FUNDAÇÃO GETULIO VARGAS

ESCOLA DE MATEMÁTICA APLICADA

MESTRADO 2015.3

APRENDIZAGEM POR MÁQUINAS

Prof Eduardo Mendes

Resolução do dever de casa #1

KIZZY TERRA

RIO DE JANEIRO

OUTUBRO/2015

1 Problem 1.3

a) Se w^* é um conjunto ótimo de pesos para os dados separáveis então temos que:

$$h(x_n) = sign({w^*}^T x_n) = y_n \; \forall n \in \{1, 2, .., N\} \quad (1)$$

Assim, podemos considerar dois casos:
1- $y_n > 0$:

$$y_n > 0 \Rightarrow sign({w^*}^T x_n) > 0 \Rightarrow y_n({w^*}^T x_n) > 0 \quad (2)$$

1- $y_n < 0$:

$$y_n < 0 \Rightarrow sign({w^*}^T x_n) < 0 \Rightarrow y_n({w^*}^T x_n) > 0 \quad (3)$$

Portanto, de (2) e (3) têm-se que $y_n({w^*}^T x_n) > 0 \; \forall n \in \{1, 2, .., N\}$ então dado $\rho = min_{1 \leq n \leq N} \, y_n({w^*}^T x_n)$ teremos que $\rho > 0$.

b) O resultado anterior nos diz que $\rho = min_{1 \leq n \leq N} \, y_n({w^*}^T x_n)$ é maior do que zero para todo n, de outra forma podemos escrever:

$$y_n({w^*}^T x_n) \geq \rho, \; \rho > 0; \; \forall n \in \{1, 2, .., N\} \quad (4)$$

Além disso, sabemos que em cada iteração o algoritmo PLA escolhe um par $(x_*, \, y_*)$ que foi incorretamente classificado e atualiza w segundo a seguinte equação:

$$w(t) = w(t-1) + x_* y_* \quad (5)$$

Decorre que:

$$w^{*T} w(t) = w^{*T} w(t-1) + y_*(w^{*T} x_*) \quad (6)$$

Utilizando (4) obtemos:

$$w^{*T} w(t) \geq w^{*T} w(t-1) + \rho \quad (7)$$

Queremos provar agora que$w^{*T} w(t) \geq t\rho$ para tanto escrevemos a seguinte prova por indução:
Temos que:

$$w^{*T} w(1) \geq w^{*T} w(0) + \rho \Rightarrow w^{*T} w(1) \geq \rho \quad (8)$$

Agora assuma que $w^{*T} w(t) \geq t\rho$ é verdade para $\forall t \in \{1, 2, .., n-1\}$. Iremos mostrar que também é verdade para $t = n$:

$$w^{*T} w(n) \geq w^{*T} w(n-1) + \rho \Rightarrow w^{*T} w(n) \geq (n-1)\rho + \rho = n\rho \quad (9)$$

Portanto:

$$w^{*T} w(t) \geq t\rho$$

c) Sabemos que:

$$w(t) = w(t-1) + x(t-1)y(t-1)$$

onde um par $(x(t-1), y(t-1))$ que foi incorretamente classificado. Assim, podemos escrever:

$$\| w(t) \|^2 = \| w(t-1) + x(t-1)y(t-1) \|^2$$

$$\| w(t) \|^2 = \| w(t-1) \|^2 + 2(w^T(t-1)x(t-1)).y(t-1) + \| x(t-1)y(t-1) \|^2$$

$$\| w(t) \|^2 \leq \| w(t-1) \|^2 + 2(w^T(t-1)x(t-1)).y(t-1) + \| x(t-1) \|^2 \| y(t-1) \|^2$$

Porém, $(w^T(t-1)x(t-1)).y(t-1) < 0$ dado que o par foi incorretamente classificado e $\| y(t-1) \|^2 = 1$, então:

$$\| w(t) \|^2 \leq \| w(t-1) \|^2 + \| x(t-1) \|^2$$

d) Definindo $R = max_{1 \leq n \leq N} \| x_n \|$, então $\| x_n \| \leq R$, utilizando este resultado iremos mostrar por indução que $\| w(t) \|^2 \leq tR^2$.

Temos que:

$$\| w(1) \|^2 \leq \| w(0) \|^2 + \| x(0) \|^2$$

Sabendo que $w(0) = 0$, obtemos:

$$\| w(1) \|^2 \leq \| x(0) \|^2 \leq R^2$$

Agora assuma que $\| w(t) \|^2 \leq tR^2$é verdade para $\forall t \in \{1, 2, .., n-1\}$. Iremos mostrar que também é verdade para $t = n$:

$$\| w(n) \|^2 \leq \| w(n-1) \|^2 + \| x(n-1) \|^2 \leq (n-1)R^2 + R^2 = nR^2$$

Portanto:

$$\| w(t) \|^2 \leq tR^2$$

e) A partir dos resultados obtidos nos itens b) e d) podemos concluir:

$$w(t)^T w^* \geq t\rho$$

$$\frac{w(t)^T}{\| w(t) \|} w^* \geq \frac{t\rho}{\| w(t) \|}$$

Entretanto, mostramos que $\| w(t) \|^2 \leq tR^2 \Rightarrow \| w(t) \| \leq \sqrt{t}R$, assim:

$$\frac{w(t)^T}{\| w(t) \|} w^* \geq \frac{t\rho}{\| w(t) \|} \geq \frac{t\rho}{\sqrt{t}R} = \sqrt{t}\frac{\rho}{R}$$

Além disso, se θ for definido como o ângulo entre $w(t)$ e w^* então seu cosseno pode ser escrito como:

$$cos\,\theta = \frac{w(t)^T w^*}{\| w(t) \| \| w^* \|}$$

Independente do valor de θ sabemos que seu cosseno deve ser menor ou igual a um. Portanto:

$$\frac{w(t)^T w^*}{\| w(t) \| \| w^* \|} \leq 1 \Rightarrow \frac{w(t)^T w^*}{\| w(t) \|} \leq \| w^* \|$$

Logo,

$$\| w^* \| \geq \frac{w(t)^T}{\| w(t) \|} w^* \geq \sqrt{t}\frac{\rho}{R}$$

$$\| w^* \|^2 \geq t\frac{\rho^2}{R^2}$$

Assim:

$$t \leq \frac{R^2 \| w^* \|^2}{\rho^2}$$

2 Problem 1.10

a) Queremos calcular o valor de $E_{off}(h, f)$ dado pela seguinte fórmula:

$$E_{off}(h, f) = \frac{1}{M} \sum_{m=1}^{M} [h(x_{N+m}) \neq f(x_{N+m})]$$

Para tanto iremos considerar os casos a seguir:

Caso 1: N é par e M é par

Nesse casso os números pares são : $\{N+2, N+4, ..., N+M\}$, resultando em $\frac{M}{2}$ números pares.

$$E_{off}(h, f) = \frac{1}{M} \sum_{m=1}^{M} [(N+m)\%2] = \frac{1}{M}\left(\frac{M}{2}\right) = \frac{1}{2}$$

Caso 2: N é par e M é ímpar

Nesse casso os números pares são : $\{N+2, N+4, ..., N+(M-1)\}$, resultando em $\frac{M-1}{2}$ números pares.

$$E_{off}(h, f) = \frac{1}{M} \sum_{m=1}^{M} [(N+m)\%2] = \frac{1}{M}(\frac{M-1}{2}) = \frac{M-1}{2M}$$

Caso 3: N é ímpar e M é par

Nesse casso os números pares são : $\{N+1, N+3, ..., N+(M-1)\}$, resultando em $\frac{M}{2}$ números pares.

$$E_{off}(h,f) = \frac{1}{M}\sum_{m=1}^{M}[(N+m)\%2] = \frac{1}{M}(\frac{M-1+1}{2}) = \frac{1}{2}$$

Caso 4: N é ímpar e M é ímpar

Nesse casso os números pares são : $\{N+1, N+3, ..., N+M\}$, resultando em $\frac{M+1}{2}$ números pares.

$$E_{off}(h,f) = \frac{1}{M}\sum_{m=1}^{M}[(N+m)\%2] = \frac{1}{M}(\frac{M+1}{2}) = \frac{M+1}{2M}$$

b) Todas as funções f que geram D em uma configuração sem ruído são tais que $f(x_n) = h(x_n)\ \forall n \in \{1, 2, ..., N\}$. Portanto, essas funções f podem variar os valores de $x_n\ \forall n \in \{N+1, ..., N+M\}$ os quais podem assumir os valores $+1$ e -1, resultando em 2^M funções f possíveis.

c) Para uma dada função hipótese h e um inteiro k tal que $0 < k < M$ queremos saber quantas das funções f satisfazem $E_{off}(h,f) = \frac{k}{M}$. O número k indica para quantos elementos do conjunto $X = x_1, x_2, ..., x_{N+M}$ são tais que $h(x_{N+m}) \neq f(x_{N+m})$.

Sabemos pelo item anterior que $f(x_n) = h(x_n)\ \forall n \in \{1, 2, ..., N\}$ logo para todas as funções f temos o número de pontos em que a função hipótese h difere de f será necessariamente maior ou igual a 0 e menor ou igual a M.

Entretanto, existe apenas uma função f tal que o este número de pontos é zero e outra função f única tal que o número de pontos é M. Portanto, o número de funções f que satisfazem $E_{off}(h,f) = \frac{k}{M},\ 0 < k < M$ é $2^M - 2$.

d) Se todas as funções f são igualmente prováveis para uma dada hipótese h então o valor esperado $E_f[E_{off}(h,f)]$ é dado por:

$$E_f[E_{off}(h,f)] = \sum E_{off}(h,f)P(E_{off}(h,f) = e_{off}(h,f)) = \sum_{k=0}^{M}\frac{k}{M}.C_m^k.\left(\frac{1}{2^M}\right)^k.\left(1-\frac{1}{2^M}\right)^{M-k}$$

e) Para quaisquer dois algoritmos determinísticos A_1 e A_2 o valor esperado para o erro fora do conjunto de treinamento é o mesmo, para um conjunto de funções f sem ruído, isto é:

$$E_f[E_{off}(A_1(\mathcal{D}), f)] = E_f[E_{off}(A_2(\mathcal{D}), f)]$$

Este resultado deve-se ao fato de que o valor esperado $E_f[E_{off}(h,f)]$ encontrado no item anterior, não depende da função hipótese h, e portanto também não depende de A_1 e A_2.

3 Problem 1.12

a) Para encontrar a hipótese h que minimizaa soma de desvios padrão dentro da amostra, iremos encontrar o mínimo para a função $E_{in}(h)$.

$$E_{in}(h) = \sum_{n=1}^{N}(h - y_n)^2$$

Primeiramente, encontramos o ponto crítico de $E_{in}(h)$:

$$\frac{dE_{in}(h)}{dh} = 2\sum_{n=1}^{N}(h - y_n) = 0$$

$$h_{mean} = \frac{1}{N}\sum_{n=1}^{N} y_n$$

Para verificar se h_{mean} é um ponto de mínimo ou máximo, calculamos o valor da segunda derivada:

$$\frac{d^2E_{in}(h)}{dh^2} = 2N > 0$$

Logo, h_{mean} é um ponto de mínimo.

b) Para encontrar a hipótese h que minimizaa soma de desvios padrão dentro da amostra, iremos encontrar o mínimo para a função $E_{in}(h)$.

$$E_{in}(h) = \sum_{n=1}^{N} \mid h - y_n \mid$$

O valor da derivada de $E_{in}(h)$ será dado pelo valor da soma das derivadas de cada diferença $\mid h - y_n \mid$. Essas derivadas podem ser $+1$ou -1 dependendo do sinal de $(h - y_n)$. Portanto, para que a derivada de $E_{in}(h)$ se anule é necessário que se tenha um mesmo número de derivadas de $\mid h - y_n \mid$ com valor $+1$ e -1. Isto siginifica que metade dos pontos devem ser maiores do que h e os pontos da outra metade devem ser menores do que h. Portanto, o estimador que minimiza $E_{in}(h)$ é a mediana, representado por h_{med}.

c) Os estimadores para h encontrados nos itens anteriores são dados por:

$$h_{mean} = \frac{1}{N}\sum_{n=1}^{N} y_n$$

e h_{med} que é a mediana do conjunto.

Se y_N é perturbado, o valor da mediana não se altera, pois continua sendo o valor que é maior do que metade dos pontos e menor do que a outra metade dos pontos. Já o valor de h_{mean} depende de y_N e portanto irá aumentar caso y_N seja perturbado de $\varepsilon \to \infty$. De outra forma:

$$h_{mean} = \frac{1}{N}\sum_{n=1}^{N-1} y_n + \frac{1}{N}y_N \Rightarrow h_{mean} \to \infty \, se \, y_N \to \infty$$

4 PLA Problem

Para resolver este exercício foi necessário implementar o Perceptron, bem como métodos para gerar um conjunto de dados aleatório, gerar a função objetivo, calcular erro e média de iterações e plotar os resultados encontrados. O código-fonte pode ser acessado em: https://github.com/kizzyterra14/AM-2015-3/blob/master/Homeworks/hw1.ipynb.

a) Em média o algoritmo Perceptron implementado levou 6,995 iterações para convergir com $N = 10$.

b) A probabilidade encontrada para que f e g concordem na classificação de um ponto gerado aleatoriamente para $N = 10$ foi de $p = 0,9191$.

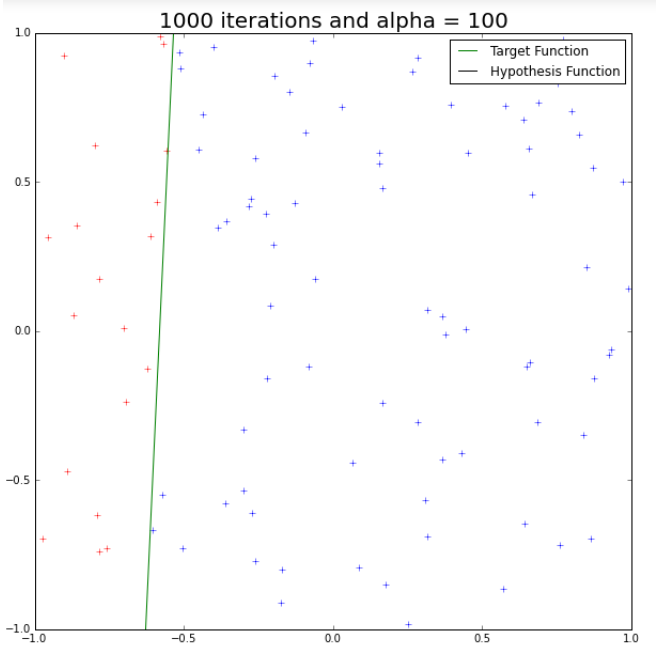
c) Em média o algoritmo Perceptron implementado levou 72,801 iterações para convergir com $N = 100$.

d) A probabilidade encontrada para que f e g concordem na classificação de um ponto gerado aleatoriamente para $N = 100$ foi de $p = 0,98952$.

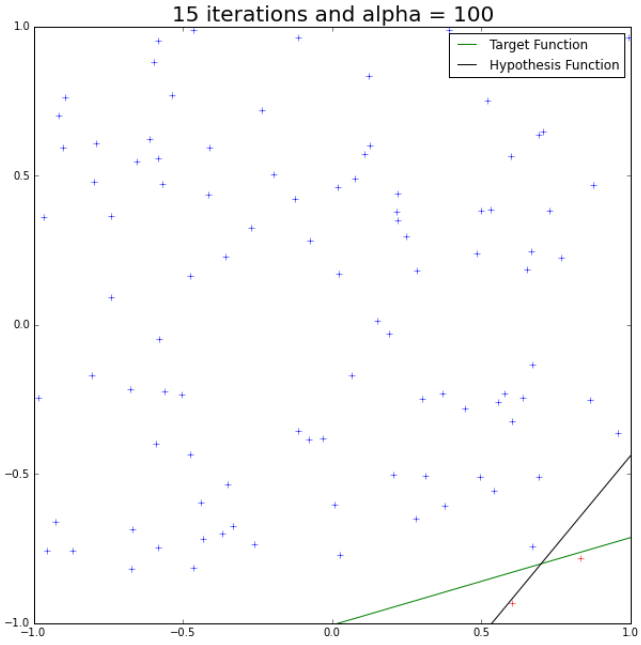
5 Problem 1.5

Para resolver este exercício utilouz-se as mesmas implementações feitas para o exercício anterior, mas foi implementado um algoritmo Perceptron modificado. O código-fonte pode ser acessado em: https://github.com/kizzyterra14/AM-2015-3/blob/master/Homeworks/hw1.ipynb.

a) Para $\alpha = 100$ obteve-se overflow para os valores de w estimados com o algoritmo Perceptron. Foram realizadas várias iterações para vários conjuntos de dados gerados aleatoriamente e em 99% das iterações os valores de $w_0,\ w_1,\ w_2$ "explodiram" e portanto a solução não converiu para menos de 1000 iterações do algoritmo adaptativo. Testou-se também para $\alpha = 10$ e encontrou-se o mesmo problema de overflow e não convergência em menos de 1000 iterações. Como é possível ver nos gráficos a seguir nenhum função hipótese pode ser calculada. Foi possível observar, entretanto, que para alguns dataset gerados o algoritmo eventualmente convergia, isso porém ocorreu em 1% dos casos, como no exemplo mostrado na figura 1(b).



(a) Em 99% dos casos não convergiu



(b) Em 1% dos casos convergiu

Figura 1: $\alpha = 100$

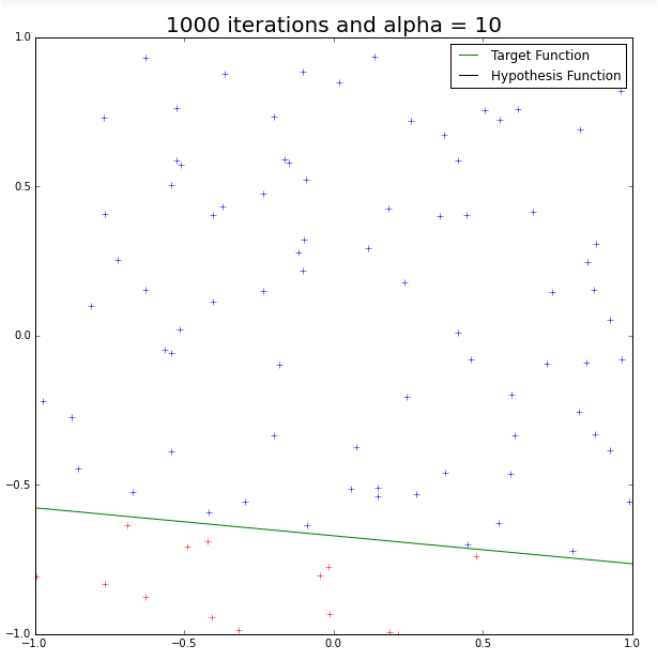


Figura 2: $\alpha = 10$

b) Para $\alpha = 1$ obteve-se o resultado a seguir:

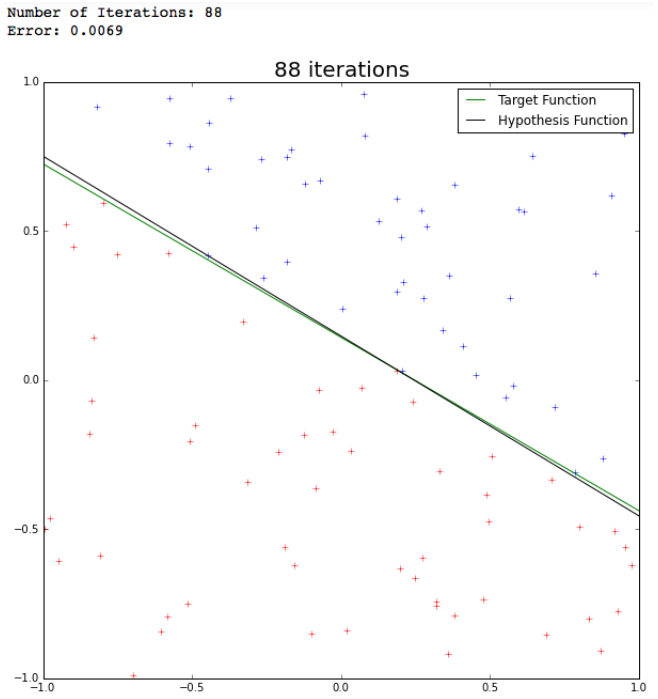


Figura 3: $\alpha = 1$

c) Para $\alpha = 0.01$ obteve-se o resultado a seguir:

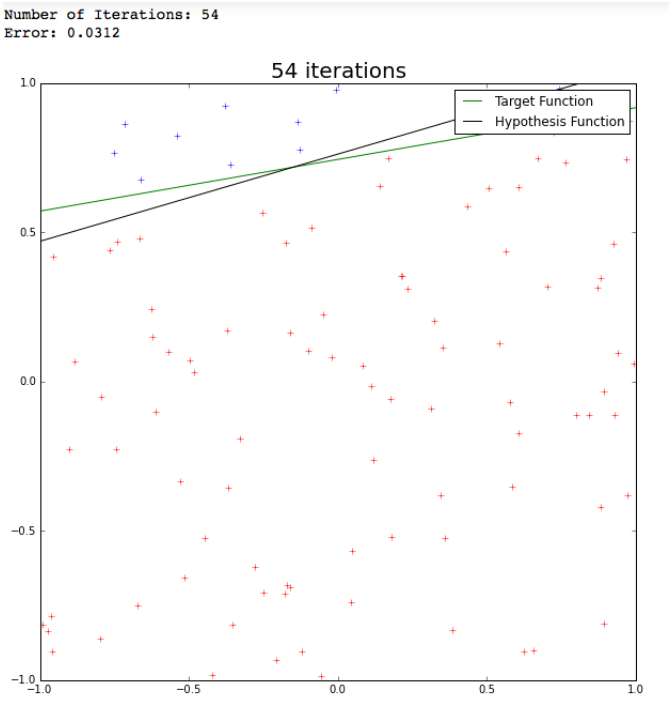


Figura 4: $\alpha = 0.01$

d) Para $\alpha = 0.0001$ obteve-se o resultado a seguir:

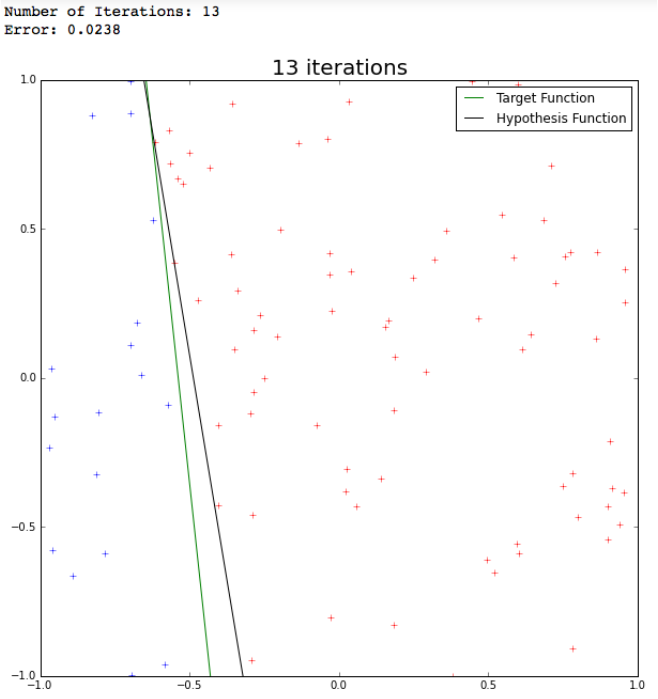


Figura 5: $\alpha = 0.0001$

Foi possível observar que conforme diminui-se o valor de α o algoritmo tende a convergir em um número menor de iterações.