1 a) let K be the corresponding kernel matrix. Then we create a n by 1 column vector of 1 and write it as 1_n then we have

$$v'Kv = v'\left(c1_n1_n'\right)v = c(v'1_n)(1_n'v) = c\left(\sum v_i\right)^2$$

When c <0 v'Kv <0 so k(,) is negative semidefinite. When c >0 v'Kv >0 so k(,) is positive semidefinite.

b) suppose $K^1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ $K^2 = \begin{pmatrix} 3 & -2 \\ -2 & 3 \end{pmatrix}$ so $k_l(x_i, x_j) = K_{i,j}^l$

$\mathrm{k} = \min\{k_1, k_2\}$ is $\begin{pmatrix} 1 & -2 \\ -2 & 1 \end{pmatrix}$ the determinant is -3 < 0 so k is not a positive semidefinite matrix.

c) define $(K_{XU})_{ij} = k(x_i, u_j)$ the kernel of k(,) on $\{u_1 \dots u_m\} \cup \{x_1 \dots x_n\}$ is given by

$$\mathrm{H} = \begin{pmatrix} K_{UU} & K_{XU}^T \\ K_{XU} & K \end{pmatrix}$$

K is psd and H is also psd. Thus there is a vector (v, x) with normal distribution N(0,H) the conditional distribution of v given x is Gaussian with mean zero and covariance matrix

$$K_{UU} - K_{XU}^T K^{-1} K_{XU}$$

K is invertible and as a covariance matrix, it is must be psd.

2 the lagragian form

$$L = R^2 - \sum_j (R^2 + \xi_j - ||\Phi(\mathbf{x}_j) - \mathbf{a}||^2)\beta_j - \sum \xi_j \mu_j + C \sum \xi_j \ ,$$

β_j *and* μ_j *are multipliers and* $C\sum \xi_j$ *is the penalty term.*

Taking derivative of with respect to R, a, ε_j we have

$$\sum_j \beta_j = 1$$

$$\mathbf{a} = \sum_j \beta_j \Phi(\mathbf{x}_j)$$

$$\beta_j = C - \mu_j \, .$$

$$\xi_j \mu_j = 0,$$

$$(R^2 + \xi_j - ||\Phi(\mathbf{x}_j) - \mathbf{a}||^2)\beta_j = 0.$$

The dual form can be written as

$$W = \sum_j \Phi(\mathbf{x}_j)^2 \beta_j - \sum_{i,j} \beta_i \beta_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j).$$

$$0 \leq \beta_j \leq C, \; j = 1, \ldots, N.$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-q||\mathbf{x}_i - \mathbf{x}_j||^2}$$

$$W = \sum_j K(\mathbf{x}_j, \mathbf{x}_j)\beta_j - \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j).$$

$$R^2(\mathbf{x}) = ||\Phi(\mathbf{x}) - \mathbf{a}||^2 \, .$$

$$R^2(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}) - 2\sum_j \beta_j K(\mathbf{x}_j, \mathbf{x}) + \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \, .$$

Radius of the sphere is

$$R = \{R(\mathbf{x}_i) \mid \mathbf{x}_i \text{ is a support vector } \} \, .$$

$$\{\mathbf{x} \mid R(\mathbf{x}) = R\} \, .$$

So points inside the cluster are normal, otherwise are abnormal.

3 a) the loss function can be written as

$$\mathbb{E}[\ell(-yf(x))|x] = P(y=+1|x)\ell(-f(x)) + P(y=-1|x)\ell(f(x)).$$

the derivative of the loss function equal to zero

$$\frac{\partial}{\partial f(x)}\mathbb{E}[\ell(-yf(x))|x] = -P(y=+1|x)\ell'(-f(x)) + P(y=-1|x)\ell'(f(x)),$$

where ℓ' is the derivative of ℓ. Therefore, we have

$$P(y=+1|x) = \frac{\ell'(f(x))}{\ell'(f(x)) + \ell'(-f(x))}$$

For y=-1 we can get the similar answer. And for both logistic and Exponetial loss function if we take derivative of the function we can easily get

$$P(y|x) = \frac{\exp(f(x))}{\exp(f(x)) + \exp(-f(x))}$$

Suppose we have a logistic model of probability and it can written as

$$P(Y=y \mid x) = \frac{1}{1+\exp(-y(w^T x + b))}.$$

with

Y to be equal $y \in \{-1, 1\}$

The likelihood function is

$$l(w,b) = \prod_{i=1}^{m} \frac{1}{1+e^{-y_i(w^T x_i + b)}}.$$

And the log-likelihood is

$$\max_{w,b} L(w,b) := -\sum_{i=1}^{m} \log(1+e^{-y_i(w^T x_i + b)})$$

If we replace the linear regression function as f(x) then the form is very like the log-loss function so we can say that the log-loss function can be considered as the negative log-likelihood function but the exponential loss function does not have such property.

b)

$$\mathcal{J}_{R_c(f)}(x_i) \quad = \quad -\frac{\nabla_{R_c(f)}(x_i)}{\nabla^2_{R_c(f)}(x_i)}.$$

We have the function

$$L[y, f(x)] \; = \log(1+e^{-2yf(x)})$$

Take the first and second derivative we have

$$\begin{aligned}
\nabla_{R_c(f^k)}(x_i) \quad &= \quad \frac{2y_i e^{-2y_i f^k(x_i)}}{1+e^{-2y_i f^k(x_i)}}, \\
\nabla^2_{R_c(f^k)}(x_i) \quad &= \quad \frac{4y_i^2 e^{-2y_i f^k(x_i)}}{\left(1+e^{-2y_i f^k(x_i)}\right)^2}
\end{aligned}$$

So

$$\mathcal{J}_{R_c(f^k)}(x_i) \quad = \quad \frac{1}{2} y_i (1+e^{-2y_i f^k(x_i)}).$$

At iteration k. and the solution becomes

$$\arg\min_{g\in\mathcal{G}} \sum_i w_i [\mathcal{J}_{R_c(f^k)}(x_i) - c_g g(x_i)]^2$$

So $z_n^{(m)}$ is

$$\frac{1}{2}y_i(1+e^{-2y_i f^k(x_i)})$$

$\omega_n^{(m)}$ is

$$w_i = 4\left(e^{f^k(x_i)} + e^{-f^k(x_i)}\right)^{-2}$$

4 a) the closed form is

$$\beta_\lambda = (\mathbf{X^T X} + \lambda \mathbf{I})^{-1}\mathbf{X^T y}$$

Y is the guassian distribution $\mathrm{N}(X\beta^*, \sum_\epsilon \quad)$

It also can be written as

$$\beta_\lambda \sim \mathcal{N}(\mathbf{M}_\lambda \mathbf{X}\beta^*, \mathbf{M}\Sigma_\epsilon \mathbf{M^T})$$
$$\text{where } \mathbf{M}_\lambda = (\mathbf{X^T X} + \lambda \mathbf{I})^{-1}\mathbf{X^T}$$

b) bias(λ)=E[$x^T\beta_\lambda$] -$x^T\beta^*$= $x^T E[\beta_\lambda]$ -$x^T\beta^*$=

$$x^T M_\lambda X\beta^* - x^T\beta^* = \left(x^T M_\lambda X - x^T\right)\beta^* = x^T\left(M_\lambda X - I\right)\beta^*$$

$$\mathbf{M}_\lambda = (\mathbf{X^T X} + \lambda \mathbf{I})^{-1}\mathbf{X^T}$$

c) Var(λ)=E[$(x^T\beta_\lambda - E(x^T\beta_\lambda))^2$]= E[$(x^T(\beta_\lambda - E\left(\beta_\lambda\right)))^2$]=

x^TE[$\beta_\lambda - E\left(\beta_\lambda\right)$]x = x^TVar(β_λ)x = $x^T(M_\lambda \sum_\epsilon M_\lambda^T)$x

$$\mathbf{M}_\lambda = (\mathbf{X^T X} + \lambda \mathbf{I})^{-1}\mathbf{X^T}$$

d) When λ is large, the bias term is large while variance is small this may cause underfitting and is too simple. When λ is small, variance term is large and bias is small which may cause overfitting and is likely too complex. The challenge with the above linear regression model lies in striking a balance between the two terms.

5.1 please refer to the code preprocess

5.2 for trainsvm.m and testsvm.m please refer to the code

5.3 a) for the 3-fold cross validation please refer to the code crossvalidation and here is the result

c	time	accuracy
0.0002	0.0988	0.5313
0.0010	0.0728	0.5313
0.0039	0.0832	0.5313
0.0156	0.0832	0.5313
0.0625	0.0780	0.5313
0.2500	0.0676	0.5313
1.0000	0.0936	0.5417
4.0000	0.0780	0.6042
16.0000	0.0936	0.6289

We can see that as c get higher the accuracy get higher.

b) So the c we should choose is 16

c) for the code please refer to the own_linear.m

The test accuracy is 0.6815

5.4 a)

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 54.1667%

Cross Validation Accuracy = 61.1979%

C	accuracy	endtime
0.0002	53.1250	0.0624
0.0010	53.1250	0.0936
0.0039	53.1250	0.0624
0.0156	53.1250	0.0624
0.0625	53.1250	0.0468
0.2500	53.1250	0.0624
1.0000	53.1250	0.0624
4.0000	54.1667	0.0624
16.0000	61.1979	0.0624

Accuracy = 60.5744% (232/383) (classification)

the best c is 16.0000

b) from the result we can see that result is the same but time is faster

my own code is about 0.0936 and this one is about 0.0624

5.5 a) the poly_libsvm result is

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 61.1979%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 61.7188%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 64.5833%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 66.4063%

Cross Validation Accuracy = 59.7656%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 68.8802%

Cross Validation Accuracy = 60.1563%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 72.3958%

Cross Validation Accuracy = 63.2813%

Cross Validation Accuracy = 53.125%

Accuracy = 73.6292% (282/383) (classification)(this is the test accuracy)

the best c is   16384.0000, the best d is 1

accuracy for test is 73.62924282

b) the result for RBF_libsvm is

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 54.4271%

Cross Validation Accuracy = 60.1563%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.9063%

Cross Validation Accuracy = 59.7656%

Cross Validation Accuracy = 65.1042%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 54.4271%

Cross Validation Accuracy = 61.1979%

Cross Validation Accuracy = 65.3646%

Cross Validation Accuracy = 69.0104%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 54.4271%

Cross Validation Accuracy = 61.7188%

Cross Validation Accuracy = 64.1927%

Cross Validation Accuracy = 70.1823%

Cross Validation Accuracy = 70.7031%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 54.4271%

Cross Validation Accuracy = 61.849%

Cross Validation Accuracy = 64.0625%

Cross Validation Accuracy = 69.6615%

Cross Validation Accuracy = 72.7865%

Cross Validation Accuracy = 72.526%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 54.4271%

Cross Validation Accuracy = 61.849%

Cross Validation Accuracy = 63.5417%

Cross Validation Accuracy = 67.5781%

Cross Validation Accuracy = 72.1354%

Cross Validation Accuracy = 74.8698%

Cross Validation Accuracy = 73.6979%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 54.4271%

Cross Validation Accuracy = 61.849%

Cross Validation Accuracy = 63.6719%

Cross Validation Accuracy = 67.3177%

Cross Validation Accuracy = 71.224%

Cross Validation Accuracy = 75.651%

Cross Validation Accuracy = 74.6094%

Cross Validation Accuracy = 74.2188%

Cross Validation Accuracy = 53.125%

Cross Validation Accuracy = 54.4271%

Cross Validation Accuracy = 61.849%

Cross Validation Accuracy = 63.2813%

Cross Validation Accuracy = 66.9271%

Cross Validation Accuracy = 69.401%

Cross Validation Accuracy = 73.5677%

Cross Validation Accuracy = 74.4792%

Cross Validation Accuracy = 74.6094%

Cross Validation Accuracy = 73.8281%

Cross Validation Accuracy = 54.4271%

Cross Validation Accuracy = 61.849%

Cross Validation Accuracy = 63.4115%

Cross Validation Accuracy = 66.0156%

Cross Validation Accuracy = 68.75%

Cross Validation Accuracy = 73.3073%

Cross Validation Accuracy = 73.9583%

Cross Validation Accuracy = 75.3906%

Cross Validation Accuracy = 75.1302%

Cross Validation Accuracy = 71.224%

Cross Validation Accuracy = 61.5885%

Cross Validation Accuracy = 63.4115%

Cross Validation Accuracy = 66.0156%

Cross Validation Accuracy = 68.2292%

Cross Validation Accuracy = 72.6563%

Cross Validation Accuracy = 74.349%

Cross Validation Accuracy = 74.6094%

Cross Validation Accuracy = 74.6094%

Cross Validation Accuracy = 74.6094%

Cross Validation Accuracy = 69.5313%

Accuracy = 75.718% (290/383) (classification)(this is the test accuracy)

the best c is 256.0000, the best d is 1

accuracy for test data is 75.71801567

from the result we can see that RBF is better since it has higher accuracy and the c I would choose is 256.