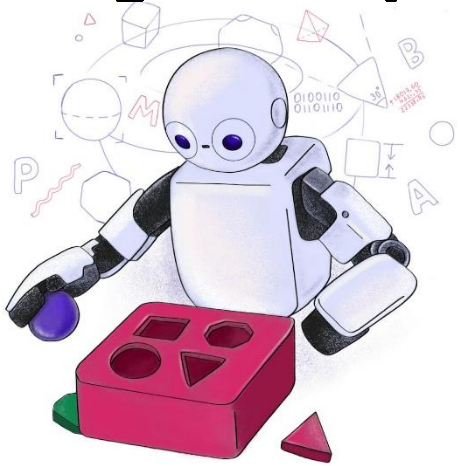
T320 - Introdução ao Aprendizado de Máquina II: *Classificação (Parte III)*



Inatel

Felipe Augusto Pereira de Figueiredo
felipe.figueiredo@inatel.br

Recapitulando

- Anteriormente, aprendemos que a ***classificação*** pode ser feita usando-se uma ***função discriminante***, que nada mais é do que um ***polinômio***, que tem sua saída passada através de outra função chamada de ***função de limiar***.
- Como na ***regressão linear***, o problema da classificação está em encontrar os pesos da ***função discriminante*** de tal forma que as classes sejam separadas da melhor forma possível.
- Vimos que a função de limiar mais simples é a de ***limiar rígido***, porém, ela apresenta alguns problemas como não poder ser utilizada para encontrar uma ***solução em forma fechada*** ou com o ***gradiente descendente*** e não nos dar a ***confiança de um resultado*** de classificação.
- Aprendemos também, uma forma intuitiva e iterativa de encontrar os pesos da ***função discriminante*** quando usamos o ***limiar rígido***.
- Na sequência, introduziremos outra função de limiar, chamada de ***função logística***, com a qual é possível se encontrar uma solução eficiente com o ***gradiente descendente*** e termos o ***grau de confiança*** de uma classificação.

Classificação com função de limiar logístico

- Como discutimos anteriormente, a ***função hipótese***, $h_{\boldsymbol{a}}(\boldsymbol{x}) = f(g(\boldsymbol{x}))$, com ***limiar de decisão rígido*** é descontínua em $g(\boldsymbol{x}) = 0$ e tem derivada igual a zero para todos os outros valores de $g(\boldsymbol{x})$.
- Além disso, o ***classificador*** sempre faz ***previsões*** completamente confiantes das classes (i.e., 0 ou 1), mesmo para exemplos muito próximos da ***fronteira de decisão***.
- Em muitas situações, nós precisamos de previsões mais graduadas, que indiquem incertezas quanto à classificação.
- Todos esses problemas são resolvidos com a ***suavização*** da ***função de limiar rígido*** através de sua aproximação por uma função que seja **contínua**, **diferenciável** e **assuma valores reais dentro do intervalo de 0 a 1**.

Classificação com função de limiar logístico

- A ***função logística*** (ou ***sigmóide***), mostrada na figura ao lado e definida como

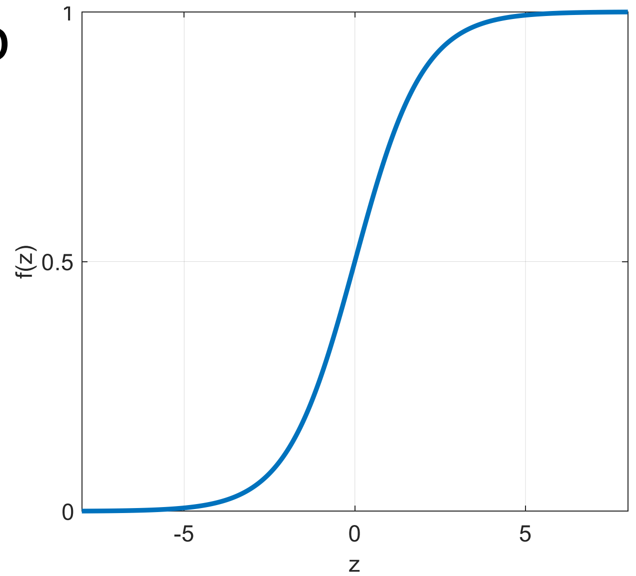
$$Logistic(z) = f(z) = \frac{1}{1+e^{-z}} \in [0, 1],$$

apresenta tais propriedades matemáticas.

- Utilizando a ***função logística*** como ***função de limiar***, temos

$$h_{\boldsymbol{a}}(\boldsymbol{x}) = Logistic\big(g(\boldsymbol{x})\big) = \frac{1}{1+e^{-g(\boldsymbol{x})}} \in [0, 1].$$

- $g(\boldsymbol{x})$ pode ser um ***hiperplano***, um ***polinômio***, etc.
- A saída será um número real entre 0 e 1, o qual pode ser interpretado como uma ***probabilidade*** de um dado exemplo pertencer à classe C_2 (ou seja, à ***classe positiva***).
- A nova ***função hipótese***, $h_{\boldsymbol{a}}(\boldsymbol{x})$, forma uma ***fronteira de decisão suave***, a qual confere uma probabilidade igual a 0.5 para exemplos em cima da ***fronteira de decisão*** e se aproxima de 0 ou 1 conforme a posição do exemplo se distancia da ***fronteira de decisão***.



A função logística realiza um mapeamento $\mathbb{R} \rightarrow [0,1]$.

Quanto mais longe da ***fronteira de decisão***, mais próximo o valor de saída da ***função hipótese*** será de 0 ou de 1 e, portanto, mais certeza teremos sobre uma classificação.

Em resumo, quanto mais longe da fronteira, maior será o valor absoluto de $g(\boldsymbol{x})$.

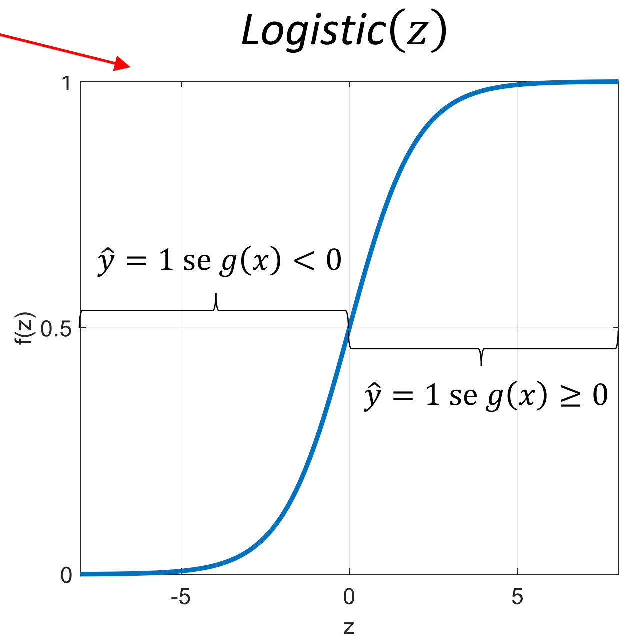
Regressão logística

- Esse classificador com função de ***limiar logística*** é conhecido como ***regressor logístico***.
- O ***regressor logístico*** estima a ***probabilidade*** de um exemplo pertencer a uma classe específica.
  - Por exemplo, qual é a probabilidade de uma dado email ser um spam?
- O ***regressor logístico*** é um algoritmo usado para ***classificação binária*** (i.e., classificação entre duas classes, C_1 e C_2), mas para isso, precisamos quantizar sua saída.
- Geralmente, se quantiza a saída da ***função hipótese***, $h_{\boldsymbol{a}}(\boldsymbol{x})$, nos valores 0 ou 1.
- Se a ***probabilidade*** estimada para um exemplo for igual ou maior que 50%, o classificador ***prediz*** que o exemplo pertence à ***classe positiva***, rotulada como 1, caso contrário ***prediz*** que pertence à ***classe negativa***, rotulada como 0.
- Ou seja, a saída ***quantizada*** do ***regressor logístico*** é dada por

$$Classe = \hat{y} = \begin{cases} 0 \text{ (classe } C_1 - \text{Negativa), se } h_{\boldsymbol{a}}(\boldsymbol{x}) < 0.5 \\ 1 \text{ (classe } C_2 - \text{Positiva)}, se\ h_{\boldsymbol{a}}(\boldsymbol{x}) \geq 0.5 \end{cases}$$

Regressão logística

- Note que $Logistic(z) < 0.5$ quando $z < 0$ e $Logistic(z) \geq 0.5$ quando $z \geq 0$, portanto, o modelo de ***regressão logística*** prediz a classe positiva, C_2 (i.e., $\hat{y} = 1$), se $g(\boldsymbol{x}) \geq 0$ e a classe negativa, C_1 (i.e., $\hat{y} = 0$), se $g(\boldsymbol{x}) < 0$.
- A ***regressão logística*** funciona usando uma ***combinação linear*** dos ***atributos***, para que várias fontes de informação (i.e., atributos) possam ditar a saída do modelo.
- Os ***parâmetros do modelo*** são os ***pesos*** associados aos vários ***atributos*** e representam a importância relativa de cada ***atributo*** para o resultado.
- Mesmo sendo uma técnica bastante simples, a ***regressão logística*** é muito utilizada em várias aplicações do mundo real em áreas como medicina, marketing, análise de crédito, etc.
- Além disto, toda a teoria por trás da ***regressão logística*** foi a base para a criação das primeiras ***redes neurais***.



Exemplos: classificar críticas de filmes como positivas ou negativas, probabilidade de um paciente desenvolver uma doença, detecção de spam, classificar transações como fraudulentas ou não, etc.

Propriedades da regressão logística

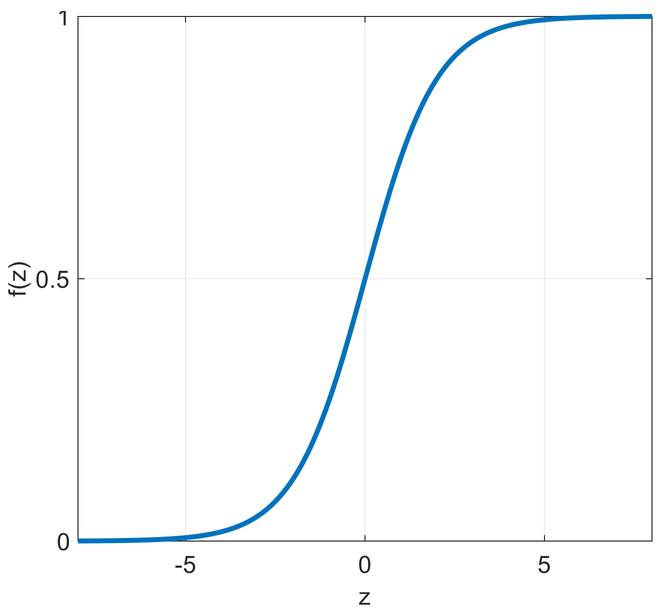
- Os valores de saída da ***função hipótese***, $h_{\boldsymbol{a}}(\boldsymbol{x})$, ficam restritos ao intervalo $0 \leq h_{\boldsymbol{a}}(\boldsymbol{x}) \leq 1$.
- A saída de $h_{\boldsymbol{a}}(\boldsymbol{x})$ representa a ***probabilidade*** do vetor de atributos $\boldsymbol{x}$ pertencer à classe positiva, C_2, para um dado vetor de pesos, $\boldsymbol{a}$.
- Ou seja, $h_{\boldsymbol{a}}(\boldsymbol{x})$ dá a probabilidade condicional da ***classe positiva***, C_2:

$$h_{\boldsymbol{a}}(\boldsymbol{x}) = P(C_2 \mid \boldsymbol{x}; \boldsymbol{a}).$$

- Consequentemente, o complemento de $h_{\boldsymbol{a}}(\boldsymbol{x})$ ou seja,

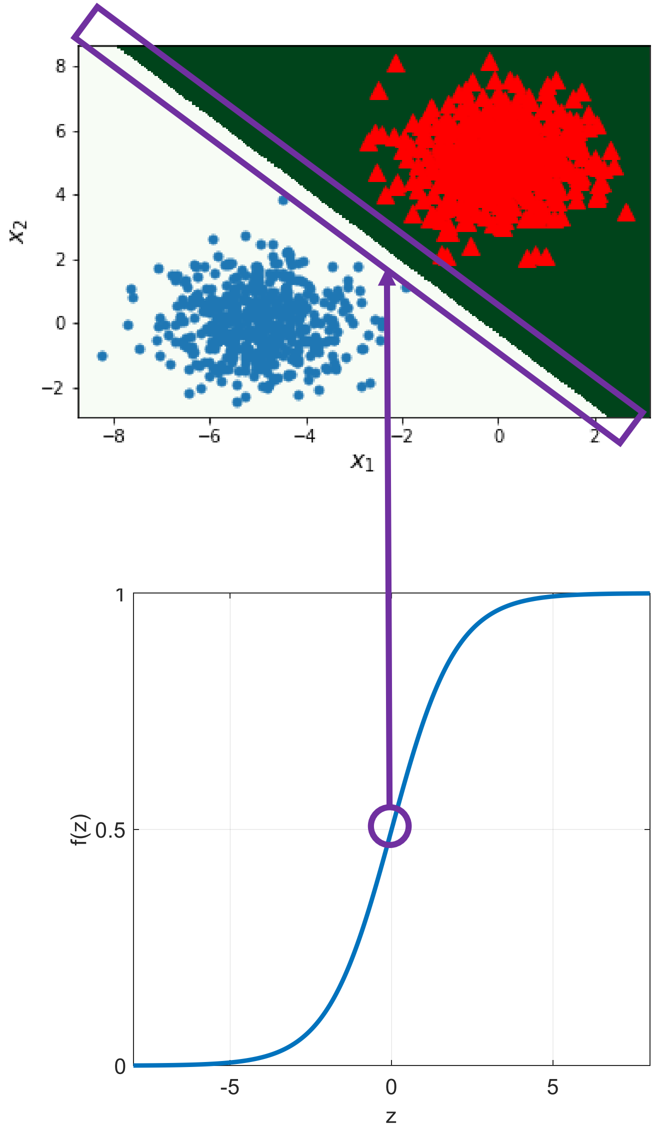
$$(1 - h_{\boldsymbol{a}}(\boldsymbol{x})) = P(C_1 \mid \boldsymbol{x}; \boldsymbol{a}),$$

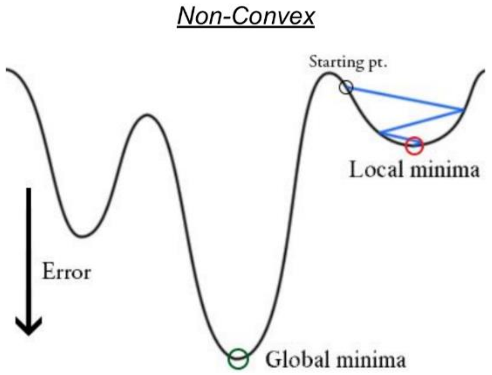
é a probabilidade condicional da ***classe negativa***, C_1.



Propriedades da regressão logística

- A ***fronteira de decisão*** é determinada quando há uma ***indecisão*** entre as classes, ou seja, quando $P(C_1 \mid \boldsymbol{x}; \boldsymbol{a}) = P(C_2 \mid \boldsymbol{x}; \boldsymbol{a})$, que ocorre quando $P(C_2 \mid \boldsymbol{x}; \boldsymbol{a}) = h_{\boldsymbol{a}}(\boldsymbol{x}) = 0.5$.
- Observando a figura da ***função logística***, nós percebemos que $Logistic(z) = 0.5$ quando $z = 0$.
- Ou seja, quando $g(\boldsymbol{x}) = 0$ ($\boldsymbol{x}$ em cima da ***fronteira de decisão***), a probabilidade de $\boldsymbol{x}$ pertencer à classe C_1ou C_2 é de 50% para as duas classes, indicando que o classificador está indeciso.





Função de erro

- Para treinarmos um ***regressor logístico*** e encontrarmos os ***pesos*** da ***função hipótese***, nós precisamos, assim como fizemos com a ***regressão linear***, definir uma ***função de erro***.
- Porém, adotar o ***erro quadrático médio*** como ***função de erro*** não é uma boa escolha para a ***adaptação dos pesos*** no caso da ***regressão logística*** como veremos a seguir.
- A ***função de erro***, $J_e(\boldsymbol{a})$, utilizando o ***erro quadrático médio*** é dada por

$$J_e(\boldsymbol{a}) = \frac{1}{N}\sum_{i=1}^{N}\left(y(i) - h_{\boldsymbol{a}}(\boldsymbol{x})\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y(i) - \mathit{Logistic}\big(g(\boldsymbol{x})\big)\right)^2.$$

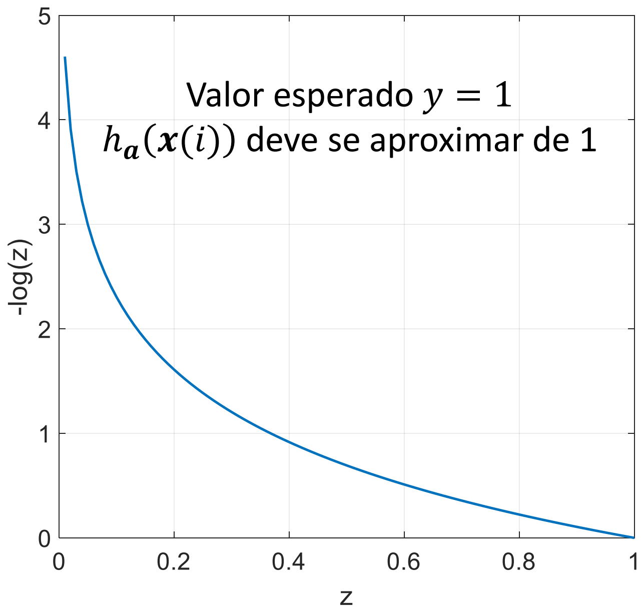
- Como *Logistic(.)* é uma função ***não-linear***, $J_e(\boldsymbol{a})$ não será, consequentemente, uma função ***convexa***, de forma que a ***superfície de erro*** poderá apresentar vários mínimos locais que vão dificultar o aprendizado (e.g., o algoritmo pode ficar preso em um mínimo local).
- **Ideia**: encontrar uma ***função de erro*** que tenha ***superfície de erro*** resultante ***convexa***.
- Uma proposta ***intuitiva*** para a ***função de erro para cada exemplo*** de entrada é dada por

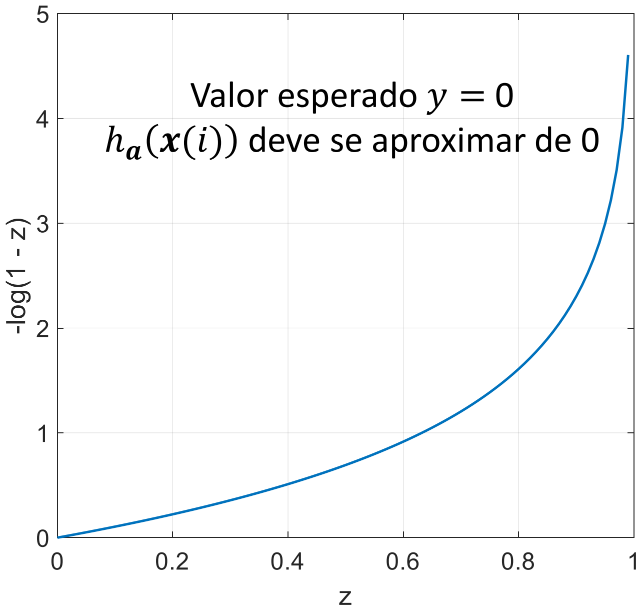
$$Erro\big(h_{\boldsymbol{a}}(\boldsymbol{x}(i)); y(i)\big) = \begin{cases} -\log(h_{\boldsymbol{a}}(\boldsymbol{x}(i))), & \text{se } y(i) = 1 \\ -\log\big(1 - h_{\boldsymbol{a}}(\boldsymbol{x}(i))\big), & \text{se } y(i) = 0 \end{cases}$$

onde $y(i)$ é o i-ésimo valor esperado.

- Veremos a seguir uma justificativa para esta escolha.

Função de erro





- As figuras ao lado mostram as duas situações possíveis para a ***função de erro***.
- Como podemos observar, a penalização aplicada a cada saída reflete o ***erro de classificação***.

- O uso dessa ***função de erro*** faz sentido pois:
  - O valor de $-\log(z)$ se torna muito grande quando z se aproxima de 0, então o erro será grande se o classificador estimar uma probabilidade próxima a 0 para um exemplo positivo (i.e., pertencente à classe C_2).
  - O valor de $-\log(1-z)$ será muito grande se o classificador estimar uma probabilidade próxima de 1 para um exemplo negativo (i.e., pertencente à classe C_1) .
  - Por outro lado, $-\log(z)$ se torna próximo de 0 quando z se aproxima de 1, portanto, o erro será próximo de 0 se a probabilidade estimada for próxima de 1 para um exemplo positivo.
  - O valor $-\log(1-z)$ se torna próximo de 0 quando z se aproxima de 0, portanto, o erro será próximo de 0 para um exemplo negativo.

Função de erro

- Nós podemos unir a ***função de erro*** para ***cada exemplo*** em uma expressão única, dada por

$$Erro\left(h_{\boldsymbol{a}}(\boldsymbol{x}(i)); y(i)\right) = \underbrace{-y(i)\log\left(h_{\boldsymbol{a}}(\boldsymbol{x}(i))\right)}_{\text{Só exerce influência no erro se } y(i)=1} \underbrace{-(1-y(i))\log\left(1-h_{\boldsymbol{a}}(\boldsymbol{x}(i))\right)}_{\text{Só exerce influência no erro se } y(i)=0}.$$

- Com isto, podemos definir a seguinte ***função de erro médio***

$$J_e(\boldsymbol{a}) = -\frac{1}{N}\sum_{i=0}^{N-1} y(i)\log\left(h_{\boldsymbol{a}}(\boldsymbol{x}(i))\right) + \left(1-y(i)\right)\log\left(1-h_{\boldsymbol{a}}(\boldsymbol{x}(i))\right).$$

- A má notícia é que não existe uma ***equação de forma fechada*** para encontrar os ***pesos*** que minimizem essa ***função de erro*** (ou seja, não há um equivalente da ***equação normal***).
- A boa notícia é que essa ***função de erro*** é ***convexa*** e, portanto, é garantido que o algoritmo do ***gradiente descendente*** encontre o mínimo global (dado que a ***taxa de aprendizagem*** não seja muito grande e você espere tempo suficiente).

Processo de treinamento

- Portanto, da mesma forma como fizemos com a ***regressão linear***, usamos o algoritmo do ***gradiente descendente*** para encontrar os ***pesos*** que ***minimizam*** a ***função de erro médio***.
- A ***atualização iterativa*** dos ***pesos*** é dada por

$$\boldsymbol{a} = \boldsymbol{a} - \alpha \frac{\partial J_e(\boldsymbol{a})}{\partial \boldsymbol{a}}.$$

Aqui consideramos $g(\boldsymbol{x})$ como sendo a equação de um ***hiperplano***: $g(\boldsymbol{x}) = \sum_{k=0}^{K} a_k x_k$, mas o resultado pode ser diretamente estendido para polinômios.

- O ***vetor gradiente*** da ***função de erro médio*** é dado por

$$\frac{\partial J_e(\boldsymbol{a})}{\partial \boldsymbol{a}} = -\frac{1}{N}\sum_{i=0}^{N-1}\left[y(i) - h_{\boldsymbol{a}}\big(\boldsymbol{x}(i)\big)\right]\boldsymbol{x}(i)^T = -\frac{1}{N}\boldsymbol{X}^T(\boldsymbol{y} - \widehat{\boldsymbol{y}}).$$

Forma matricial: $\boldsymbol{X} \in \mathbb{R}^{\mathrm{N}\times \mathrm{K}+1}$, $\boldsymbol{y}$ e $\widehat{\boldsymbol{y}} = h_{\boldsymbol{a}}\big(\boldsymbol{x}(i)\big) \in \mathbb{R}^{\mathrm{N}\times 1}$

- Percebam que o ***vetor gradiente*** da ***função de erro médio*** para a ***regressão logística*** é idêntico àquele obtido para a ***regressão linear*** utilizando a função de ***erro quadrático médio***.
- O ***vetor gradiente*** da ***função de erro médio*** vai variar dependendo da ***função discriminante*** adotada. Vejamos alguns exemplos na sequência.

Vetor Gradiente

- O ***vetor gradiente*** da ***função de erro médio*** quando $g(\boldsymbol{x}) = a_0 + a_1x_1 + a_2x_2$ (equação de uma reta) é dado por

$$\frac{\partial J_e(\boldsymbol{a})}{\partial \boldsymbol{a}} = -\frac{1}{N}\sum_{i=0}^{N-1}\left[y(i) - h_{\boldsymbol{a}}\big(\boldsymbol{x}(i)\big)\right]\boldsymbol{x}(i)^T = -\frac{1}{N}\boldsymbol{X}^T(\boldsymbol{y} - \widehat{\boldsymbol{y}}),$$

onde $\boldsymbol{X} = [\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2] \in \mathbb{R}^{\mathrm{N \times K+1}}$, $\boldsymbol{x}_0, \boldsymbol{x}_1$, e $\boldsymbol{x}_2 \in \mathbb{R}^{\mathrm{N \times 1}}$ e $\boldsymbol{y}$ e $\widehat{\boldsymbol{y}} \in \mathbb{R}^{\mathrm{N \times \mathbf{1}}}$.

- O ***vetor gradiente*** da ***função de erro médio*** quando $g(\boldsymbol{x}) = a_0 + x_1^2 + x_2^2$(equação de um círculo) é dado por

$$\frac{\partial J_e(\boldsymbol{a})}{\partial \boldsymbol{a}} = -\frac{1}{N}\sum_{i=0}^{N-1}\left[y(i) - h_{\boldsymbol{a}}\big(\boldsymbol{x}(i)\big)\right]\boldsymbol{x}(i)^T = -\frac{1}{N}\boldsymbol{X}^T(\boldsymbol{y} - \widehat{\boldsymbol{y}}),$$

onde $\boldsymbol{X} = [\boldsymbol{x}_0, \boldsymbol{x}_1^2, \boldsymbol{x}_2^2] \in \mathbb{R}^{\mathrm{N \times K+1}}$, $\boldsymbol{x}_0, \boldsymbol{x}_1^2$, e $\boldsymbol{x}_2^2 \in \mathbb{R}^{\mathrm{N \times 1}}$ e $\boldsymbol{y}$ e $\widehat{\boldsymbol{y}} \in \mathbb{R}^{\mathrm{N \times \mathbf{1}}}$.

Vetor Gradiente

- O ***vetor gradiente*** da ***função de erro médio*** quando $g(\boldsymbol{x}) = a_0 + a_1 x_1 * x_2$ (equação de uma hipérbole retangular) é dado por

$$\frac{\partial J_e(\boldsymbol{a})}{\partial \boldsymbol{a}} = -\frac{1}{N}\sum_{i=0}^{N-1}\left[y(i) - h_{\boldsymbol{a}}\big(\boldsymbol{x}(i)\big)\right]\boldsymbol{x}(i)^T = -\frac{1}{N}\boldsymbol{X}^T(\boldsymbol{y} - \hat{\boldsymbol{y}}),$$

onde $\boldsymbol{X} = [\boldsymbol{x}_0, \boldsymbol{x}_1 \odot \boldsymbol{x}_2] \in \mathbb{R}^{\mathrm{N \times K+1}}$, $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2,$ e $\boldsymbol{x}_1 \odot \boldsymbol{x}_2 \in \mathbb{R}^{\mathrm{N \times 1}}$, $\boldsymbol{y}$ e $\hat{\boldsymbol{y}} \in \mathbb{R}^{\mathrm{N} \times \mathbf{1}}$ e $\odot$ é a multiplicação elemento-a-elemento.

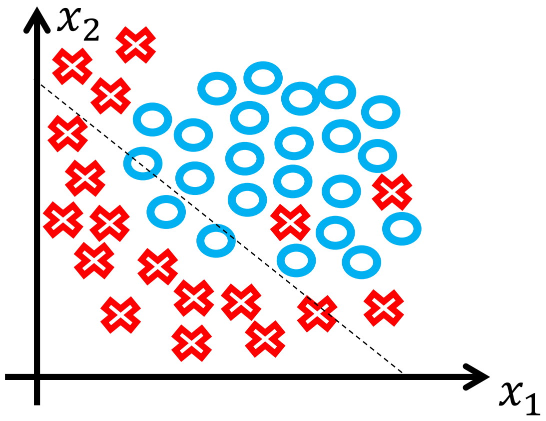
- Agora, de posse do ***vetor gradiente***, podemos usá-lo com o ***gradiente descendente*** (nas versões em batelada, estocástico ou mini-batch) para atualizar os pesos.

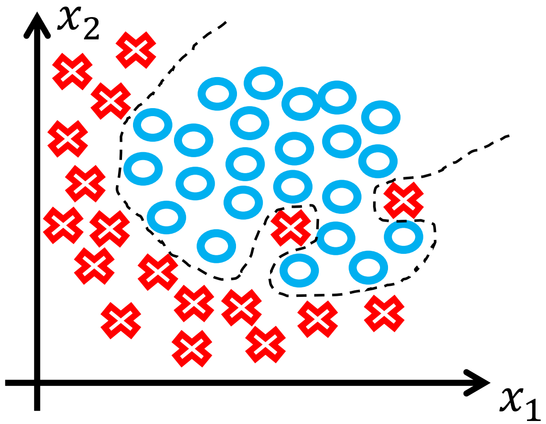
$$\boldsymbol{a} = \boldsymbol{a} - \alpha \frac{\partial J_e(\boldsymbol{a})}{\partial \boldsymbol{a}}.$$

Exemplo: logistic_regression_with_gradient_descent.ipynb

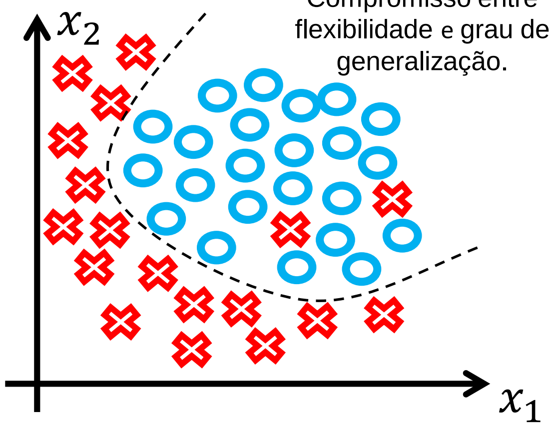
Observações

- Como vimos, a ***função discriminante***, $g(\boldsymbol{x})$, pode também assumir a forma de um ***polinômio***, mas, muitas vezes, nós não sabemos qual a ***melhor ordem*** para este polinômio.
- Assim, como nós discutimos no caso da ***regressão linear***, modelos de ***regressão logística*** também estão sujeitos à ocorrência de ***sobreajuste*** e ***subajuste***. Vejam as figuras ao lado.
  - Na primeira figura, a ***falta de flexibilidade*** da reta usada faz com que o erro de classificação seja alto.
  - Na segunda figura, a ***flexibilidade excessiva*** do modelo (explorando um polinômio de ordem elevada) dá origem a contorções na ***fronteira de decisão*** na tentativa de minimizar o erro de classificação junto aos ***dados de treinamento***. Porém, o modelo ficou mais susceptível a erros de classificação para dados inéditos, ou seja, não irá generalizar bem.
  - Já a última figura mostra o que seria uma boa ***hipótese de classificação***.
- Por isso, técnicas de ***regularização*** (e.g., LASSO, Ridge, Elastic-Net, Early-stop) assim como de ***validação cruzada*** também podem ser empregadas durante o treinamento quando não conhecemos a melhor ordem para o polinômio da ***função discriminante***, $g(\boldsymbol{x})$.





Compromisso entre flexibilidade e grau de generalização.



Tarefas

- **Quiz**: “*T320 - Quiz - Classificação (Parte III)*” que se encontra no MS Teams.
- **Exercício Prático**: **Laboratório #3**.
  - Pode ser acessado através do link acima (Google Colab) ou no GitHub.
  - Se atentem aos prazos de entrega.
  - Instruções para resolução e entrega dos laboratórios.
  - **Laboratórios podem ser resolvidos em grupo, mas as entregas devem ser individuais.**

Obrigado!

Encontrando o vetor gradiente

- Antes de encontrarmos o ***vetor gradiente*** de $J_e(\boldsymbol{a})$, vamos reescrever a ***função de erro*** utilizando as seguintes equivalências

$$\log(h_{\boldsymbol{a}}(\boldsymbol{x}(i))) = \log\left(\frac{1}{1+e^{-\boldsymbol{x}(i)^T\boldsymbol{a}}}\right) = -\log\left(1+e^{-\boldsymbol{x}(i)^T\boldsymbol{a}}\right),$$

$$\log(1-h_{\boldsymbol{a}}(\boldsymbol{x}(i))) = \log\left(1-\frac{1}{1+e^{-\boldsymbol{x}(i)^T\boldsymbol{a}}}\right) = -\boldsymbol{x}(i)^T\boldsymbol{a} - \log\left(1+e^{-\boldsymbol{x}(i)^T\boldsymbol{a}}\right).$$

- Assim, a nova expressão para a ***função de erro médio*** é dada por

$$J_e(\boldsymbol{a}) = -\frac{1}{N}\sum_{i=0}^{N-1} -y(i)\log\left(1+e^{-\boldsymbol{x}(i)^T\boldsymbol{a}}\right) + (1 - y(i))\left[-\boldsymbol{x}(i)^T\boldsymbol{a} - \log\left(1+e^{-\boldsymbol{x}(i)^T\boldsymbol{a}}\right)\right]$$

Encontrando o vetor gradiente

- O termo $-y(i)\log\left(1+e^{-\boldsymbol{x}(i)^T\boldsymbol{a}}\right)$ é cancelado com um dos elementos gerados a partir do produto envolvido no segundo termo, de forma que
$$J_e(\boldsymbol{a}) = -\frac{1}{N}\sum_{i=0}^{N-1} -\boldsymbol{x}(i)^T\boldsymbol{a} + y(i)\boldsymbol{x}(i)^T\boldsymbol{a} - \log\left(1+e^{-\boldsymbol{x}(i)^T\boldsymbol{a}}\right).$$
- Se $-\boldsymbol{x}(i)^T\boldsymbol{a} = -\log\left(e^{\boldsymbol{x}(i)^T\boldsymbol{a}}\right)$, então
$$-\boldsymbol{x}(i)^T\boldsymbol{a} - \log\left(1+e^{-\boldsymbol{x}(i)^T\boldsymbol{a}}\right) = -\log\left(1+e^{\boldsymbol{x}(i)^T\boldsymbol{a}}\right).$$
- Desta forma, a ***função de erro médio*** se torna
$$J_e(\boldsymbol{a}) = -\frac{1}{N}\sum_{i=0}^{N-1} y(i)\boldsymbol{x}(i)^T\boldsymbol{a} - \log\left(1+e^{\boldsymbol{x}(i)^T\boldsymbol{a}}\right).$$
- Em seguida, encontramos o ***vetor gradiente*** de cada termo da equação acima.

Encontrando o vetor gradiente

- Assim, o ***vetor gradiente*** do primeiro termo da equação anterior é dado por

$$\frac{\partial[y(i)\boldsymbol{x}(i)^T\boldsymbol{a}]}{\partial\boldsymbol{a}} = y(i)\boldsymbol{x}(i)^T$$

- O ***vetor gradiente*** do segundo termo da equação anterior é dado por

$$\begin{aligned}\frac{\partial\left[\log\left(1+e^{\boldsymbol{x}(i)^T\boldsymbol{a}}\right)\right]}{\partial\boldsymbol{a}} &= \frac{1}{1+e^{\boldsymbol{x}(i)^T\boldsymbol{a}}}e^{\boldsymbol{x}(i)^T\boldsymbol{a}}\boldsymbol{x}(i)^T \\ &= \frac{1}{1+e^{-\boldsymbol{x}(i)^T\boldsymbol{a}}}\boldsymbol{x}(i)^T \\ &= h_{\boldsymbol{a}}(\boldsymbol{x}(i))\boldsymbol{x}(i)^T.\end{aligned}$$

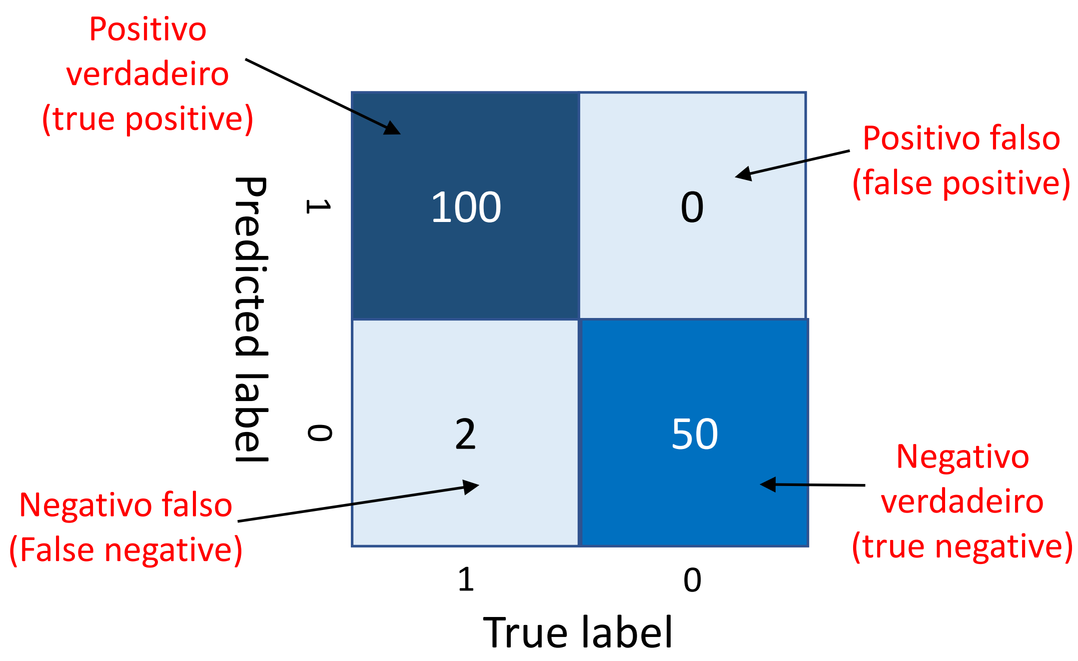
- Usamos a ***regra da cadeia*** para encontrar o vetor gradiente do segundo termo.

Encontrando o vetor gradiente

- Portanto, combinando os 2 resultados anteriores, temos que o ***vetor gradiente*** da ***função de erro médio*** é dado por

$$\frac{\partial J_e(\boldsymbol{a})}{\partial \boldsymbol{a}} = -\frac{1}{N}\sum_{i=0}^{N-1}\left[y(i) - h_{\boldsymbol{a}}\left(\boldsymbol{x}(i)\right)\right]\boldsymbol{x}(i)^T = -\frac{1}{N}\boldsymbol{X}^T(\boldsymbol{y} - \widehat{\boldsymbol{y}}).$$

Forma matricial: $\boldsymbol{X} \in \mathbb{R}^{\mathrm{N \times K+1}}$, $\boldsymbol{y}$ e $\widehat{\boldsymbol{y}} \in \mathbb{R}^{\mathrm{N \times \mathbf{1}}}$


Positivo
verdadeiro
(true positive)
Positivo falso
(false positive)
Predicted label
1
0
100
0
2
50
Negativo falso
(False negative)
Negativo
verdadeiro
(true negative)
1
0
True label