CSCI567 2013 Homework Assignment 4

Student Name	Arpit Bansal
Last 4 digits of USC ID	0979
I have collaborated with	Jia Li Udit Agrawal Yin Ray Rick Huang

1 Gaussian Process

we are given kernel function $k(x_i, x_j) = x_i^T x_j + \lambda\delta_{ij}$ where $\delta_{ij} = 1$ if $i = j$ otherwise it will be zero. So we can write corresponding kernel matrix as

$$\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^T + \lambda\boldsymbol{I}$$

so the resulting prediction distribution as per from lecture notes is

$$\begin{aligned} p(f(\boldsymbol{x})|f(\boldsymbol{x}_1) = y_1, ...f(\boldsymbol{x}_N) = y_N) \\ &= N(\boldsymbol{k}_{\boldsymbol{x}}^T\boldsymbol{K}^{-1}\boldsymbol{y}, k(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{k}_{\boldsymbol{x}}^T\boldsymbol{K}^{-1}\boldsymbol{k}_{\boldsymbol{x}}) \\ &= N(\boldsymbol{x}^T\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{y}, \boldsymbol{x}^T\boldsymbol{x} + \lambda - \boldsymbol{x}^T\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}\boldsymbol{x}) \end{aligned}$$

Now lets look at the Bayes Regression when $\alpha = 1$ and $\beta = \frac{1}{\lambda}$. Predictive distribution for Bayes regression will be

$$\begin{aligned} p(y|\boldsymbol{x}, D, \alpha, \beta) &= N(\boldsymbol{x}^T\beta(\alpha\boldsymbol{I} + \beta\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}, \frac{1}{\beta} + \boldsymbol{x}^T(\alpha\boldsymbol{I} + \beta\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}) \\ &= N(\boldsymbol{x}^T(\lambda\boldsymbol{I} + \boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}, \lambda + \boldsymbol{x}^T(\boldsymbol{I} + \frac{1}{\lambda}\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}) \end{aligned}$$

Now, to prove that both above predictive distributions are same, we will try to equate μ and variance. So lets look at μ first. By applying the inverse matrix lemma as explained in lecture notes, we can get $(\lambda\boldsymbol{I} + \boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T = \boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T + \lambda\boldsymbol{I})^{-1}$, then we can easily see that $\boldsymbol{x}^T(\lambda\boldsymbol{I}+\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} = \boldsymbol{x}^T\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T+\lambda\boldsymbol{I})^{-1}\boldsymbol{y}$. So the μ for both Gaussian process and Bayesian regression are same.

Now, lets look at variance for both concepts. For the Gaussian Process

$$\begin{aligned} \text{Gaussian process variance} &= \boldsymbol{x}^T\boldsymbol{x} + \lambda - \boldsymbol{x}^T\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}\boldsymbol{x} \\ &= \boldsymbol{x}^T\boldsymbol{x} + \lambda - \boldsymbol{x}^T\boldsymbol{X}^T\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{x};\ \text{by applying the inverse matrix lemma} \\ &= \boldsymbol{x}^T\boldsymbol{x} + \lambda - \boldsymbol{x}^T(\boldsymbol{I} + (\frac{1}{\lambda}\boldsymbol{X}^T\boldsymbol{X})^{-1})^{-1}\boldsymbol{x} \\ &= \lambda + \boldsymbol{x}^T[\boldsymbol{I} - (\boldsymbol{I} + (\frac{1}{\lambda}\boldsymbol{X}^T\boldsymbol{X})^{-1})^{-1}]\boldsymbol{x} \\ &= \lambda + \boldsymbol{x}^T[(\boldsymbol{I} + (\frac{1}{\lambda}\boldsymbol{X}^T\boldsymbol{X})^{-1})^{-1}(\boldsymbol{I} + (\frac{1}{\lambda}\boldsymbol{X}^T\boldsymbol{X})^{-1} - \boldsymbol{I})]\boldsymbol{x} \\ &= \lambda + \boldsymbol{x}^T[(\boldsymbol{I} + (\frac{1}{\lambda}\boldsymbol{X}^T\boldsymbol{X})^{-1})^{-1}(\frac{1}{\lambda}\boldsymbol{X}^T\boldsymbol{X})^{-1}]\boldsymbol{x} \\ &= \lambda + \boldsymbol{x}^T(\boldsymbol{I} + \frac{1}{\lambda}\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x} \\ &= \text{Bayesian regression variance} \end{aligned}$$

So, for $\alpha = 1, \beta = \frac{1}{\lambda}$, the given kernel behaves same as Bayes regression.

2 Kernel Methods

2.1

We have seen in the lecture notes that kernel $(x_m^T x_n)^2$ for 2-D feature space corresponds to $\phi(x) = [x_1^2 \; x_2^2 \; \sqrt{2}x_1x_2]^T$. So if we look at kernel $(x_m^T x_n)^2$ for N-D feature space, it corresponds to $\phi(x) = [x_1^2 \; x_2^2 \ldots x_N^2 \; \sqrt{2}x_1x_2 \; \ldots \sqrt{2}x_1x_N \; \sqrt{2}x_2x_3 \ldots \sqrt{2}x_{N-1}x_N]^T$. Here we can see that this feature space contains all the second order combinations of original feature with $\sqrt{2}$ as a multiple of cross terms. For the given kernel, for d=2, we will get

$$(x_m^T x_n + c)^2 = (x_m^T x_n)^2 + 2c(x_m^T x_n) + c^2$$

For 2-D feature space, the corresponding non linear feature space $\phi(x)$ for this specific kernel will be,

$$\phi(x) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \\ \sqrt{2c}x_1 \\ \sqrt{2c}x_2 \\ c \end{bmatrix}$$

Here we can observe that there are only two extra terms from the polynomial kernel $(x_m^T x_n)^2$. So in corresponding $\phi(x)$, we added only 2 set of extra features i.e., one for inner product of scalar and features, and sencond as c only. Similarily, for N-D space, corresponding $\phi(x)$ is $[x_1^2 \; x_2^2 \ldots x_N^2 \; \sqrt{2}x_1x_2 \; \ldots \sqrt{2}x_1x_N \; \sqrt{2}x_2x_3 \ldots \sqrt{2}x_{N-1}x_N \; \sqrt{2c}x_1 \; \sqrt{2c}x_2 \; \ldots \sqrt{2c}x_N \; c]^T$.

Now, we can do binomial expansion of the given kernel $(x_m^T x_n + c)^d$,

$$(x_m^T x_n + c)^d = (x_m^T x_n)^d + \mathbf{C}_1^d \, (x_m^T x_n)^{d-1} \, c^1 + \cdots + \mathbf{C}_{d-1}^d \, (x_m^T x_n)^1 \, c^{d-1} + c^d. \tag{1}$$

then we can construct features for each order of inner product term as explained earlier, and add the square root of $\mathbf{C}_k^d \, c^k$ in front of dth order term.

2.2

The kernel matrix for $\alpha k_1(x_m, x_n) + \beta k_2(x_m, x_n)$ will be

$$K = \begin{pmatrix} \alpha k_1(x_1,x_1) + \beta k_2(x_1,x_1) & \alpha k_1(x_1,x_2) + \beta k_2(x_1,x_2) & \cdots & \alpha k_1(x_1,x_n) + \beta k_2(x_1,x_n) \\ \alpha k_1(x_2,x_1) + \beta k_2(x_2,x_1) & \alpha k_1(x_2,x_2) + \beta k_2(x_2,x_2) & \cdots & \alpha k_1(x_2,x_n) + \beta k_2(x_2,x_n) \\ \vdots & \vdots & \vdots & \vdots \\ \alpha k_1(x_n,x_1) + \beta k_2(x_n,x_1) & \alpha k_1(x_n,x_2) + \beta k_2(x_n,x_2) & \cdots & \alpha k_1(x_n,x_n) + \beta k_2(x_n,x_n) \end{pmatrix}$$

$$\Rightarrow \alpha \begin{pmatrix} k_1(x_1,x_1) & k_1(x_1,x_2) & \cdots & k_1(x_1,x_n) \\ k_1(x_2,x_1) & k_1(x_2,x_2) & \cdots & k_1(x_2,x_n)) \\ \vdots & \vdots & \vdots & \vdots \\ k_1(x_n,x_1) & k_1(x_n,x_2) & \cdots & k_1(x_n,x_n) \end{pmatrix} + \beta \begin{pmatrix} k_2(x_1,x_1) & k_2(x_1,x_2) & \cdots & k_2(x_1,x_n) \\ k_2(x_2,x_1) & k_2(x_2,x_2) & \cdots & k_2(x_2,x_n)) \\ \vdots & \vdots & \vdots & \vdots \\ k_2(x_n,x_1) & k_2(x_n,x_2) & \cdots & k_2(x_n,x_n) \end{pmatrix}$$

$$\Rightarrow \alpha K_1 + \beta K_2$$

where K_1 and K_2 are kernel matrices of kernel functions $k_1(x_m, x_n)$ and $k_2(x_m, x_n)$ respectively. We know that K_1 and K_2 are symmetric and PSD matrices. Now, if we multiply any symmetric and PSD matrix with a nonnegative number, or add two such matrices, the output is also a symmetric PSD matrix. So above matrix i.e. $\alpha K_1 + \beta K_2$ is also a symmetric PSD matrix if α,β are nonnegative. So $\alpha k_1(x_m, x_n) + \beta k_2(x_m, x_n)$ is also a kernel function.

2.3

Suppose the given function is a kernel function, then the kernel matrix for it will be

$$K = \begin{bmatrix} k_1(x_1,x_1)k_2(x_1,x_1) & k_1(x_1,x_2)k_2(x_1,x_2) & \cdots & k_1(x_1,x_N)k_2(x_1,x_N) \\ k_1(x_2,x_1)k_2(x_2,x_1) & k_1(x_2,x_2)k_2(x_2,x_2) & \cdots & k_1(x_2,x_N)k_2(x_2,x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k_1(x_N,x_1)k_2(x_N,x_1) & k_1(x_N,x_2)k_2(x_N,x_2) & \cdots & k_1(x_N,x_N)k_2(x_N,x_N) \end{bmatrix}$$

The above matrix is called a Hadamard product as it is element wise product of two kernel matrices K_1 and K_2 which are formed using the kernel functions $k_1(x_m, x_n)$ and $k_2(x_m, x_n)$ respectively. Lets take $\phi_1(x)$, $\phi_2(x)$ and $\phi(x)$ be feature space for K_1, K_2 and K respectively. We know that $k(x_i, x_j) = k_1(x_i, x_j)k_2(x_i, x_j)$. So now we have to prove for any particular vector v,

$$v^T K v \geq 0$$

We can also write left hand side of above equation as below

$$v^T K v = \sum_{l,k} v_l v_k k_1(x_l, x_k) k_2(x_l, x_k)$$

We can also see here that

$$k_1(x_m, x_n)k_2(x_m, x_n) = \phi_1(x_m)^T \phi_1(x_n) \phi_2(x_m)^T \phi_2(x_n)$$

$$\Rightarrow (\sum_i \phi_1(x_m)_i \phi_1(x_n)_i)(\sum_j \phi_2(x_m)_j \phi_2(x_n)_j) = \sum_{i,j} \phi_1(x_m)_i \phi_1(x_n)_i \phi_2(x_m)_j \phi_2(x_n)_j$$

So by putting above multiplication value in $v^T K v$, we get

$$\begin{aligned} \boldsymbol{v}^T K \boldsymbol{v} &= \sum_{l,k} v_l v_k \sum_{i,j} \phi_1(x_m)_i \phi_1(x_n)_i \phi_2(x_m)_j \phi_2(x_n)_j \\ &= \sum_{i,j} \sum_{l,k} v_l v_k \phi_1(x_l)_i \phi_1(x_k)_i \phi_2(x_l)_j \phi_2(x_k)_j \\ &= \sum_{i,j} (\sum_l v_l \phi_1(x_l)_i \phi_2(x_l)_j)(\sum_k v_k \phi_1(x_k)_i \phi_2(x_k)_j) \\ &= \sum_{i,j} (\sum_l v_l \phi_1(x_l)_i \phi_2(x_l)_j)^2 \geq 0 \end{aligned}$$

So we proved that matrix K is PSD, so $k(x_m, x_n) = k_1(x_m, x_n)k_2(x_m, x_n)$ is a kernel function.

2.4

We are given $\psi(x) = \phi(x) - \frac{1}{N}\sum_n \phi(x_n)$. Let's compute inner product using this function,

$$\psi(x_m)^T\psi(x_n) = \phi(x_m)^T\phi(x_n) - \frac{1}{N}\sum_i \phi(x_m)^T\phi(x_i) - \frac{1}{N}\sum_i \phi(x_i)^T\phi(x_n) + \frac{1}{N^2}\sum_i \phi(x_i)^T\sum_j \phi(x_j)$$

$$\Rightarrow k(x_m, x_n) - \frac{1}{N}\sum_i k(x_m, x_i) - \frac{1}{N}\sum_i k(x_i, x_n) + \frac{1}{N^2}\sum_{i,j} k(x_i, x_j)$$

Now we can write the above form into a matrix format as below

$$\begin{aligned} G &= K - (\frac{1}{N}O)K - K(\frac{1}{N}O) + (\frac{1}{N}O)K(\frac{1}{N}O) \\ &= (I - \frac{1}{N}O)K(I - \frac{1}{N}O) \end{aligned}$$

So our Gram matrix G will be the inner products of all the mapped non-linear features which is defined as above equation. Here K is kernel matrix produced by the kernel function $k(x_m, x_n)$ i.e. original features and, the centering matrix is

$$H = I - \frac{1}{N}O = I - \frac{1}{N}11^T$$

where *1* is column vector of 1's of size N. The kernel matrix K is centered because HKH substracts the mean of columns and rows from each corresponding elements of K. Also, to prove the rank of H is N-1, we can see that H^2 is equal to H. So it means it is an idempotent matrix and its eigen values will always be either 0 or 1. We know that rank of matrix is equal to number of non-zero eigen values. So we can calculate its rank using the trace of the matrix which is equal to sum of the diagonal elements i.e.,

$$rank(H) = tr(H) = \sum_1^N (1 - \frac{1}{N}) \Rightarrow N - 1$$

2.5

In previous question, we have already defined the definition of centering matrix *H*. So for some matrix D, HD removes the mean of columns from each of corresponding column element, while DH removes the mean of rows from each of corresponding row element. Here, the given distance function can be represented as elements' inner product as below,

$$d(x_i, x_j) =< x_i, x_i > + < x_j, x_j > -2 < x_i, x_j >$$

Now, for computing HDH, lets first compute DH. First calculate the average for i^{th} row over j:

$$\frac{1}{N}\sum_{j=1..N} d_{i,j} =< x_i, x_i > + \frac{1}{N}\sum_{k=1..N} < x_k, x_k > - 2 < x_i, \bar{x} >$$

Now update d_{x_i,x_j} for some i by subtracting the above calculated average. This will give us matrix DH or $D^{'}$.

$$d^{'}_{i,j} =< x_j, x_j > -2 < x_i, x_j > -\frac{1}{N}\sum_{k=1..N} < x_k, x_k > + 2 < x_i, \bar{x} >$$

Then, we will calculate the average for each column over i to compute $HD^{'}$,

$$\frac{1}{N}\sum_{i=1..N} d^{'}_{i,j} =< x_j, x_j > -2 < \bar{x}, x_j > -\frac{1}{N}\sum_{k=1..N} < x_k, x_k > + 2 < \bar{x}, \bar{x} >$$

Now, subtract above calculated average for corresponding column element and we get HDH, the (i,j) element of it is

$$-2 < x_i, x_j > +2 < x_i, \bar{x} > +2 < \bar{x}, x_j > -2 < \bar{x}, \bar{x} > \quad \Rightarrow \quad -2 < x_i - \bar{x}, x_j - \bar{x} >$$

So HDH is a matrix with inner products of original features like above equation. As $G = -HDH$, where $G_{i,j} = 2 < x_i - \bar{x}, x_j - \bar{x} >$. Hence we can write $\phi(x_i) = x_i - \bar{x}$. So,

$$G = 2\ times\ of\ kernel\ matrix\ of\ \phi(x)$$

So we can say that, G is a positive semi definite matrix.

3 Support Vector Machines

3.1

We can write the SVM's dual form in matrix format as below

$$J(\alpha) = 1^T\alpha - \frac{1}{2}\alpha^T K\alpha$$

where *1* is column vector of 1's and K is kernel matrix formed using kernel function $k(x_m, x_n)$. Now, let's differentiate w.r.t. to α,

$$\frac{\partial J(\alpha)}{\partial \alpha} = 1 - \frac{1}{2}(K + K^T)\alpha = 1 - K\alpha$$

as K is a symmetric matrix. Now differentiating again,

$$\frac{\partial^2 J(\alpha)}{\partial \alpha^2} = -K \leq 0$$

We know that K is PSD, so $-K$ will be non-PSD. It means, double differentiation is non-positive which proves that given function is concave.

3.2

SVM with gaussian kernels and almost no regularization essentially classifies test data by the average class of training data weighted by their distance to the test data. When one add more regularization, the algorithm starts choosing support vectors.

If we are using a very small $\sigma^2 = min_{i \neq j}||x_i, x_j||_2^2/10$ for the gaussian kernel, then we get

$$\begin{aligned} k(x_m, x_n) &= \exp\{-||x_m - x_n||_2^2/\sigma^2\} \\ &\begin{cases} = 1 & \text{if } m = n; \\ \leq e^{-10} & \text{if } m \neq n. \end{cases} \end{aligned}$$

So the kernel matrix K has actually became an Identity matrix I. Then in the dual form of SVM we get
$\max_\alpha \sum_n \alpha_n - \sum_n \alpha_n^2 y_n^2 = \max_\alpha \sum_n \alpha_n - \sum_n \alpha_n^2$ st. $0 \leq \alpha_n \leq C$ and $\sum_n \alpha_n y_n = 0$
From constraint $\sum_n \alpha_n y_n = 0$, we can get $\sum_{y_n=1} \alpha_n y_n = \sum_{y_n=-1} \alpha_n y_n$
So in the final solution of α, it will be positive value, and only related to its corresponding y_n. If the count of negative sample and positive sample are equal in the training data, then the α for positive y and negative y will also be same.

Now let look at the decision function $f(x) = \sum_i \alpha_i y_i k(x, x_i)$, As the α is only related to its corresponding y_n, and all equal when the training data is balance, so the $f(x)$ can be treated as a weighted vote. As we using a very small σ^2, then the weighted vote will be dominated by the closest one. So it much like a nearest neighbour.

3.3

SVM relies upon weighted distance rather than simple distance. This makes SVM simpler and less computationally expensive than KNN. This effect can be understood by the following example. Suppose in KNN, a test point sees about 50% of the K points from class 1 close enough and other 50% of K points from class 2 very far, then it cannot make a decision that whether the point should fall in class 1 or class 2. However, SVM will classify this point to class 1 due to weighted distances. Also, KNN is prone to overfitting and costly in storage due to its non-parameteric approach. SVMs avoids overfitting by finding a hyperplane that maximizes the minimum distance from the hyperplane to the closest training point.

3.4

The primal form for given labeling function will be

$$min_{w,\epsilon_n} C \sum_n \epsilon_n + \frac{1}{2}||w||_2^2$$

$$s.t. \ 1 - y_n w^T \phi(x_n) \leq \epsilon_n, \forall n$$

$$\epsilon_n \geq 0, \forall n$$

So corresponding Lagrange function will be,

$$L_{w,\epsilon_n} = C \sum_n \epsilon_n + \frac{1}{2}||w||_2^2 + \sum_n \alpha_n \left(1 - y_n w^T \phi(x_n) - \epsilon_n\right) - \sum_n \lambda_n \epsilon_n \tag{2}$$

$$s.t. \ \alpha_n, \lambda_n \geq 0, \forall n$$

where α_n and λ_n are lagrangian multipliers. So minimizing w.r.t. w and ϵ_n,

$$\frac{\partial L}{\partial w} = w - \sum_n \alpha_n y_n \phi(x_n) = 0 \Rightarrow w = \sum_n y_n \alpha_n \phi(x_n)$$

$$\frac{\partial L}{\partial \epsilon_n} = C - \lambda_n - \alpha_n = 0 \Rightarrow C = \lambda_n + \alpha_n$$

After putting above two values in equation (1), we will get

$$G = \sum_n \alpha_n + \frac{1}{2}||\sum_n y_n \alpha_n \phi(x_n)||_2^2 - \sum_{m,n} \alpha_m \alpha_n y_m y_n \phi(x_m)^T \phi(x_n)$$

$$\Rightarrow G = \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} \alpha_m \alpha_n y_m y_n k(x_m, x_n) \tag{3}$$

$$s.t.\ C = \lambda_n + \alpha_n, \forall n \ and\ \alpha_n, \lambda_n \geq 0, \forall n$$

Now above dual form doesn't depend upon λ_n, so we can simplyfy the contraints as we did in lecture notes i.e., $0 \leq \alpha_n \leq C$.

4 Anomaly/Outlier Detection

4.1

The given probelm can be written as below

$$\min_{r,\boldsymbol{c},\boldsymbol{xi}} \frac{1}{2}r^2 + C\sum_n \xi_n$$

s.t. $||\phi(\boldsymbol{x}_n) - \boldsymbol{c}||_2^2 \leq r^2 + \xi_n$ and $\xi_n \geq 0, \forall n$

Then our Langrangian function will be

$$L = \frac{1}{2}r^2 + C\sum_n \xi_n + \sum_n \alpha_n[||\phi(\boldsymbol{x}_n) - \boldsymbol{c}||_2^2 - r^2 - \xi_n] - \sum_n \lambda_n \xi_n$$

Now, minimizing over $r, \boldsymbol{c}, \xi_n$ we get

$$\begin{aligned}\frac{\partial L}{\partial r} &= r - 2r\sum_n \alpha_n = 0 \\ \frac{\partial L}{\partial c} &= -2\sum_n \alpha_n \phi(\boldsymbol{x}_n) + 2c\sum_n \alpha_n = 0 \\ \frac{\partial L}{\partial \xi_n} &= C - \alpha_n - \lambda_n = 0\end{aligned}$$

For an optimal solution, by using KKT conditions we get

$$\begin{aligned}\alpha_n[||\phi(\boldsymbol{x}_n - \boldsymbol{c})||_2^2 - r^2 - \xi_n] &= 0, \forall n \\ \lambda_n \xi_n &= 0, \forall n\end{aligned}$$

As we know $r > 0$, $\frac{\partial L}{\partial r}$ euqation gives us $\sum_n \alpha_n = \frac{1}{2}$. Also as we know $\lambda_n \geq 0$, we can simplify our constraint as $0 \leq \alpha_n \leq C$ using equation given by $\frac{\partial L}{\partial \xi_n}$. Now, to get the dual form, we put the conditions got from differentiations in Lagrangian function. So we get the dual form

$$\begin{aligned} L_D &= \sum_n \alpha_n \phi(\boldsymbol{x}_n)^T \phi(\boldsymbol{x}_n) - 2\sum_{m,n} \alpha_m \alpha_n \phi(\boldsymbol{x}_m)^T \phi(\boldsymbol{x}_n) \\ &= \sum_n \alpha_n k(\boldsymbol{x}_n, \boldsymbol{x}_n) - 2\sum_{m,n} \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n) \\ & \quad s.t. \sum_n \alpha_n = \frac{1}{2} \text{ and } 0 \leq \alpha_n \leq C \end{aligned}$$

4.2

From $\frac{\partial L}{\partial c}$ calculated in previous part,

$$\boldsymbol{c} = \frac{\sum_n \alpha_n \phi(\boldsymbol{x}_n)}{\sum_n \alpha_n}$$

To calculate r^2, we need to find some support vector $\phi(\boldsymbol{x}_n)$, whose $0 \leq \alpha_n \leq C$. So by using KKT conditions as we did for calculating b in lecture notes, we can write r^2 as

$$r^2 = ||\phi(\boldsymbol{x}_n) - \boldsymbol{c}||_2^2 = k(\boldsymbol{x_n}, \boldsymbol{x}_n) - 4\boldsymbol{\alpha}^T \boldsymbol{k_x} \boldsymbol{x}_n + 4\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}$$

4.3

Decision Rule for the above problem will be

$$y = sign(k(\boldsymbol{x}, \boldsymbol{x}) - 4\boldsymbol{\alpha}^T \boldsymbol{k_x} \boldsymbol{x} + 4\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha} - r^2)$$

where positive sign represents the data is abnormal and negative means normal.