Statistical Methods for Machine Learning
Assignment 2

Tudor Dragan (xlq880)
Nicolae Mariuta (rqt629)
Gabriel Carp (slp670)

March 10, 2015

II.1 Classification

II.1.1 Linear discriminant analysis

We have implemented the LDA algorithm according to the slides from the Linear Classification lecture. We also checked with the MATLAB *predict* function from the *ClassificationDiscriminant* class and we have the same results. As expected the train error is lower than the test error.

$$train_{ERR} = 0.1400$$

$$test_{ERR} = 0.2105$$

II.1.2 Linear discriminant analysis

We normalized both data sets and applied the LDA algorithm. We noticed that we get the same results as the non-normalized data. This would imply that the normalization has no effect on the accuracy of the LDA classifier. This happens because in

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - 1/2 {\mu_k}^T \Sigma^- 1 \mu_k + lnPr(Y = C_k)$$

the data points are multiplied by the covariance inverse and the mean is subtracted, thus doing a normalization inside the classifier. If we would normalize the data before applying the LDA classifier, the mean is 0 and the covariance is I which would imply that we obtain the same results as we did with the non normalized data set.

II.1.3 Bayes optimal classification and probabilistic classification

For the given problem, we have thy hypothesis class

$$H = h0(x) = 0, h1(x) = 1$$

because there it is only one element in the input space and possible values are *0* or *1.* The Bayes optimal classifier is the hypothesis for which we have the minimal risk: min(1 ? 1/4, 1 ? 3/4) = 1?4 which corresponds to the risk of hypothesis h1(x). The risk of the classifier is the sum of the risks for each classifier.

$$p(y = 0, h(x) = 1) = p(y = 0 \| h(x) = 1)p(h(x) = 1) = 0.25 * 0.75 = 0.1875$$

$$p(y = 1, h(x) = 0) = p(y = 1 \| h(x) = 0)p(h(x) = 0) = 0.75 * 0.25 = 0.1875$$

$$Rp(h) = 0.1875 + 0.1875 = 0.375$$

As a conclusion, the risk of this classifier is worse than using the Bayes optimal classifier.

II.2 Regression: Sunspot Prediction

II.2.1 Maximum likelihood solution

We used 3.15 and 3.16 to obtain the construct the design matrixes and train each of these models on the training set by finding the maximum likelihood estimate for the 3 selections. We have plotted for each selection the measured values from the test set and the predicted ones from our algorithm. As we can see from the 3 graphs the best results are obtained from considering all parameters.

The RMSs for each selection are:

$$RMS_1 = 35.4651$$

$$RMS_2 = 28.8398$$

$$RMS_3 = 18.7700$$

For the 2^{nd} selection we have plotted the predicted values shown in Figure 4. Because we have only one set of parameters we have only a 1^{st} degree equation which is shown as a straight line.

II.2.2 Maximum a posteriori solution

We have implemented the Maximum a posteriori algorithm using (3.52-3.54) equations. We used $\beta = 1$ and we chose α with values between 10^{-10} and 10^{10}. As we can see from Figure 5 with the values of RMS spanned across our alpha interval the best outcome is set by using Selection 3. We will analyze only the α values for selection 3 further. Between 10^{-10} and 10^{-2} we have the same RMS as the Maximum likelihood solution and for 10^{-1} the RMS is lower and the best estimation we get from α set to 10. Almost the same happens with the other selections, only on different α values.

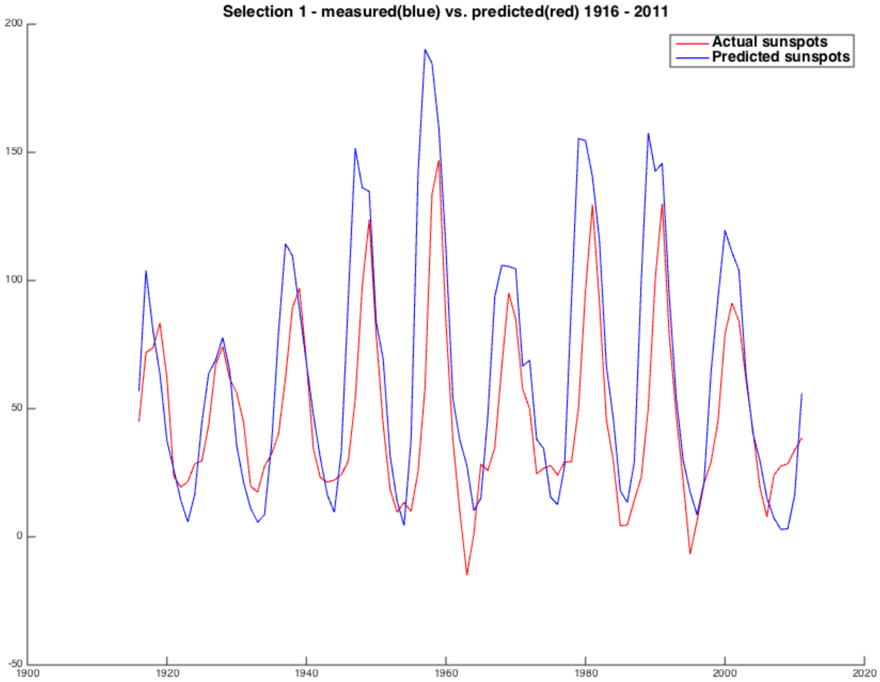

Figure 1: Selection 1 - measured(blue) vs. predicted(red) 1916 - 2011

II.2.3 Weighted sum-of-squares

According to Equation 3.11:

$$logp(t\|w,\beta) = \frac{N}{2}log\beta - \frac{N}{2}log2\pi - \beta E_D(w)$$

we determine the gradient for $E_D(w)$ and equalize it to 0

$$\nabla E_D(w) = \sum_{i=1}^{n} r_n\{t_n - w^T\phi(x_n)\}\phi(x_n)^T$$

$$0 = \sum_{i=1}^{n} r_n t_n \phi(x_n)^T - r_n w^T(\sum_{i=1}^{n} \phi(x_n)^T)$$

$$0 = \Phi^T Rt - w^T(\Phi^T R\Phi)$$

,where $R = diag(r_1, r_2, ...r_n)$. Solving it for w we get

$$w = (\Phi^T R\Phi)^{-1}\Phi^T Rt$$

Given the interpretation in terms of

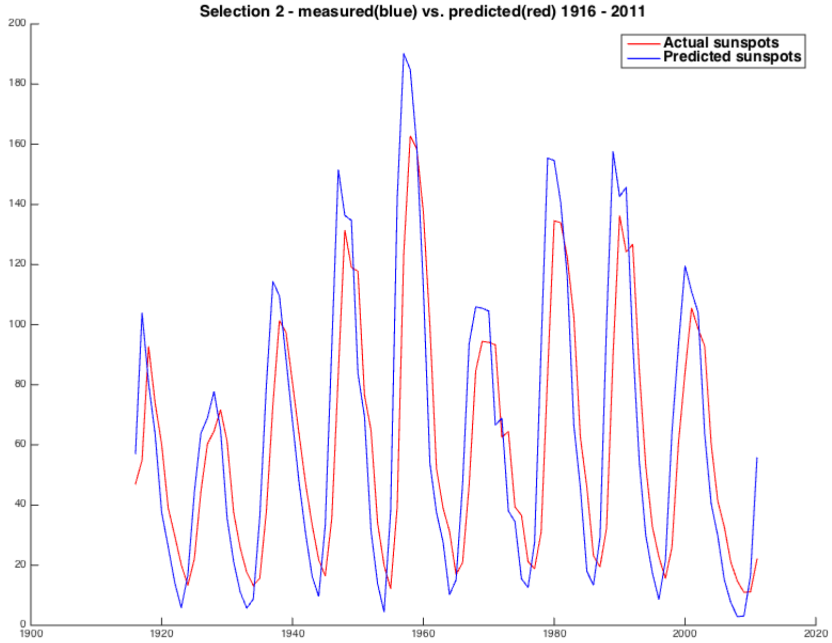

Figure 2: Selection 3 - measured(blue) vs. predicted(red) 1916 - 2011

(i) data dependent noise variance
We can see that r_n can be considered as an inverse variance vector specific to the data point (x_n, t_n) that can be interpreted as a scaling of β.

(ii) replicated data points
We can say that r_n can represent an effective number of replicated observations of this specific instance (x_n, t_n) particularly if we restrict r_n to be a positive integer.

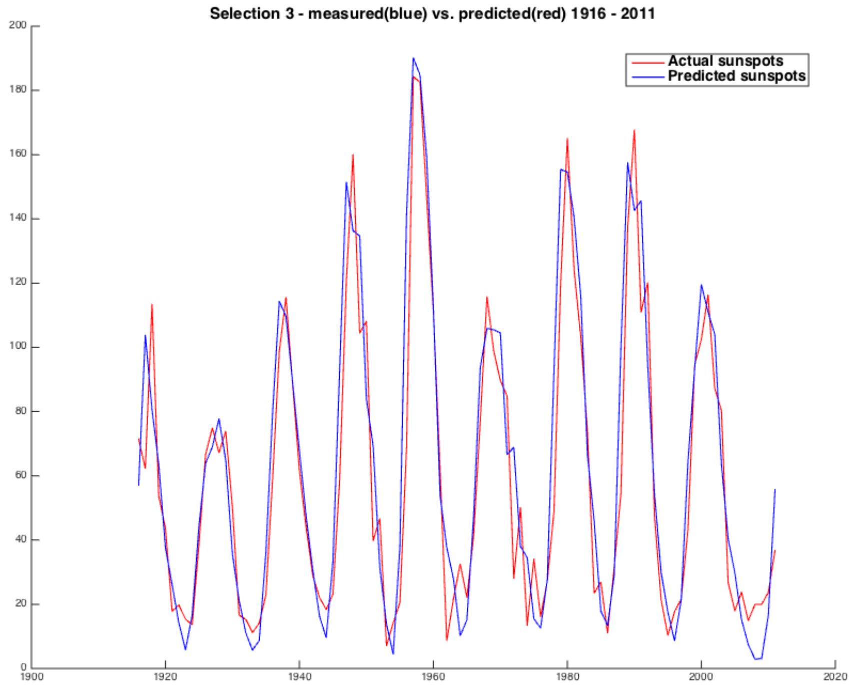

Figure 3: Selection 3 - measured(blue) vs. predicted(red) 1916 - 2011

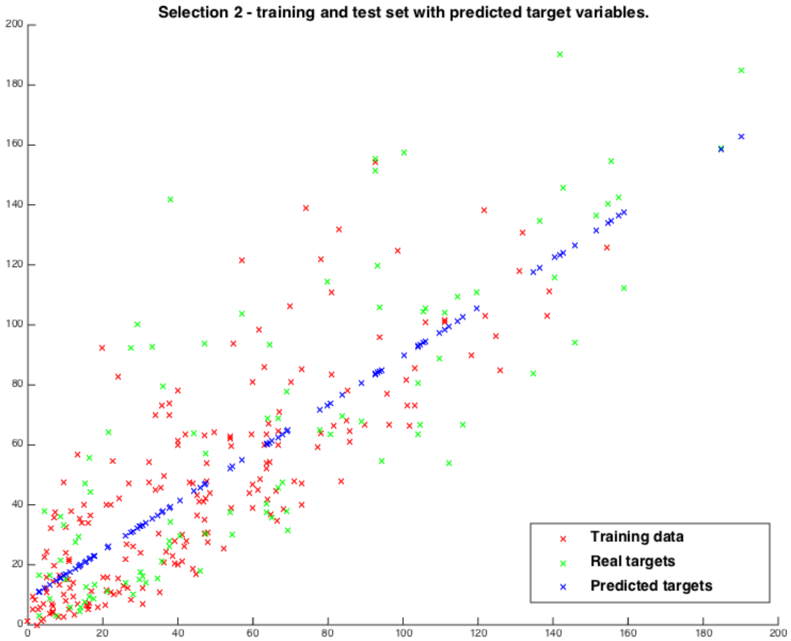

Figure 4: Predicted values for Selection 2

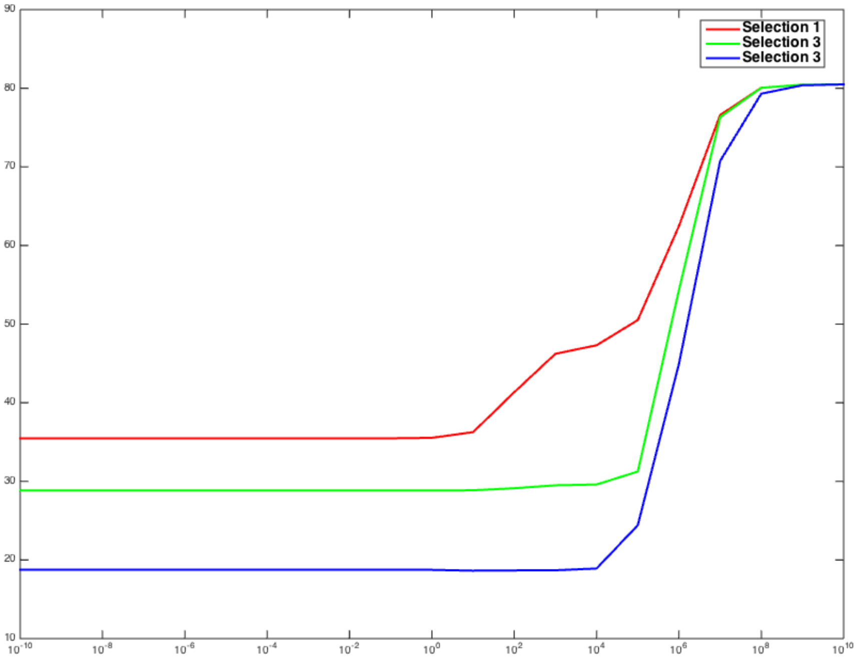

Figure 5: RMS error plotted over alpha values between 10^{-10} and 10^{10}