Karan Vora (kv2154)
Machine Learning Assignment 3

Problem 1):

Solution A):

Solution 1):

$k(x,\tilde{x}) = \alpha k_1(x,\tilde{x}) = \beta k_2(x,\tilde{x}) for\ \alpha,\beta \geqslant 0$

= $\langle \sqrt{\alpha}\Phi_1(x), \sqrt{\beta}\Phi_1(\tilde{x})\rangle + \langle \sqrt{\beta}\Phi_2(x), \sqrt{\beta}\Phi_2(\tilde{x})\rangle$

= $\langle [\sqrt{\alpha}\Phi_1(x)\sqrt{\beta}\Phi_2(x)], [\sqrt{\alpha}\Phi_1(\tilde{x})\sqrt{\beta}\Phi_2(\tilde{x})]\rangle$

Because k_1 and k_2 are valid kernels, thus $k_1 + k_2$ is a valid kernels and the obtained kernel is symmetric

Solution 2):

$k(x,\tilde{x}) = k_1(x,\tilde{x})k_2(x,\tilde{x})$

= $\langle \Phi_1(x), \Phi_1(\tilde{x})\rangle \langle \Phi_2(x), \Phi_2(\tilde{x})\rangle$ --- 1

= $\langle \Phi_1(x)\Phi_2(x), \Phi_1(\tilde{x})\Phi_2(\tilde{x})\rangle$ --- 2

Since 1 is a symmetric, positive and definitive matrix. Hence, 2 is a valid kernel function on $\Phi_1(X)\Phi_2(X)$

Solution 3):

$k(x,\tilde{x}) = f(k_1(x,\tilde{x}))$

Lets assume f is a polynomial function, the highest power it has is d. from solution 2, we know that k_1 is a valid kernel and from solution 1 we know that multiplying it with a coefficient yields a valid kernel thus the given kernel is also a valid kernel.

Solution 4):

$k(x,\tilde{x}) = \exp(k_1(x,\tilde{x}))$

from taylor's expansion of exponential series, we know that

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots\ldots\ldots$$

now, substituting $x = k_1(x,\tilde{x})$

$$= \quad 1+k_1(x,\tilde{x})+\frac{k_1(x,\tilde{x})^2}{2!}+\ldots\ldots$$

from solution 1 we know that k_1 is a valid kernel and multiplying it with a coefficient yields a valid kernel as well. From the same solution 1, we implied that addition of 2 kernels yields a valid kernel as well. Thus the given kernel is a valid kernel

Solution B):

Given kernel,

$K(x,y)=\exp(-\frac{1}{2}\|x-y\|^2)=\varphi(x).\varphi(y)$ write explicit formula for φ

The given kernel can be written as an inner product between mapping φ.

$$\varphi_p(x)=\frac{\Pi}{2}^{\frac{-d}{4}}\exp(-\|x-p\|^2)$$

which is infinite dimensional function for y, we have

$$\varphi_p(y)=\frac{\Pi}{2}^{\frac{-d}{4}}\exp(-\|y-p\|^2)$$

we can define the kernel as,

$$K(x,y)=\langle\varphi_p(x),\varphi_p(y)\rangle$$

$$= \quad \int \varphi_p(x)\varphi_p(y)$$

$$K(x,y)=\int_p \frac{\Pi}{2}^{\frac{-d}{4}}\exp(-\|x-p\|^2)\frac{\Pi}{2}^{\frac{-d}{4}}\exp(-\|y-p\|^2)dp$$

$$= \quad (\frac{\Pi}{2})^{-\frac{d}{2}}\int_p \exp(-x^T x-p^T p+2x^T p)\exp(-y^T y-p^T p+2y^T p)dp$$

$$= \quad (\frac{\Pi}{2})^{-\frac{d}{2}}\exp(-x^T x-y^T y)\int_p \exp(-2p^T p+2(y+x))dp$$

Now lets assume $w=\frac{x+y}{2}$

$$= \quad (\frac{\Pi}{2})^{-\frac{d}{2}}\exp(-x^T x-y^T y)\int_p \exp(-2p^T p+4w)dp$$

$$= \quad (\frac{\Pi}{2})^{-\frac{d}{2}}\exp(-x^T x-y^T y)\int_p \exp(-2p^T p+4w^T p)dp$$

$$= \quad (\frac{\Pi}{2})^{-\frac{d}{2}} \exp(-x^T x - y^T y) \exp(2 w^T w) \int_p \exp(-2 p^T p + 4 w^T p - 2 w^T w) dp$$

$$= \quad (\frac{\Pi}{2})^{-\frac{d}{2}} \exp(-x^T x - y^T y) \exp(2 w^T w) \int_p \exp(-2 \| p - w \|^2) dp$$

$$= \quad (\frac{\Pi}{2})^{-\frac{d}{2}} \exp(-x^T x - y^T y) \exp(2 w^T w) (\frac{\Pi}{2})^{\frac{d}{2}}$$

$$= \quad \exp(-x^T x - y^T y) \exp(\frac{1}{2} x^T x + \frac{1}{2} y^T y + x^T y)$$

$$= \quad \exp(-\frac{1}{2} x^T x - \frac{1}{2} y^T y + x^T y)$$

$$= \quad \exp(-\frac{1}{2} \| x - y \|^2), \textit{hence proved} .$$

==

Problem 2):

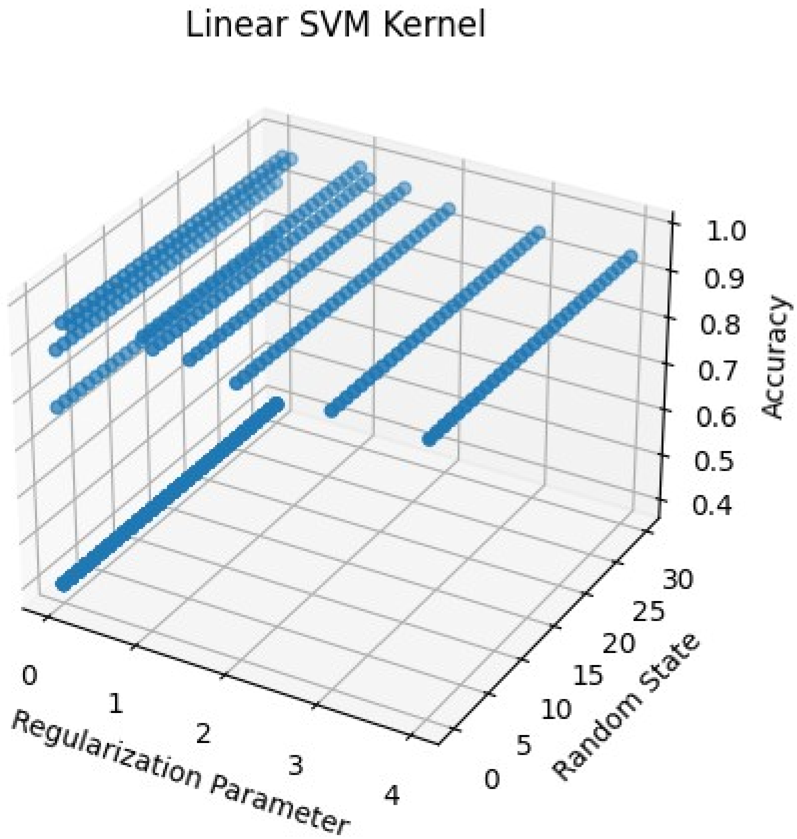
Linear Kernel,

Any hyper-plane can be written as the set of points **x** satisfying

$$w^T x - b = 0$$

where w is the normal vector to the hyperplane.

In code, while tweaking for hyper-parameters like Regularization Parameter and Random State, we get this plot distribution, We get and accuracy of 0.98 when the value of Regularization Parameter is 1.0 and Random State is 0.



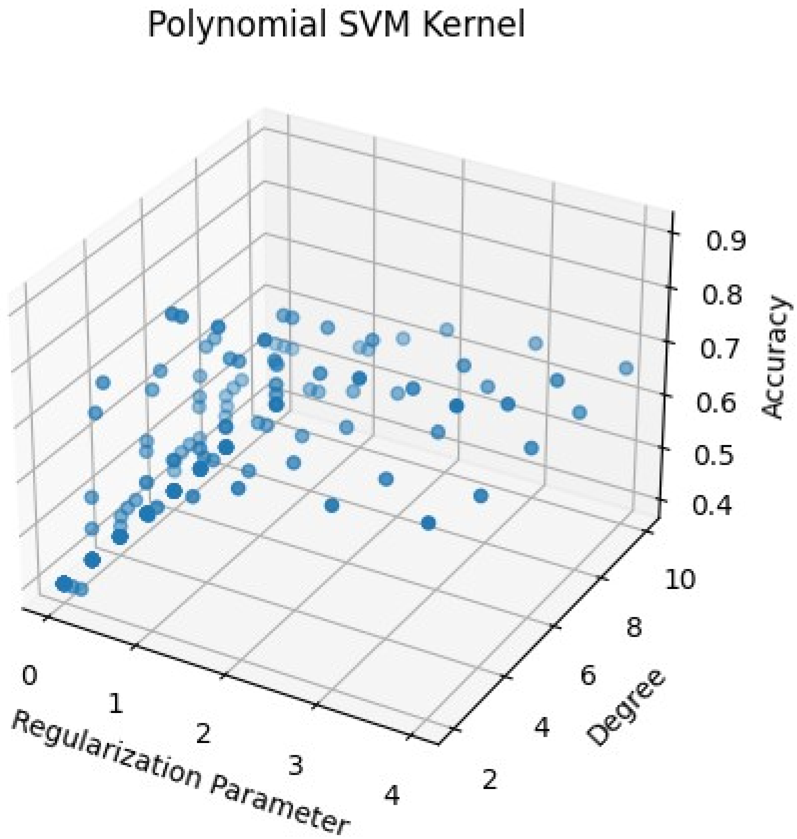
Polynomial Kernel,

For degree-d polynomials, the polynomial kernel is defined as,

$$K(x, y) = (x^T y + c)^d$$

where x and y are vectors in the input space, vectors of the features computed from training or test samples $c \geq 0$ is a free parameter trading off the influence of higher-order versus lower-order terms in the polynomial. When $c = 0$ the kernel is homogeneous.

In code, while tweaking for hyper-parameters like Regularization Parameter and Dimensionality, we get this plot distribution, We get and accuracy of 0.9 when the value of Regularization Parameter is 1.0 and Dimension is 3.



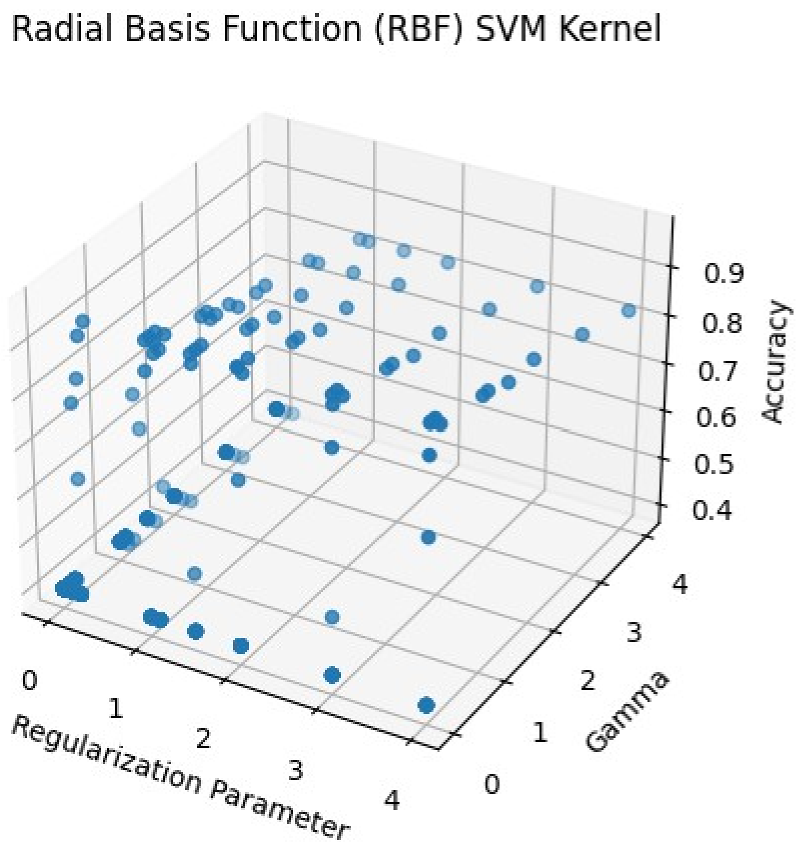
RBF Kernel,

The Radial Basis Function, or RBF Kernel on two samples $x \in \mathbb{R}^k$ and x', represented as feature vectors in some input space, is defined as

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

$\|x - x'\|^2$ may be recognized as the squared euclidean distance between the two feature vectors. σ is a free parameters.

In code, while tweaking for hyper-parameters like Regularization Parameter and Gamma, we get this plot distribution, We get and accuracy of 0.96 when the value of Regularization Parameter is 1.0 and Gamma is 0.02



Out of all, Linear SVM Kernel gives the highest performance seeing the limited information from the graph but this doesn't tell you the whole story, in real world, you cannot justify that a particular kernel is better over another. The selection of kernel for classification depends on other aspects such as compute overhead, latency etc.

===

Problem 3):

The function for pdf is $f(x|\alpha)=\alpha^{-2}x\,e^{\frac{-x}{\alpha}}$

$$= \frac{x}{\alpha^2 e^{\frac{x}{\alpha}}}$$

Now, the maximum likelihood is calculated by

$$L(\theta)=\prod_{i=1}^{n} f(x|\alpha)$$

$$L(\theta)=\prod_{i=1}^{n} \frac{x_i}{\alpha^2 e^{\frac{x_i}{\alpha}}}$$

$$L(\theta)=\frac{1}{\alpha^{2n}}\prod_{i=1}^{n}\frac{x_i}{e^{\frac{x_i}{\alpha}}}$$

The Log – Likelihood function of α

$$l(\alpha)=\log[\frac{1}{\alpha^{2n}}\prod_{i=1}^{n}\frac{x_i}{e^{\frac{x_i}{\alpha}}}]$$

$$= \quad \log(\frac{1}{\alpha^{2n}})+\log(\prod_{i=1}^{n}\frac{x_i}{e^{\frac{x_i}{\alpha}}})$$

$$= \quad -2n\log\alpha+\sum_{i=1}^{n}\log\frac{x_i}{e^{\frac{x_i}{\alpha}}}$$

$$= \quad -2n\log\alpha+\sum_{i=1}^{n}[\log x_i-\log e^{\frac{x_i}{\alpha}}]$$

$$l(\alpha)=-2n\log(\alpha)+\sum_{i=1}^{n}[\log x_i-\frac{x_i}{\alpha}]$$

$$\frac{\partial l(\alpha)}{\partial\alpha}=\frac{\partial}{\partial(\alpha)}[-2n\log(\alpha)+\sum_{i=1}^{n}[\log x_i-\frac{x_i}{\alpha}]]$$

$$= \quad \frac{-2n}{\alpha}+\sum_{i=1}^{n}[0-(-\frac{x_i}{\alpha^2})]$$

$$= \quad \frac{-2n}{\alpha}+\sum_{i=1}^{n}\frac{x_i}{\alpha^2}$$

The maximum likelihood estimator (MLE) of θ is that value of θ is that value of θ which maximizes the likelihood function. Since log(x) is an increasing function of x, therefore MLE of θ can also be obtained by maximizing the log – likelihood function l(θ).

$$\frac{\partial l(\theta)}{\partial\theta}=0$$

$$\frac{-2n}{\alpha}+\sum_{i=1}^{n}\frac{x_i}{\alpha^2}=0$$

$$\sum_{i=1}^{n}\frac{x_i}{\alpha^2}=\frac{2n}{\alpha}$$

$$\alpha=\frac{\sum_{i=1}^{n}x_i}{2n}$$

Now, $x_1 = 0.25, x_2 = 0.75, x_3 = 1.50, x_4 = 2.5, x_5 = 2.0$.

$$\alpha = \frac{0.25 + 0.75 + 1.50 + 2.50 + 2.0}{2 * 5}$$

$$\alpha = \frac{7}{10} = 0.7$$