FCI Fall 2013 Assignment 1. 50 pts.
NAME(s): **Francisco Nardi** - 16163892 **and Vinicius Cidreira** - 16164648.

Electronic submission on Blackboard is due latest by 11 pm on Monday, Sep 9th. Submissions received after the deadline will be graded only for effort for a maximum of 70% of the total grade (Refer to class syllabus for detailed grading policy). **State any assumptions you make, justify your answers, show intermediate steps and explain your results for maximum credit**. You may leave quantitative answers in the form of expressions unless a numeric value is required to address the question. All answers should be in your own words with any sources you refer to cited at the appropriate places. Any knowledge you acquire from the Internet should be written in your own words and be appropriately referenced. Copying and pasting from the Internet, each other or any other source will not count as your effort (Refer to class syllabus for detailed policy on plagiarism).
You may submit this assignment in groups of two each. Write your names on this sheet and include it as the cover page for your submission. It will suffice to use a spreadsheet (e.g. Excel) for the calculations in this assignment.

Q1. (10) Use examples of small matrices to illustrate the identities in Table 1.4.
Q2. (5) Derive Equation 1.15 from Equation 1.14 by using the identities in Table 1.4.
Q3. (35) Use any Olympic event to create and test a linear model of Olympic winning times or distances as a function of calendar year. (see http://www.databaseolympics.com/sport/sporteventlist.htm?sp=ATH for data)
Use a subset of the data for n-way training/validation and another for testing. Report average loss on training, validation and test subsets.

Francisco Nardi 16163892
Vinicius Oliveira 16164648

Homework 1 - Comp. Intelligence

01.

	$f(w)$	$\delta f / \delta w$
a)	$w^T x$	x
b)	$x^T w$	x
c)	$w^T w$	$2w$
d)	$w^T C w$	$2Cw$

a) $f(w) = \begin{bmatrix} w_0 & w_1 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \end{bmatrix} = w_0 + w_1 x_1.$

$$\frac{\partial f(w)}{\partial w} = \begin{bmatrix} \frac{\partial f(w)}{\partial w_0} \\ \frac{\partial f(w)}{\partial w_1} \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \end{bmatrix} = x \Leftrightarrow (w^T x)' = x.$$

b) $f(w) = \begin{bmatrix} 1 & x_1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = w_0 + w_1 x_1.$

$$\frac{\partial f(w)}{\partial w} = \begin{bmatrix} \frac{\partial f(w)}{\partial w_0} \\ \frac{\partial f(w)}{\partial w_1} \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \end{bmatrix} = x \Leftrightarrow (x^T w)' = x.$$

c) $f(w) = \begin{bmatrix} w_0 & w_1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = w_0^2 + w_1^2.$

$$\frac{\partial f(w)}{\partial w} = \begin{bmatrix} \frac{\partial f(w)}{\partial w_0} \\ \frac{\partial f(w)}{\partial w_1} \end{bmatrix} = \begin{bmatrix} 2w_0 \\ 2w_1 \end{bmatrix} = 2 \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = 2w.$$

d) $$f(w) = \begin{bmatrix} w_0 & w_1 \end{bmatrix} \begin{bmatrix} a_{11}b_{11}+a_{12}b_{21} & a_{11}b_{12}+a_{12}b_{22} \\ a_{21}b_{11}+a_{22}b_{21} & a_{21}b_{12}+a_{22}b_{22} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} =$$

$$\begin{bmatrix} w_0 w_1 \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} w_0 w_1 \end{bmatrix} \begin{bmatrix} c_{11}w_0 + c_{12}w_1 \\ c_{21}w_0 + c_{22}w_1 \end{bmatrix} =$$

$$(c_{11}w_0 + c_{12}w_1)w_0 + (c_{21}w_0 + c_{22}w_1)w_1 .$$

$$\frac{\partial f(w)}{\partial w} = \begin{bmatrix} \dfrac{\partial f(w)}{\partial (w_0)} \\ \dfrac{\partial f(w)}{\partial (w_1)} \end{bmatrix} = \begin{bmatrix} (c_{11}w_0 + c_{12}w_1) + c_{11}w_0 + c_{21}w_1 \\ (c_{21}w_0 + c_{22}w_1) + c_{12}w_0 + c_{22}w_1 \end{bmatrix} =$$

(as the indices are inverted in $c_{21}w_1$, $c_{12}w_0 \to c_{12}w_1, c_{21}w_0$)

$$\begin{bmatrix} 2(c_{11}w_0 + c_{12}w_1) \\ 2(c_{21}w_0 + c_{22}w_1) \end{bmatrix} = 2\begin{bmatrix} c_{11}w_0 + c_{12}w_1 \\ c_{21}w_0 + c_{22}w_1 \end{bmatrix} =$$

$$2\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = 2Cw.$$

Q2. $$\frac{1}{N} w^T X^T X w - \frac{2}{N} w^T X^T t + \frac{1}{N} t^T t = \mathcal{L}.$$

$$\frac{\partial \mathcal{L}}{\partial w} = \left(\frac{1}{N} w^T \boxed{X^T X} w - \frac{2}{N} w^T \boxed{X^T t} \right)' \Rightarrow$$

$X^T X \to$ C | matrix$_{K,n}$ · matrix$_{n,K}$ = matrix$_{K,K}$

$X^T t \to$ matrix$_{n,n}$ · matrix$_{n,1}$ = matrix$_{n,1}$; which has the same shape of $\boxed{x}$.

Using the table 1.4, let's derivate the equation.

$$\frac{\partial L}{\partial w} = \left(\frac{1}{N} w^T C w - \frac{2}{N} w^T x \right)' \Rightarrow$$

$$\frac{\partial L}{\partial w} = \frac{2}{N} \underset{\boxed{X^T X}}{\underset{\downarrow}{\bigcirc\!\!\!\!\!C}}\, w - \frac{2}{N} \underset{\boxed{X^T t}}{\underset{\downarrow}{\bigcirc\!\!\!\!\!x}} \Rightarrow$$

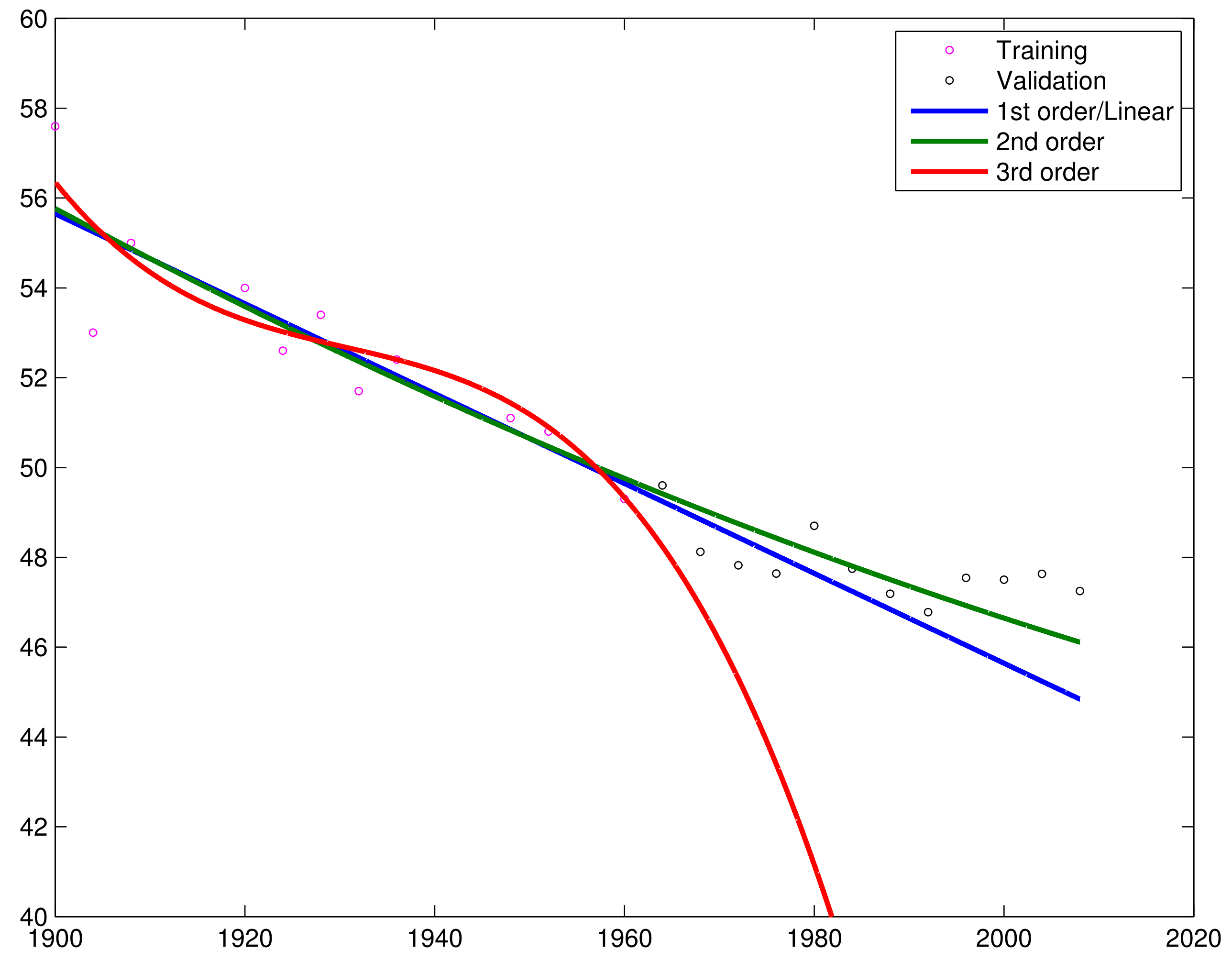
$$\frac{\partial L}{\partial w} = \cancel{\frac{2}{N}} X^T X w - \cancel{\frac{2}{N}} X^T t \Leftrightarrow$$

$$X^T X w = X^T t .$$

```
%1 In this algorithm, we will use the own dataset for both
%1 training and validation
%1 We are using the data obtained at
%1 http://www.databaseolympics.com/sport/sportevent.htm?sp=ATH&enum=200,
%1 which is related to the winning times in Olympics envolving
%1 the 400m Hurdles Men

%2 In order to build this algorithm, we studied another types of
%2 MatLab algorithms throught the internet, especially those ones
%2 referenced on the book, which can be accessed throught
%2 http://www.dcs.gla.ac.uk/~srogers/firstcourseml/matlab/chapter1/



%First, we need to clear the screen and erase the variables
%perhaps already stored in order to start running our program
clear all;
close all;


%The txt file 400mHurdlesMen will provide us with the dataset needed to
%test and validate our model
filename = '400mHurdlesMen.txt';
data = importdata (filename);

%The element 'a' will be a matrix of one column and 24 lines representing
%the years
%The element 'b' will be a matrix of one column and 24 lines representing
%the winning times
a = data (: , 1);
b = data (: , 2);

%The element 'valid' will be a matrix of one column and 24 lines
%which has all the indices of the matrix with the original dataset where
%the year is greater than 1960
% 12 ocurrences
valid = find (a > 1960);

%The element 'vala' will be a matrix of one column and 24 lines
%representing the years used in the validation set
%The element 'valb' will be a matrix of one column and 24 lines
%representing the winning times used in the validation set
% 12 ocurrences
vala = a (valid: end);
valb = b (valid: end);

%The values that will be used in the validation set
%through the element 'vala' will be removed from the element 'a',
%which will be used in the training set
%The values that will be used in the validation set
%through the element 'valb' will be removed from the element 'b',
%which will be used in the training set
a (valid:end) = [];
b (valid:end) = [];
```

```matlab
%'pwr' will be the vector which holds the powers of the polynomial
%functions that will be candidates to best model
pwr = [1 2 3];

%Commands to plot both markers
%magenta for training points
%and black for validation points
figure (1); hold off
plot (a, b, 'mo', 'markersize', 3);
hold all
plot(vala, valb, 'ko', 'markersize', 3);

%http://www.dcs.gla.ac.uk/~srogers/firstcourseml/matlab/chapter1/
%olympval.html
%Define the interval in which 'a' will be plotted and its accuracy
plota = [min(a):0.01:max(vala)]';

%This loop will execute three times, one for each power
for i = 1:length(pwr)
    %Initializing the matrices
    A = [];
    plotA = [];
    TrnT = [];
    valA = [];


    %Raises the value of each element to the power of k
    %valA will be used to find the validation loss, because it covers
    %the interval of vala
    %TrnT will be used to find the training loss, because it covers
    %the interval of a
    for k = 0:pwr(i)
        A = [A a.^k];
        valA = [valA vala.^k];
        TrnT = [TrnT a.^k];
        plotA = [plotA plota.^k];
    end

    %Find w according to its equation w = (Xtransp*X)^-1 * Xtransp * t
    w = inv(A'*A)*A'*b;
    plot(plota,plotA*w,'linewidth',2);

    %It finds the validation loss according to least squares
    val_loss(i) = mean((valA*w - valb).^2);

    %It find the training loss according to least squares
    trn_loss(i) = mean((TrnT*w - valb).^2);
end

%Limit the values in the axis y between 40 and 60
ylim([40 60]);
legend('Training','Validation','1st order/Linear','2nd order','3rd order');
```

```
for i = 1:length(pwr)
 fprintf('\n Model order: %g, Training loss: %g, Validation loss: %g',...
        pwr(i),trn_loss(i), val_loss(i));
end
```



60
58
56
54
52
50
48
46
44
42
40
1900
1920
1940
1960
1980
2000
2020
Training
Validation
1st order/Linear
2nd order
3rd order