

The mapped data plot is shown below, with correlation -0.95211:

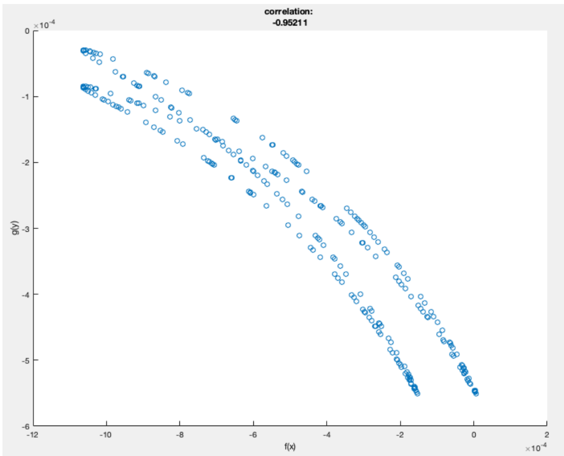

Figure 7: f(x) vs g(y)

The data also become more dependent after mapping, even more dependent than the ones obtained by COCO.

The functions obtained by CCA both have wider range (around 10^{-4}), while the functions computed by COCO have a much narrower range (around 10^{-7}). This difference is caused by different constraints in 2 cases. The computation of CCA is more accurate, however, it also has higher computational cost.