Dennis Cornwell
CSE881 - HW2
9/18/13

1.a:

$$C = \frac{1}{N-1}(A - E[A])(A^T - E[A^T])$$
$$E[M] = \frac{1}{N}1_N M \text{ for any matrix } M$$
$$C = \frac{1}{N-1}(A - \frac{1}{N}1_N A)(A^T - \frac{1}{N}1_N A^T)$$
$$C = \frac{1}{N-1}A^T(I_N - \frac{1}{N}1_N)A$$

1.b:

$$C = \frac{1}{N-1}A^T(I_N - \frac{1}{N}1_N)A\ ,\ C = X\Lambda X^T\ ,\ A = U\Sigma V^T$$
$$A^T A = (U\Sigma V^T)^T(U\Sigma V^T)$$
$$A^T A = U^T\Sigma V\ (U\Sigma V^T) \text{ where } U^T U = I$$
$$A^T A = V\Sigma^2 V^T$$
$$\frac{1}{N-1}A^T(I_N - \frac{1}{N}1_N)A = X\Lambda X^T$$
$$A^T(I_N - \frac{1}{N}1_N)A = (N-1)X\Lambda X^T$$
$$A^T A - \frac{1}{N}A^T 1_N A = (N-1)X\Lambda X^T$$
$$V\Sigma^2 V^T - \frac{1}{N}A^T 1_N A = (N-1)X\Lambda X^T$$

1.c:

$$X = eig(\frac{(A^T - E[A^T])(A - E[A])}{n-1}) \text{ with } A \text{ centered } = eig(A^T A)$$
$$V \text{ is the set of eigenvectors for } A^T A = X$$

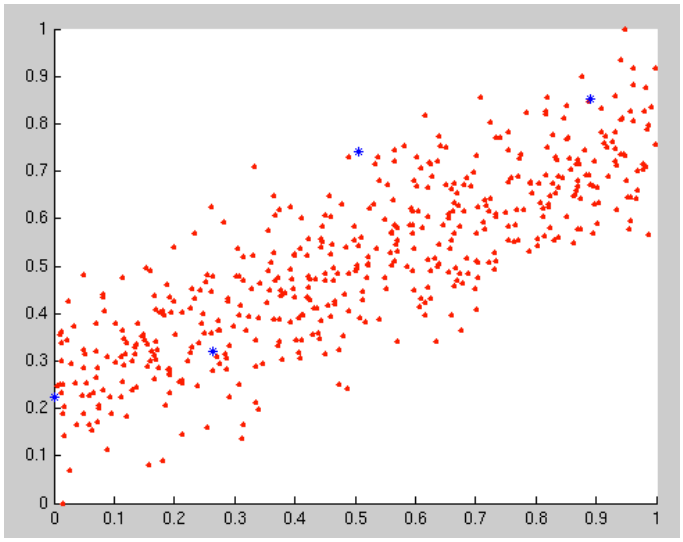
2.a:

```
>> problem2a.m
Loading data into matrix A...
New data projected onto 1st principal component
ans =
   -0.1147
   -0.3896
   -0.8128
   -1.2003

Variance of the projection of A onto 1st principal component
ans =
    0.1051
```

Dennis Cornwell
CSE881 - HW2
9/18/13

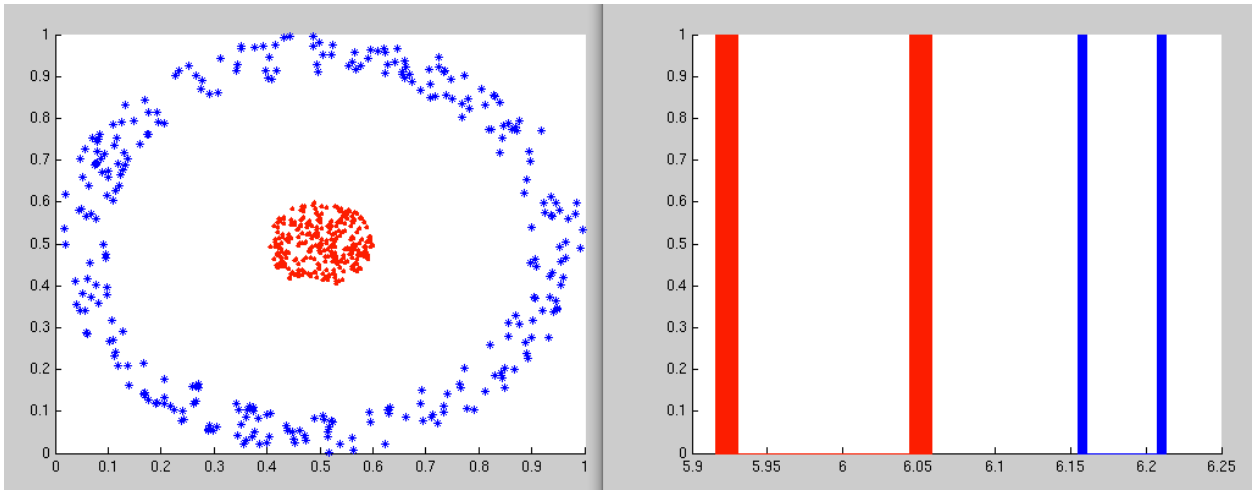
```
Variance of projection of A onto the X axis
ans =
    0.0800
```

Initial plot of the data:



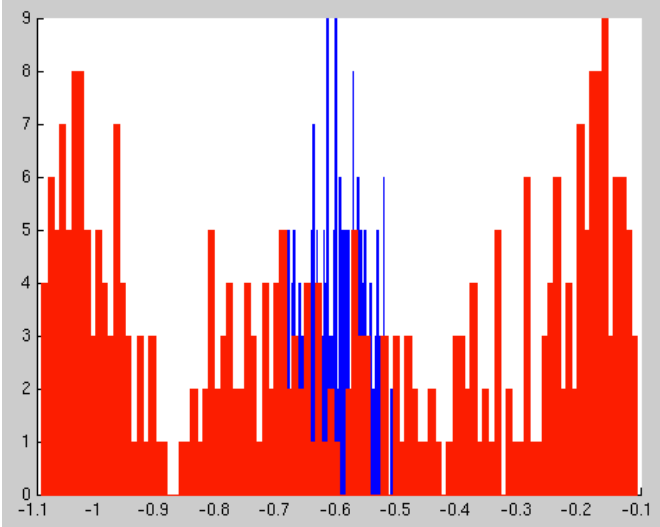
2.b: problem2b.m

Initial plot and kernel PCA histogram (note that red and blue data points have been swapped in the histogram)



Dennis Cornwell
CSE881 - HW2
9/18/13

Linear PCA Diaggram (again, red and blue have been swapped in this histogram)



Comparing the linear graph with the kernel method, it is clear that the kernel method does a much better job differentiating the features during pricipal component analysis.

3: The relatively low correlation here is likely because there are too many features and not enough training data to sufficiently form an accurate prediction model.

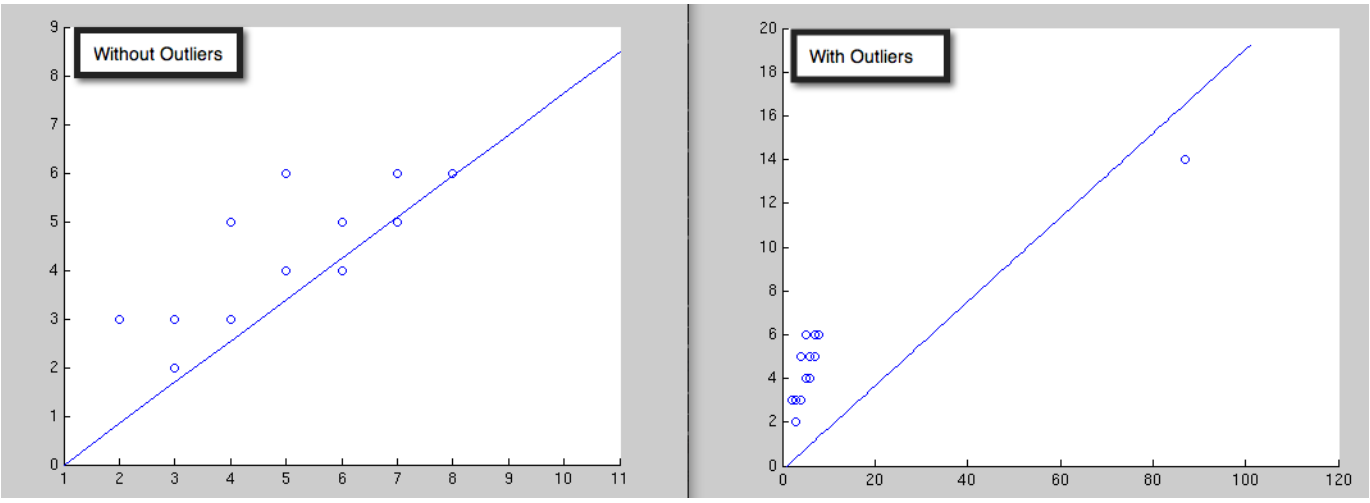
```
>> problem3.m
Correlation between projected data and actual results
ans =
    0.4143

First 10 projections of test data
ans =
  -33.6853
  -31.8821
  -13.3905
  -13.7858
    5.7995
  -26.3544
  -10.6662
   -7.9952
   -7.8465
  -32.5863
```

```
Actual test results
ans =
 -285.4427
 -285.4427
 -284.4427
 -285.4427
 -270.4427
 -286.4427
 -190.4427
 -283.4427
 -285.4427
 -285.4427
```

4.a:

When solving a simple curve fitting function such as $w = (X^T X)^{-1} X^T y$ outliers get factored in at the same weight as all other data points and as such the model will make an equal attempt to accomodate them as it will the normal data. For large datasets with few outliers, the effect is minimal, but the following simple example shows the effects when outliers are a larger percentage of the otherwise regular data set:

problem4a.m



4.b: Derivation of weighted ridge regression:

$$\sum_i \alpha_i (y_i - \sum_k X_{ik} w_k)^2 + \lambda \sum_j w_j^2$$

$$\frac{\delta}{\delta w_j} = -2 \sum_i \alpha_i (y_i - \sum_k X_{ik} w_k) X_{ij} + 2\lambda w_j = 0$$

$$\lambda w_j = \sum_i \alpha_i (y_i - \sum_k X_{ik} w_k) X_{ij}$$

$$\lambda w_j = \sum_i \alpha_i y_i X_{ij} - \sum_i \sum_k \alpha_i X_{ik} w_k X_{ij}$$

$$\lambda w_j = \sum_i X_{ij}^T \alpha_i y_i - \sum_i X_{ij}^T \sum_k \alpha_i X_{ik} w_k$$

$$Let\ A = diagonal\ matrix\ where\ A_{ii} = \alpha_i\ and\ A_{ij} = 0$$

$$\lambda w_j = (X^T A y)_j - \sum_i X_{ij}^T (XAw)$$

$$\lambda w_j = (X^T A y)_j - (X^T A X w_k)_j$$

$$[X^T A X w_k] + \lambda w = X^T A y$$

$$[X^T A X + \lambda I] w = X^T A y$$

$$w = [X^T A X + \lambda I]^{-1} X^T A y$$

4.c:

```
>> problem4c.m
Values of unweighted and weighted coefficients
w =
    2.0049
   -1.4474

wWeighted =
    2.5962
   -1.5148
```