1 Problem1:

(a) Set $R(\beta) = \sum_i (x_i^T \cdot w - y_i)^2 + \lambda \|w\|_2^2$. Then, in order to find the optimal w^* we need to compute the derivative $\frac{dR}{dw}$ and solve $\frac{dR}{dw} := 0$. The derivative is,

$$\frac{dR}{dw} = 2\sum_i (x_i^T \cdot w - y_i)x_i + 2\lambda w.$$

Now, solve $\frac{dR}{dw} := 0$,

$$\sum_i x_i(x_i^T \cdot w - y_i) + \lambda w = 0 \Leftrightarrow \sum_i x_i x_i^T w + \lambda w = \sum_i x_i y_i$$

$$\Leftrightarrow (\sum_i x_i x_i^T + \lambda I)w = \sum_i x_i y_i \Leftrightarrow w^* = (\sum_i x_i x_i^T + \lambda I)^{-1} \sum_i x_i y_i.$$

(b) If $x \mapsto \Phi(x)$ then,

$$w^* = \Big(\sum_i \Phi(x_i)\Phi(x_i^T) + \lambda I\Big)^{-1} \sum_i \Phi(x_i)\Phi(y_i) = \Big(\sum_i \Phi(x_i)\Phi(x_i)^T + \lambda I\Big)^{-1} \sum_i \Phi(x_i)\Phi(y_i)$$

(c) From (b), w^* can be written as :
$w^* = (\Phi^T\Phi + \lambda I)^{-1}\Phi^T y$ where $\Phi = [\Phi(x_1)\Phi(x_2)\dots\Phi(x_N)] \in \mathbb{R}^N$, $y \in \mathbb{R}^N$.
In class we used a trick to change the dimensionality, hence it holds that :

$$(\Phi^T\Phi + \lambda I)^{-1}\Phi^T = \Phi^T(\Phi\Phi^T + \lambda I)^{-1}$$

Now, if we set $a = (\Phi\Phi^T + \lambda I)^{-1} y$ we get that $w^* = \Phi^T a$. Since, we know that $y_{est} = w^* x_{new}$, in that particular case we have that :

$$y_{new} = f(\Phi(x_{new})) = a^T\Phi \cdot \Phi(x_{new}) = \sum_i a_i \Phi(x_i)^T \Phi(x_{new}) =$$

$$= \sum_i a_i K(x_{new}, x_i).$$

(d)

- (1) From the plots we can notice that the plots from KRRS are exactly the same as BERR as we suspected. Moreover, when we employ the polynomial basis, there is a big difference between the plots of degree 2 and degree 6 , whereas, when we use the trig basis, the plots are pretty similar for the different orders. This is natural, since as M gets bigger up to a number (after this number we have overfitting which results in worse performance) the prediction gets better(As M increases the bias decreases).

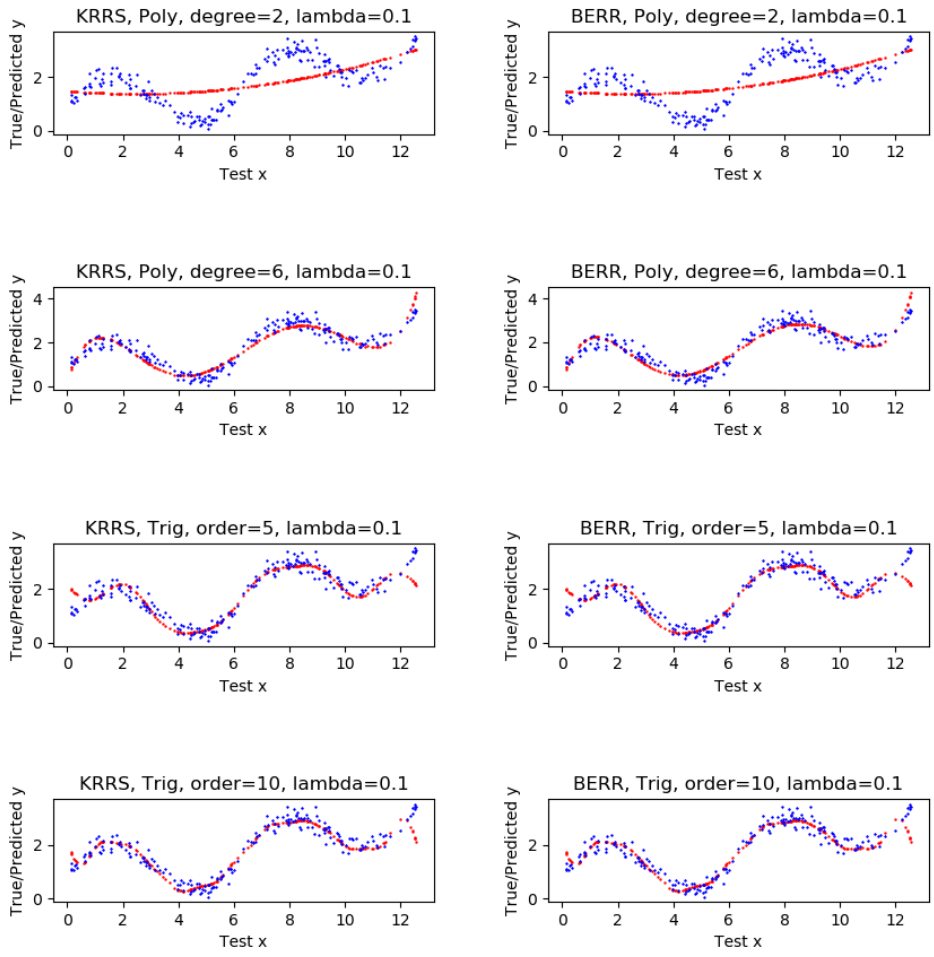

Figure 1: Plots represented as a 4x2 grid, using KRRS and BERR with polynomial and trigonometric bases for different polynomial degrees and trigonometric orders.

- (2)

	Kernel $k(x_1, x_2)$	Basis Expansion $\Phi(x)$
Polynomial degree 1	0.582340444	0.582340444
Polynomial degree 2	0.536822783	0.536822781
Polynomial degree 4	0.441507463	0.441507773
Polynomial degree 6	0.104134500	0.100380540
Trigometric degree 3	0.134081986	0.134081986
Trigometric degree 5	0.118868540	0.118868540
Trigometric degree 10	0.093916251	0.093916251

We notice that the MSEs for all different models are the same either using kernels or bases expansions. Moreover, notice that small changes(eg poly degree 6) occur when we use polynomial basis for KRRS and BERR but for the trigonometric basis the errors are exactly equal. That has to do with the stability(numerically) of trig basis over poly basis.

(e) The model with the lowest out of sample error is the RBF with parameters $\alpha = 0.01$ and $\gamma = 0.1$. The MSE of this model is 8.83646895 while the MSE we get from Kaggle is 2.54386. The big difference on the errors between the best out of sample error and the test error is due to the large fraction $\frac{504}{711}$ where 504 is the number of the test data and 711 the number of the training data. There are not enough training data,so the bias is high. For the cross validation I used KFold with $K = 5$. The parameter α represents the regularization parameter and so looking at the table, we notice that if α is large then we have high bias, thus large MSE. The parameter γ affects the rhythm with which the similarity functions moves to zero. In RBF, the larger γ the smaller the variance which means that the similarity functions are approaching one as two points are getting closer, hence RBF performs better among the other kernels.

	RBF	Poly.degree 3	Linear
$a = 1$ $\gamma = 1e-3$	20.08864157	9.39227097	10.07792204
$a = 1$ $\gamma = 1e-2$	14.88897073	10.62678927	—-
$a = 1$ $\gamma = 1e-1$	10.99198289	18.71113088	—-
$a = 1$ $\gamma = 1$	9.60601469	14.43433907	—-
$a = 1e-1$ $\gamma = 1e-3$	15.31637244	10.02448228	9.6208387
$a = 1e-1$ $\gamma = 1e-2$	11.37669386	9.18777203	—-
$a = 1e-1$ $\gamma = 1e-1$	9.0552491	10.7067749	—-
$a = 1e-1$ $\gamma = 1$	9.09508378	14.89181005	—-
$a = 1e-2$ $\gamma = 1e-3$	12.00657657	13.50232667	9.60953431
$a = 1e-2$ $\gamma = 1e-2$	9.31748887	9.13826818	—-
$a = 1e-2$ $\gamma = 1e-1$	8.83646895	9.29010488	—-
$a = 1e-2$ $\gamma = 1$	9.99960444	11.30875083	—-
$a = 1e-3$ $\gamma = 1e-3$	9.62584004	9.52375341	9.617979651
$a = 1e-3$ $\gamma = 1e-2$	9.01099229	9.024658	—-
$a = 1e-3$ $\gamma = 1e-1$	9.37277708	9.89847011	—-
$a = 1e-3$ $\gamma = 1$	11.379108230	19.77635181	—-
$a = 1e-5$ $\gamma = 1e-3$	10.8763483	10.38427886	9.6136654
$a = 1e-5$ $\gamma = 1e-2$	9.1807941	9.18997721	—-
$a = 1e-5$ $\gamma = 1e-1$	8.93701383	9.30879453	—-
$a = 1e-5$ $\gamma = 1$	10.41510284	15.0013623	—-

2 Problem2:

The model selection method I chose to use was KFold cross validation with $K = 5$. The best accuracy for this classification is obtained from the polynomial kernel with degree 3 for $C = 0.5$ and $\gamma = 1$. The accuracy is 0.99273247 and the kaggleized test results accuracy is 1. From the table (predictions from the train set), we notice that the data behave linearly since, no matter what the penalty parameter C is, the linear kernel performs relatively well. This observation also is noticeable when we compare the polynomial kernels with degrees 3,5 respectively. We see that the poly kernel with degree 3 outperforms the poly kernel with degree 5 which it tends to overfit(as the degree increases the variance increases). Moreover, the poly kernels work best when their coefficient γ is big and poorly when γ small due to underfitting. Finally, the RBF kernel works well when C is big and works poorly when it gets smaller. This is result of overfitting(high variance). The size of the data was relatively small, thus an indicator perhaps why the polynomial kernels outperform the RBF kernel.

	RBF	Poly.degree 3	Poly.degree 5	Linear
$C = 0.5$ $\gamma = 1$	0.98832117	0.99273247	0.9897916	0.98253465
$C = 0.5$ $\gamma = 0.01$	0.98691421	0.89359992	0.74773088	—
$C = 0.5$ $\gamma = 0.001$	0.97379668	0.55678621	0.55533693	—
$C = 0.05$ $\gamma = 1$	0.55533693	0.9839945	0.98106421	0.97671639
$C = 0.05$ $\gamma = 0.01$	0.97233682	0.71278959	0.66904686	—
$C = 0.05$ $\gamma = 0.001$	0.67923411	0.55533693	0.55533693	—
$C = 0.0005$ $\gamma = 1$	0.55533693	0.98836348	0.97524595	0.97087697
$C = 0.0005$ $\gamma = 0.01$	0.55533693	0.55678621	0.6194753	—
$C = 0.0005$ $\gamma = 0.001$	0.55533693	0.55533693	0.55533693	—