Max Reinhard - 359417
Ozge Sahin - 389691
Vincent Ulitzsch - 365672

ANSWERS TO QUESTIONS

Task1 :

In [34]:

```
from assignment_5 import *
%pylab inline
def train_krr(X_train, Y_train,kwidth,llambda):
    ''' Trains kernel ridge regression (krr)
    Input:       X_train  -  DxN array of N data points with D features
                 Y        -  D2xN array of length N with D2 multiple labels
                 kwdith   -  kernel width
                 llambda    -  regularization parameter
    Output:      alphas   -  NxD2 array, weighting of training data used fo
r apply_krr
    '''
    # your code here
    K = GaussianKernel(X_train,X_train,kwidth)
    alphas = sp.linalg.inv(K+llambda*sp.eye(X_train.shape[1])).dot(Y_train.T
)
    return alphas

def apply_krr(alphas, X_train, X_test, kwidth):
    ''' Applys kernel ridge regression (krr)
    Input:      alphas       -  NtrxD2 array trained in train_krr
                X_train      -  DxNtr array of Ntr train data points with D
eatures
                X_test       -  DxNte array of Nte test data points with D f
atures
                kwidht       -  Kernel width
    Output:     Y_test       -  D2xNte array
    '''
    # your code here
    K = GaussianKernel(X_test,X_train,kwidth)
    Y = (K.dot(alphas).T)
    return Y
```

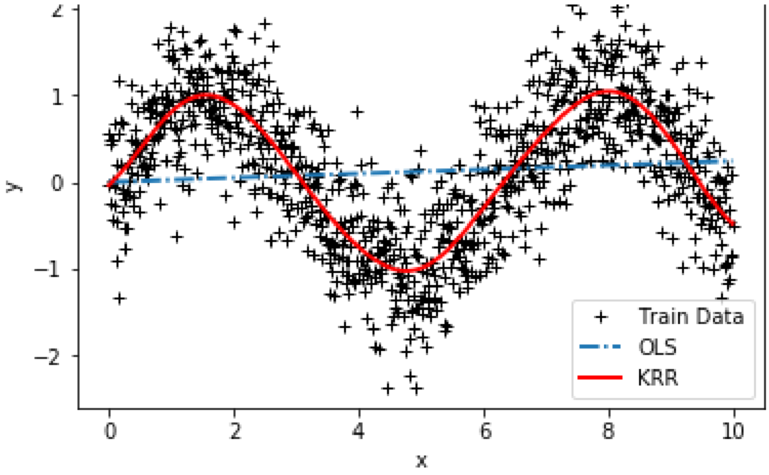
```
Populating the interactive namespace from numpy and matplotlib
```

Task2 :

a)When the the kernel width σ gets smaller the prediction fits better

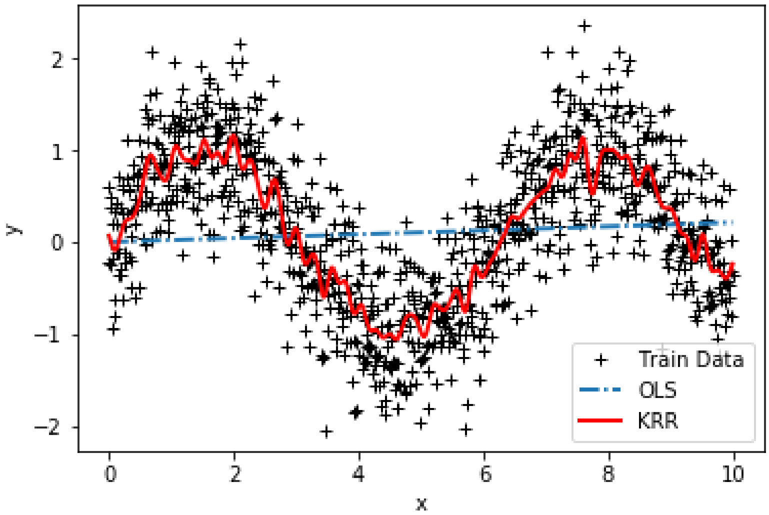
In [35]:

```
test_sine_toydata(kwidth = 1, llambda = 1)
```



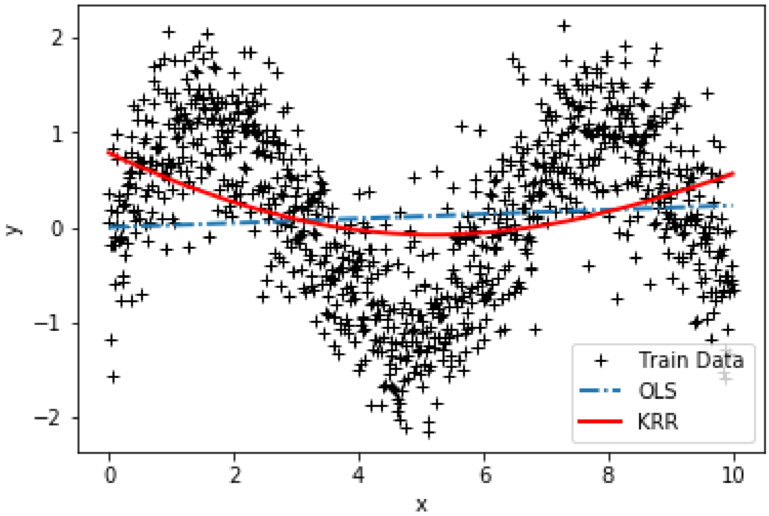
In [36]:

```
test_sine_toydata(kwidth = 0.1, llambda = 1)
```



In [37]:

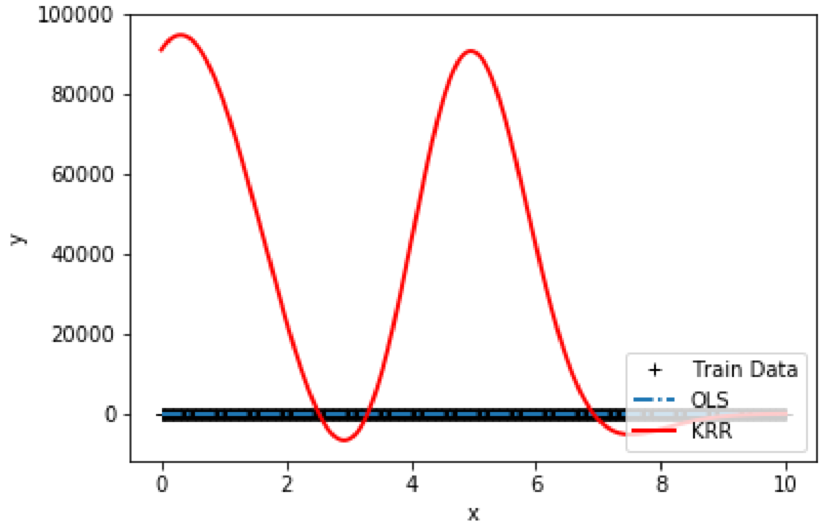
```
test_sine_toydata(kwidth = 10, llambda = 1)
```



b)When the regularization parameter gets smaller the prediction fits better.

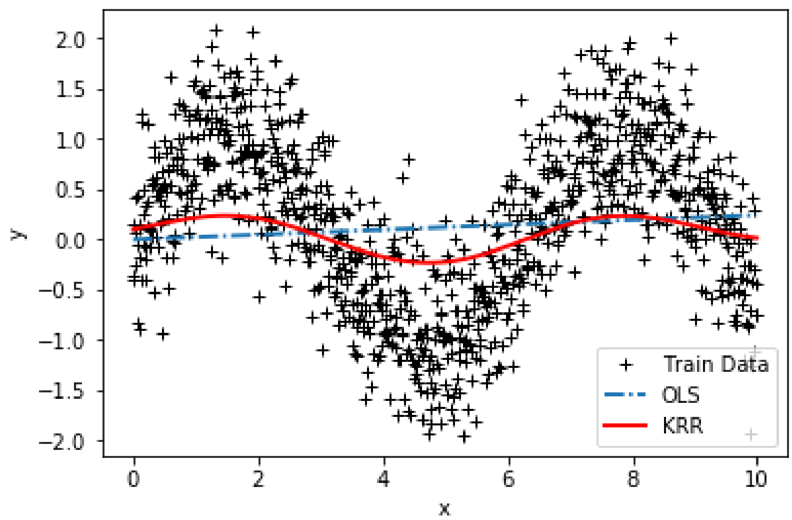
In [38]:

```
test_sine_toydata(kwidth = 1, llambda = 10 ** (-10))
```



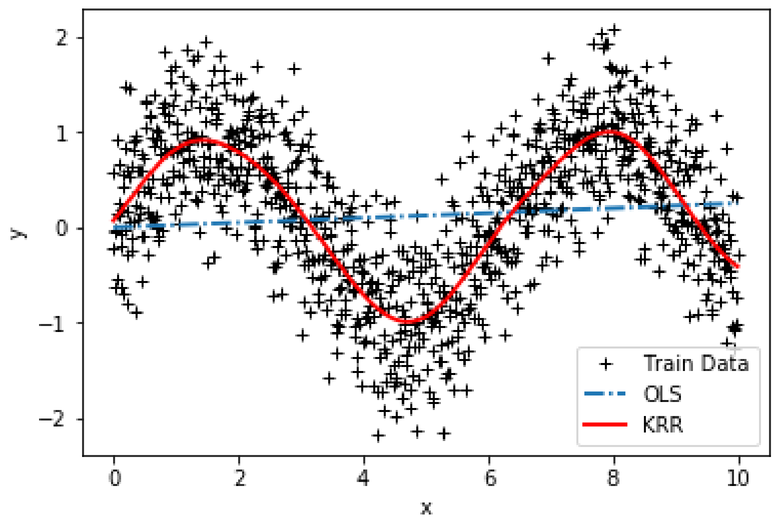
In [39]:

```
test_sine_toydata(kwidth = 1, llambda = 500)
```



In [40]:

```
test_sine_toydata(kwidth = 1, llambda = 1)
```



Task3 :

The function crossvalidate_krr uses the nested crossvalidation technique. The data is split into F disjunct folds. Then two crossvalidations are performed: An inner one and an outer one. In the outer one, we leave one fold out and pass the F-1 remaining folds to the inner one. The inner crossvalidation is a normal crossvalidation performed with the data of the F-1 remaining folds. The inner cross validation then outputs the best parameter for the F-1 remaining folds (the model selection part). The outer cross validation then evaluates the performance of the best parameter returned from the inner cross validation on the remaining fOuter fold.

This technique is used to counter the problem that in a normal cross validation, we use one test set to both select the parameters and evaluated the performance the chosen parameters give us, which might lead to a too optimistic evaluation of our performance.

Task 4 :

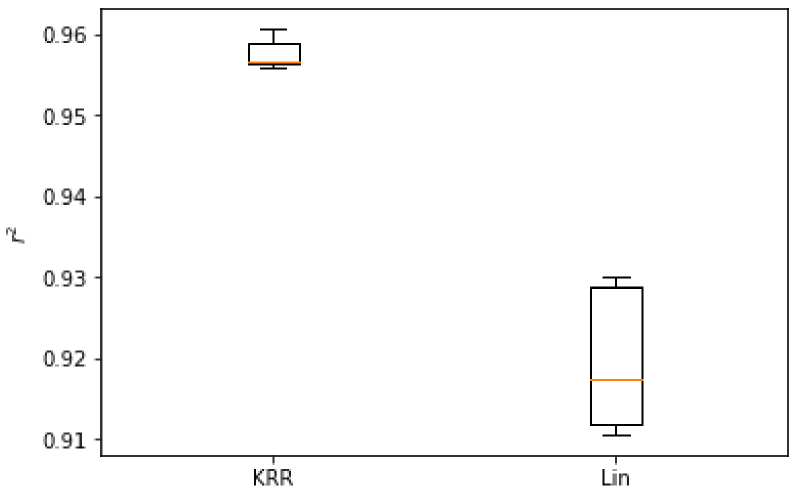
Box plot shows that distribution of R-square values across the 5 folds of cross validation for both Kernel ridge regression and Linear regression. Distributions are changing every time calling the test_handpositions function but median value of r-squares for kernel ridge is always bigger than simple linear regression.There could be some outliers for both two methods.Yes, we gain better fitted regression line from Kernel ridge regression as compared to Linear regression since
R- square values of Kernel ridge is much bigger than Linear regression .

In [41]:

```
test_handpositions()
```

```
/Users/ozgesahin/anaconda/lib/python3.6/site-
packages/numpy/core/fromnumeric.py:224: VisibleDeprecationWarning: using a
non-integer number instead of an integer will result in an error in the fut
ure
  return reshape(newshape, order=order)
```

```
Fold 0 best kernel width 10.000000 best regularizer 0.010000 rsquare 0.9606
64 rsquare linear 0.929993
Fold 1 best kernel width 10.000000 best regularizer 0.010000 rsquare 0.9559
61 rsquare linear 0.910612
Fold 2 best kernel width 10.000000 best regularizer 0.010000 rsquare 0.9589
77 rsquare linear 0.917495
Fold 3 best kernel width 10.000000 best regularizer 0.010000 rsquare 0.9563
18 rsquare linear 0.928879
Fold 4 best kernel width 10.000000 best regularizer 0.010000 rsquare 0.9566
20 rsquare linear 0.911763
```



Task5 :

It can be said that when considering computational complexity, performance of the crossvalidate_krr function used in test_handpositions function would be very bad if we use the whole dataset. Many nested for loops are used in the function.