2.1 Linear Regression

2.1.1 Standard linear regression

Considering the following set of point (with the format (input, output)): $D = \{(-2,2),(-1,3),(0,1),(2,-1)\}$.

1. Compute the parameters β using a linear regression. Don't forget the bias term.
2. Predict the output for point $x = 1$.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \qquad \hat{y}(1) = \beta_0 + \beta_1 x_1$$

$$\mathbf{X} = \begin{bmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 2 \end{bmatrix} \qquad \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 4 & -1 \\ -1 & 9 \end{bmatrix}$$

$$(\mathbf{X}^T\mathbf{X})^{-1} = \frac{1}{(4*9)-(-1*-1)} * \begin{bmatrix} 9 & 1 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} 0.2571 & 0.0286 \\ 0.0286 & 0.1143 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} 2 \\ 3 \\ 1 \\ -1 \end{bmatrix} \qquad \mathbf{X}^T\mathbf{Y} = \begin{bmatrix} 5 \\ -9 \end{bmatrix}$$

$$\beta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} = \begin{bmatrix} 1.0286 \\ -0.8857 \end{bmatrix}$$

2.1.3 Features and Learning

Consider the dataset of Fig. 3. Define the 2 features and perform a linear regression in that space of features.

Solution: The curve looks like a quadratic formula so we can define the following features:

$$\phi(x) = [1, x, x^2]$$

2.2 Ridge Regression

2.2.1 Ridge Regression

Consider the following set of points: $D = \{(-2,2),(-1,3),(0,1),(2,-1)\}$.

1. Compute the parameters of the regression β using a ridge regularization of $\lambda = 2$.

$$\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I} = \begin{bmatrix} 6 & -1 \\ -1 & 11 \end{bmatrix}$$

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1} = \frac{1}{(6*11)-(-1*-1)} * \begin{bmatrix} 11 & 1 \\ 1 & 6 \end{bmatrix} = \begin{bmatrix} 0.1692 & 0.0154 \\ 0.0154 & 0.0923 \end{bmatrix}$$

$$\mathbf{X}^T\mathbf{Y} = \begin{bmatrix} 5 \\ -9 \end{bmatrix}$$

$$\beta = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} = \begin{bmatrix} 0.7077 \\ -0.7538 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix}^T$$

3 Supervised learning - Classification

3.1 Perceptron

Consider the following training set with 5 data points:

Tabela 2: Training set.

Color (C)	Rigidity (R)	Smoothness (S)	Classification
1	1	1	-1
-1	1	-1	1
1	-1	1	1
1	1	-1	1
-1	-1	-1	1

Consider the test point.

$$\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

1. Doing one round of updates $\alpha = 1$, $\mathbf{w} = [1,1,1]$ e $w_0 = 1$, what is the resulting vector of w

$$\hat{y}(\mathbf{X}^{(1)}) = h([1,1,1,1]\cdot[1,1,1,1]) = h(4) = 1 \rightarrow \hat{y} \neq y \rightarrow UPDATE$$
$$\mathbf{w} = [1,1,1,1] + 1*(-1-1)[1,1,1,1] = [1,1,1,1] + [-2,-2,-2,-2] = [-1,-1,-1,-1]$$
$$\hat{y}(\mathbf{X}^{(2)}) = h([-1,-1,-1,-1]\cdot[-1,1,-1,1]) = h(0) = 1 \rightarrow \hat{y} = y$$
$$\hat{y}(\mathbf{X}^{(3)}) = h([-1,-1,-1,-1]\cdot[1,-1,1,1]) = h(-2) = -1 \rightarrow \hat{y} \neq y \rightarrow UPDATE$$
$$\mathbf{w} = [-1,-1,-1,-1] + 1*(1-(-1))[1,-1,1,1] = [-1,-1,-1,-1] + [2,-2,2,2] = [1,-3,1,1]$$
$$\hat{y}(\mathbf{X}^{(4)}) = h([1,-3,1,1]\cdot[1,1,-1,1]) = h(-2) = -1 \rightarrow \hat{y} \neq y \rightarrow UPDATE$$
$$\mathbf{w} = [1,-3,1,1] + 1*(1-(-1))[1,1,-1,1] = [1,-3,1,1] + [2,2,-2,2] = [3,-1,-1,3]$$
$$\hat{y}(\mathbf{X}^{(5)}) = h([3,-1,-1,3]\cdot[-1,-1,-1,1]) = h(2) = 1 \rightarrow \hat{y} = y$$

1 K Nearest Neighboors

1.1 Exercise

Consider the following set of 6 points:

F1 (C)	F2 (R)	F3 (S)	Classification
1	1	1	-1
1	1	1	-1
-1	1	-1	1
1	-1	1	1
1	1	-1	1
-1	-1	-1	1

And classify the following point using K-NN.

$$\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

Solution:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

First we need to compute the distances **D**. We will use the euclidean distance between points.

$$\mathbf{D} = \begin{bmatrix} 2.83 & 2.83 & 2.83 & 2 & 2 & 2 \end{bmatrix}$$

(1.) Para $\mathbf{T} = [1,-1,-1]^T$, 3 points have the smaller distance: $\mathbf{X}^{(4)} = [1-11]^T, \mathbf{X}^{(5)} = [11-1]^T, \mathbf{X}^{(6)} = [-1-1-1]^T$. $\mathbf{Y}^{(4)} = \mathbf{Y}^{(5)} = \mathbf{Y}^{(6)} = 1$. Anyone can be choosen, and as all have the same class there are not ambiguities. Classification 1.

(2.) Para $\mathbf{T} = [1,-1,-1]^T$, três pontos têm a menor distância: $\mathbf{X}^{(4)} = [1-11]^T, \mathbf{X}^{(5)} = [11-1]^T, \mathbf{X}^{(6)} = [-1-1-1]^T$. Todos têm a mesma classificação $\mathbf{Y}^{(4)} = \mathbf{Y}^{(5)} = \mathbf{Y}^{(6)} = 1$. Qualquer par pode ser escolhido como par de vizinhos mais próximos, mas todos têm a mesma classe. Logo não há ambiguidade. A classificação retornada é 1.

Exercises Neural Networks

$$f(x) = \sigma(x) = \frac{1}{1+e^{(-2\cdot x)}}$$

Given the weights $\mathbf{w}_1=\{w_{11}=1, w_{12}=1, w_{13}=0, w_{14}=0\}$, $\mathbf{W}_1=\{W_{11}=0, W_{21}=1\}$, *and the activation function* $\sigma(x)$.
Perform one step of the stochastic gradient descent with $\eta=2$ *with the input vector* $\mathbf{x}=\{4,4,1,0\}=\{x_1=4, x_2=4, x_3=1, x_4=0\}$ *eland the target* $\mathbf{t}=\{0,1\}$, *determine* ΔW_{ij} *e* Δw_{jk} *after one adaptation step.*
Please ignor Bias!

Hidden layer

net1=1*4+1*4+0*1+0*0=8
V1= $\sigma(net_1)$ = 1/(1+Exp(-(2*8))= 1/(1 + 1/e16)=1

Output layer

net1=0*1=0
net2=1*1=1
O1= $\sigma(net_1)$ = 1/(1+Exp(-(2*0)))= 0.5
O2= $\sigma(net_2)$ = 1/(1+Exp(-(2*1)))= 0.880797

Output Layer:

$$\Delta W_{ij} = (t_i - o_i) f'(net_i) V_j$$
$$\Delta W_{ij} = \delta_i V_j$$
$$f'(x) = 2\cdot\sigma(x)\cdot(1-\sigma(x))$$

$\Delta W_{11} = (0-0.5)*2$ *0.5*(1-0.5)*1=-0.25
$\Delta W_{21} = (1-0.880797)*2$ *0.880797*(1-0.880797)*1=0.02

δ1=-0.25, δ2=0.0250311

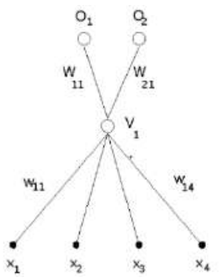
Hidden layer

$$\Delta w_{jk} = \sum_{i=1}^{2} \delta_i \cdot W_{ij} f'(net_j) \cdot x_k$$

$$\delta_j = f'(net_j) * \textstyle\sum_{i=1}^{2} \delta_i * W_{ij}$$

δ1=2*1*(1-1)*(0*(-0.25)+1*0.880797)=0

$$\Delta w_{jk} = \Delta w_{11} = \Delta w_{12} = \Delta w_{13} = \Delta w_{14} = 0$$



1.2 Exercise

Consider the following set of 4 points: $D = \{(0,1),(1,0),(2,1),(3,0)\}$, that have the following classifications $C = \{F,T,T,F\}$.

Solution: Cycle data points:

$$x = (0,1),\ y = F$$

Find closest centroid $d(x,c_1^F)^2 = .25$, $d(x,c_2^T)^2 = 1.25$, $d(x,c_3^F)^2 = 9.25$
Update closest centroid. $c_1 = c_1 + \eta(x - c_1) = (0,.5) + .1((0,1)-(0,.5)) = (0,0.55)$

$$x = (1,0),\ y = T$$

Find closest centroid $d(x,(0,.55))^2 \approx 1.2$, $d(x,(1,.5))^2 \approx 0.2$, $d(x,(3,.5))^2 \approx 4$
Update closest centroid. $c_2^T = c_2^T + \eta(x - c_2^T) = (1,.5) + .1((1,0)-(1,.5)) = (1,0.45)$

$$x = (2,1),\ y = T$$

Find closest centroid $d(x,(0,.55))^2 \approx 4.2$, $d(x,(1,.45))^2 \approx 1.3$, $d(x,(3,.5))^2 \approx 1.25$
As they have different classifications:
Update closest centroid. $c_3^T = c_3^T - \eta(x - c_3^T) = (3,.5) - .1((2,1)-(3,.5)) = (3.1,0.45)$

$$x = (3,0),\ y = F$$

Find closest centroid $d(x,(0,.55))^2 \approx 9.2$, $d(x,(1,.6))^2 \approx 4.4$, $d(x,(3,.5))^2 \approx 0.25$
Update closest centroid. $c_3^F = c_2^T + \eta(x - c_2^T) = (1,.55) + .1((2,1)-(1,.55)) = (3.11,0.395)$

1.4 Kernels

1.4.1 Exercise

Consider the following set of points with the format $(input, output)$: $D = \{(-2,2),(-1,3),(0,1),(2,-1)\}$.
Compute the prediction for $x = 1$ and $x = 0$, considering a gassian kernel of the form:

$$k(x,x') = \frac{1}{\sqrt{2\pi}} e^{-1/2||x-x'||^2}$$

Solution:
First step is to compute the distance matrix for all points:

$$D = \begin{bmatrix} 3 & 2 & 1 & 1 \\ 2 & 1 & 0 & 2 \end{bmatrix}$$

Taking into account the distances for each prediction points, we will compute the weights of the gassian kernel:

$$(-2,2) \rightarrow K(d(1,-2)) = K(3) = 0.00$$
$$(-1,3) \rightarrow K(d(1,-1)) = K(2) = 0.05$$
$$(0,1) \rightarrow K(d(1,0)) = K(1) = 0.24$$
$$(2,-1) \rightarrow K(d(1,2)) = K(1) = 0.24$$

The prediction is given as follows:

$$\hat{y}(1) = \frac{0.00*(2) + 0.05*(3) + 0.24*(1) + 0.24*(-1)}{0.00 + 0.05 + 0.24 + 0.24} = \frac{0.15}{0.53} = 0.2830$$

Similarly for the other prediction points:

Covariance matrix for a sample is:

$$c_{ij} = \frac{\sum_{k=1}^{n}\left(x_i^{(k)} - m_i\right)\left(x_j^{(k)} - m_j\right)}{n-1} \qquad C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

2 Clustering

2.1 Consider the following set of points:

$$D = \{(0,0),(1,0),(0,2),(2,2)\}$$

2.1.1 K-Means

For $K = 2$ perform a K-Means clustering. Use as initialization $u_1 = (2,0)$ and $u_2 = (2,1)$.

Solution:

First compute the distances of each point to each cluster centroid.

$$|x-u_1|^2 = |x-(2,0)|^2 = \{4,1,8,4\}$$
$$|x-u_2|^2 = |x-(2,1)|^2 = \{5,2,5,1\}$$

Choosing which cluster each element belongs:

$$(1,1,2,2)$$

Learning the new centroids.

$$u_1 = \frac{(0,0)+(1,0)}{2} = (0.5,0)$$
$$u_2 = \frac{(0,2)+(2,2)}{2} = (1,2)$$

Now we repeat the process with the new centroids

$$|x-u_1|^2 = |x-(0.5,0)|^2 = \{0.25,0.25,4.25,6.25\}$$
$$|x-u_2|^2 = |x-(1,2)|^2 = \{5,4,1,1\}$$

Choosing which cluster each element belongs:

$$(1,1,2,2)$$

As the elements belong to the same cluster no update is needed.

Ex1) PCA

Suppose we have following data points representing a sample of data from a population:

$$\vec{x}_i = \left\{\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}4\\0\end{pmatrix},\begin{pmatrix}2\\1\end{pmatrix},\begin{pmatrix}6\\3\end{pmatrix}\right\}$$

What is the K-L transformation?

Covariance matrix for a sample is:

$$c_{ij} = \frac{\sum_{k=1}^{n}\left(x_i^{(k)}-m_i\right)\left(x_j^{(k)}-m_j\right)}{n-1} \qquad C = \begin{bmatrix}c_{11} & c_{12}\\ c_{21} & c_{22}\end{bmatrix}$$

$$C = \begin{pmatrix}\frac{20}{3} & \frac{8}{3}\\ \frac{8}{3} & 2\end{pmatrix} = \begin{pmatrix}6.6666 & 2.6666\\ 2.6666 & 2\end{pmatrix}$$

(For population we divide by *n*)

m_1=(0+4+2+6)/4=3

m_2=(0+0+1+3)/4=1

c_{11}=((0-3)^2+(4-3)^2+(2-3)^2+(6-3)^2)/3=20/3=6.6666

c_{12}=c_{21}==((0-3)*(0-1)+(4-3)*(0-1)+(2-3)*(1-1)+(6-3)*(3-1))/3=8/3=2.6666

c_{22}=((0-1)^2+(0-1)^2+(1-1)^2+(3-1)^2)/3=6/3=2

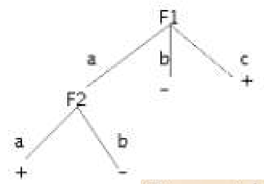
Ex 2 DECISION TREES

F1	*F2*	*F3*	*Output*
a	*a*	*a*	+
c	*b*	*c*	+
c	*a*	*c*	+
b	*a*	*a*	-
a	*b*	*c*	-
b	*b*	*c*	-

Determine the **whole decision tree** using the ID3 (information gain) and C4.5 (Gain Ratio) algorithm with the target "Output". Indicate all the computational steps!

$$E(P) = \sum_{i=1}^{n}\frac{|C_i|}{|C|}I(C_i)$$

$$gain(P) = I(C) - E(P)$$



Information Contend of the Table is:

p(+)=0.5
p(-)=0.5
I(table)= -3/6*log2(3/6) -3/6*log2(3/6) =1 bits

F1
Ca={+,-} Cb= {-,-} Cc= {+,+}

I(Ca)=-1/2*log2(1/2) -1/2*log2(1/2)= log2(2)=1 bit
I(Cb)=I(Cc)=0 bit

E(F1)=2/6*1+2/6*0+2/6*0=2/6

Gain(F1)=1-2/6= 4/6=0.666 bit

F2
Ca={+,+,-} Cb={+,-,- }

I(Ca)=-2/3*log2(2/3) -1/3*log2(1/3)= 0.9183 bit
I(Cb)=-1/3*log2(1/3) -2/3*log2(2/3)= 0.9183 bit

E(F2)=3/6*0.918+3/6*0.918=0.91830 bit
Gain(F2)=1-0.91830= 0.0817 bit

F3
Ca={+,-} Cc={+,+,-,-}
I(Ca)=-1/2*log2(1/2) -1/2*log2(1/2)= log2(2)=1 bit
I(Cb)=-2/4*log2(2/4) -2/4*log2(2/4)= log2(2)=1 bit

E(F3)=2/6*1+4/6*1=1

E(F3)=1-1
Gain(F3)=0

(Ca)=1, we can now chose either F2 or F3
Gain(F2)=1, Gain(F3)=1

Linear Unit

$$o_k = \sum_{j=0}^{D} w_j \cdot x_{k,j}$$

The update rule for gradient decent is given by

$$\Delta w_j = \eta \cdot \sum_{k=1}^{N}(t_k - o_k)\cdot x_{k,j}.$$

$$\Delta W_{ij} = \eta \sum_{d=1}^{m}\delta_i^d V_j^d$$

$$\Delta w_{jk} = \eta \sum_{d=1}^{m}\delta_j^d \cdot x_k^d$$

$$\delta_j^d = f'(net_j^d)\sum_{i=1}^{2}W_{ij}\delta_i^d$$

Sigmoid Unit

$$\sigma(net) = \frac{1}{1+e^{(-\alpha \cdot net)}} = \frac{e^{(\alpha \cdot net)}}{1+e^{(\alpha \cdot net)}}$$

$$o_k = \sigma\left(\sum_{j=0}^{N} w_j \cdot x_{k,j}\right)$$

$$\frac{\partial E}{\partial w_j} = -\alpha \cdot \sum_{k=1}^{N}(t_k - o_k)\cdot \sigma(net_{k,j})\cdot(1-\sigma(net_{k,j}))\cdot x_{k,j}.$$

$$\Delta w_j = \eta \cdot \alpha \cdot \sum_{k=1}^{N}(t_k - o_k)\cdot \sigma(net_{k,j})\cdot(1-\sigma(net_{k,j}))\cdot x_{k,j}$$

Logistic Regression

$$p(C_1|\mathbf{x}) = \sigma(net) = \frac{1}{1+e^{(-net)}} = \frac{e^{(net)}}{1+e^{(net)}}$$

$$p(C_1|\mathbf{x}) = \sigma\left(\sum_{j=0}^{N} w_j \cdot x_j\right) = \sigma\left(\mathbf{w}^T \cdot \mathbf{x}\right)$$

Error function is defined by negative logarithm of the likelihood which leads to the update rule where the target t_k can be only one or zero (a constraint)

The update rule for gradient decent is given for target $t_k \in \{0,1\}$

$$\Delta w_j = \eta \cdot \sum_{k=1}^{N}(t_k - o_k)\cdot x_{k,j}.$$

Stochastic Back-Propagation Algorithm

(mostly used)

1. Initialize the weights to small random values
2. Choose a pattern x^d_k and apply is to the input layer $V^0_k = x^d_k$ for all k
3. Propagate the signal through the network
   $$V_i^m = f(net_i^m) = f\left(\sum_j w_{ij}^m V_j^{m-1}\right)$$
4. Compute the deltas for the output layer
   $$\delta_i^M = f'(net_i^M)(t_i^d - V_i^M)$$
5. Compute the deltas for the preceding layer for $m=M,M-1,..2$
   $$\delta_i^{m-1} = f'(net_i^{m-1})\sum_j w_{ji}^m \delta_j^m$$
6. Update all connections
   $$\Delta w_{ij}^m = \eta \delta_i^m V_j^{m-1} \qquad w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij}$$
7. Goto 2 and repeat for the next pattern