1 Regression

1.1 Maximum Likelihood Solution

First we load the dataset, give it reasonable column names, and split it up into a training and a testing subset.

```
> setwd('~/github/StatML/Assignment2')
> load('./Data/bodyfat.RData')
> require(MASS)
> colnames(data)<-c('density','bodyFatPercentage','Age','Weight',
+                   'Height','Neck','Chest','Abdomen','Hip','Thigh',
+                   'Knee','Ankle','Biceps','Forearm','Wrist')
> ridx <- sample( 1:dim( data )[1], dim( data )[1] )
> trainIdx<-ridx[1:200]
> testIdx<-ridx[201:length( ridx )]
> train <- data[trainIdx,]
> test <- data[testIdx,]
```

We attempt to predict body fat using two models. The first one is based on weight, and chest, abdomen, and hip circumferences, and the second is based solely on abdominal circumference.

```
> selection1Train<-train[,c(4,7,8,9)]
> selection2Train<-train[,8]
> selection1Test<-test[,c(4,7,8,9)]
> selection2Test<-test[,8]
> targetTrain<-train[,2]
> targetTest<-test[,2]
```

We make a design matrix for every selection, and then find $\mathbf{w}_{\mathrm{ML}}$ by $\mathbf{w}_{\mathrm{ML}} = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}$ where $(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}$ is the pseudo-inverse of $\mathbf{\Phi}$.

```
> design1Train<-as.matrix(cbind(seq(1,1,length.out=nrow(selection1Train)),
+                           selection1Train))
> design2Train<-as.matrix(cbind(seq(1,1,length.out=length(selection2Train)),
+                           selection2Train))
> wML1<-ginv(design1Train)%*%targetTrain
> wML2<-ginv(design2Train)%*%targetTrain
```

Once e have found $\mathbf{w_{ML}}$ we can apply it to the test set, via $y(\mathbf{x}, \mathbf{w}) = \mathbf{w^T}\phi(\mathbf{x})$, and we can calculate the RMS via

$$\mathrm{RMS} = \sqrt{\frac{1}{N} = \sum_{n=1}^{N} (t_n - y(\mathbf{x_n}, \mathbf{w}))^{\mathbf{2}}}$$

```
> design1Test<-as.matrix(cbind(seq(1,1,length.out=nrow(selection1Test)),
+                           selection1Test))
> design2Test<-as.matrix(cbind(seq(1,1,length.out=length(selection2Test)),
+                           selection2Test))
```

```
> y1<-t(wML1)%*%t(design1Test)
> y2<-t(wML2)%*%t(design2Test)
> RMS1<-sqrt(sum((targetTest-y1)^2)/length(targetTest))
> RMS2<-sqrt(sum((targetTest-y2)^2)/length(targetTest))
> print(RMS1)

[1] 4.535804

> print(RMS2)

[1] 5.182109
```

Thus we get an RMS of 4.07 using the first model, and an RMS of 4.4 for the second model, showing the first model providing more accurate predictions.

For posterity, we will also calculate the error on the training set.

```
> y1Train<-t(wML1)%*%t(design1Train)
> y2Train<-t(wML2)%*%t(design2Train)
> RMS1Train<-sqrt(sum((targetTrain-y1Train)^2)/length(targetTrain))
> RMS2Train<-sqrt(sum((targetTrain-y2Train)^2)/length(targetTrain))
> print(RMS1Train)

[1] 4.42236

> print(RMS2Train)

[1] 4.788878
```

1.2 Maximum a posteriori Solution