Regressão Logística

Renato Assunção - DCC - UFMG

Regressão Logística Simples

- Este modelo é uma extensão de regressão linear para o caso em que a variável Y possui distribuição binomial.
- Temos n exemplos ou instâncias de dados.
- Eles são independentes mas nào são i.d.
- A variável resposta Y_i é BINÁRIA: 0 ou 1.
- Temos uma ou mais variáveis explicativas (ou features) no vetor $\mathbf{x}_i$.
- Logística: modelo para descrever como a DISTRIBUIÇÃO de Y varia com $\mathbf{x}$.
- Isto é, descrever $Y|\mathbf{x}$, a distribuição de Y condicionada nos valores de $\mathbf{x}$.

Teste de Desenvolvimento de Denver

- Para detecção precoce de problemas.
- Aplicado em crianças entre o nascimento e os seis anos de idade, para confirmação de suspeitas na avaliação subjetiva do desenvolvimento e para sua monitorização em crianças com risco de apresentar alterações.
- O teste é composto por 125 itens, subdivididos em quatro domínios de funções: pessoal-social, motor-adaptativo, linguagem e motor grosseiro.
- Os itens incluem a coordenação do olho e da mão, manipulação de pequenos objetos, produção de som, capacidade de reconhecer, entender e usar a linguagem, controle do corpo para sentar, caminhar, pular etc.

Teste de Desenvolvimento de Denver

- Cada um dos 125 itens está representado por uma barra que contém as idades em que 25%, 50%, 75% e 90% das crianças estudadas apresentaram as habilidades sugeridas.
- Uma criança de idade x realiza alguns poucos itens indicados para sua faixa etária.
- O pediatra sabe que, dentre crianças com aquela idade x, uma certa porcentagem $p(x)$ executa corretamente a tarefa do item.
- Suponha que a criança sob exame não executou a tarefa. Se $p(x) = 0.25$, não há motivos para preocupação.
- Mas se $p(x) = 0.90$, pode haver motivo de preocupação e exames mais minuciosos são então indicados.
- Como estas idades críticas são determinadas para uso rotineiro nos consultórios? Com regressão logística.

Modelo para uma única tarefa

- Amostra de n crianças de diversas idades e *sem problemas de desenvolvimento* procuram executar a tarefa de um dos itens.
- $Y_i = 1$ denota o sucesso e $Y_i = 0$, o fracasso da i-ésima criança na execução da tarefa.
- No caso de crianças sem problemas de desenvolvimento, mais cedo ou mais tarde, todas acabam executando a tarefa sem erros.
- O sucesso ou fracasso depende principalmente da idade x da criança.
- Sendo velha o suficiente, a criança executa a tarefa.
- Por outro lado, se for muito nova ainda, é quase impossível executar a tarefa.
- As faixas etárias apropriadas variam com o tipo de tarefa.

Resultado da amostra

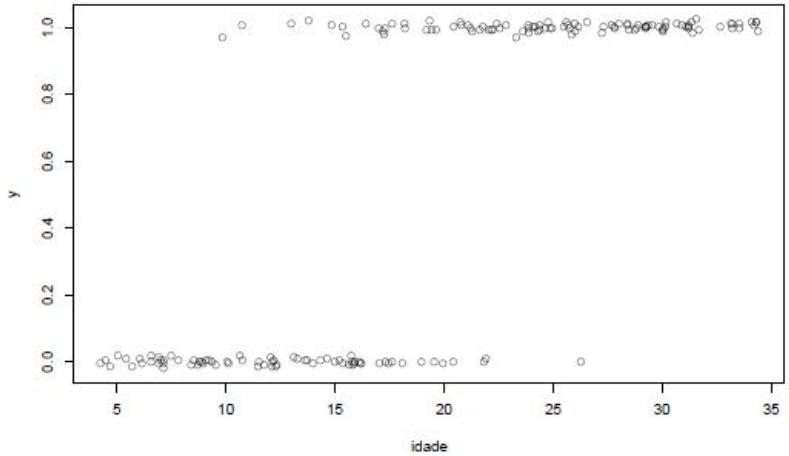


Figura: Dados de 173 crianças com informação de idade x_i em meses e variável y_i indicando sucesso ou fracasso na realização de uma tarefa. Dados estão deslocados de uma pequena quantidade aleatória para melhor visualização.

Modelo estatístico

- Resultados das crianças são independentes pois o sucesso ou fracasso de uma das crianças não afeta o desempenho das demais.
- $Y_1, \ldots, Y_n$ ensaios de Bernoulli independentes

$$Y_i = \begin{cases} 1, & \text{com probabilidade } p_i \\ 0, & \text{com probabilidade } 1 - p_i \end{cases}$$

- A probabilidade p_i vai variar de criança para criança.
- Queremos que esta variação ocorra em função de sua idade x. Isto é, queremos escrever $p_i = g(x_i)$, onde g é uma função matemática e x_i é a idade da i-ésima criança.
- Além disso, queremos que $g(x)$ seja crescente com x, que $g(x) \to 1$ quando x cresce e que $g(x) \to 0$ quando x diminui.

Escolhendo uma função

- Existem algumas escolhas populares para a função $g(x)$: logística, probit, log-log complementar.
- A mais usada é a função logística.
- Ela possui poucos parâmetros; é simples de entender; é flexível, ajustando-se facilmente a diferentes situações.
- Voltaremos às outras opções (probit, log-log complementar) mais tarde, no contexto mais geral de modelos lineares generalizados (GLM).

Função logística

- Temos

$$p(x) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x))}.$$

- Dependendo dos valores de β_0 e β_1 nós obtemos diferentes curvas

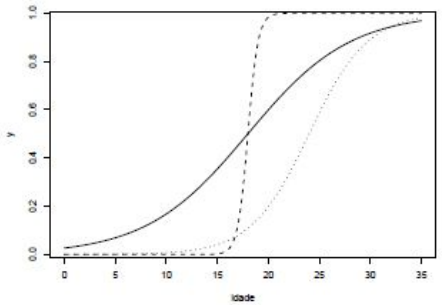


Figura: Três curvas logísticas. A curva em linha sólida possui $\beta_0 = -3.6$ e $\beta_1 = 0.2$. A curva em linha tracejada possui $\beta_0 = -368$ e $\beta_1 = 2.0$. A curva em linha pontilhada possui $\beta_0 = -8.4$ e $\beta_1 = 0.35$.

Dados (SLIDES DE ICD - Flávio e Pedro Olmo)

Lances do Lebron James. Observe a coluna

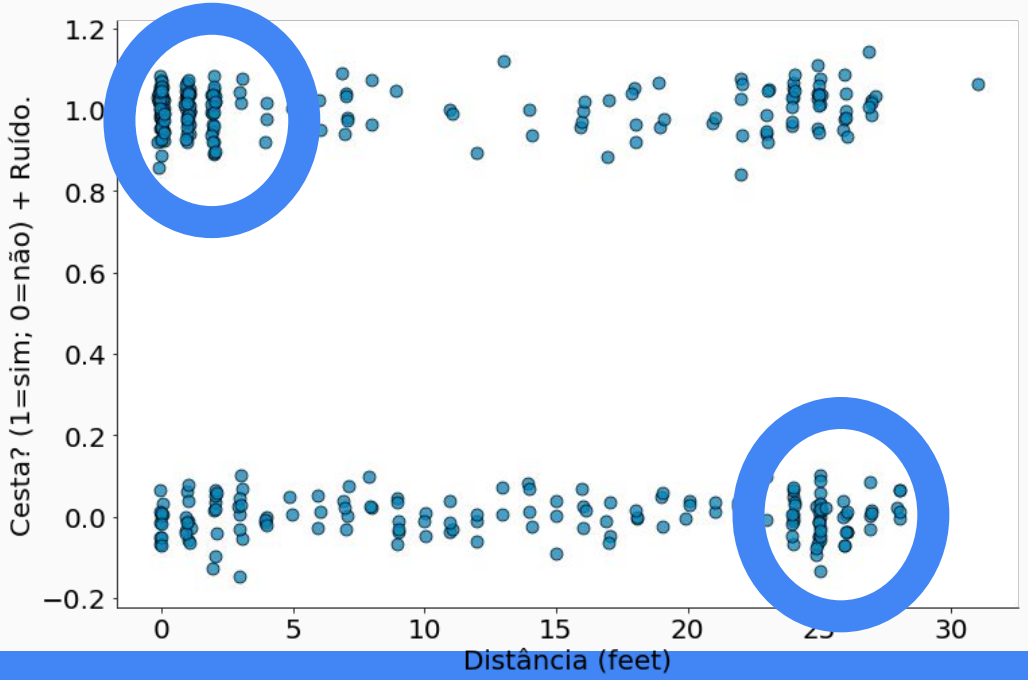
shot_distance → distância da cesta em pés

```
df = pd.read_csv('./lebron.csv')
df.head()
```

	game_date	minute	opponent	action_type	shot_type	shot_distance	shot_made
0	20170415	10	IND	Driving Layup Shot	2PT Field Goal	0	0
1	20170415	11	IND	Driving Layup Shot	2PT Field Goal	0	1
2	20170415	14	IND	Layup Shot	2PT Field Goal	0	1
3	20170415	15	IND	Driving Layup Shot	2PT Field Goal	0	1
4	20170415	18	IND	Alley Oop Dunk Shot	2PT Field Goal	0	1

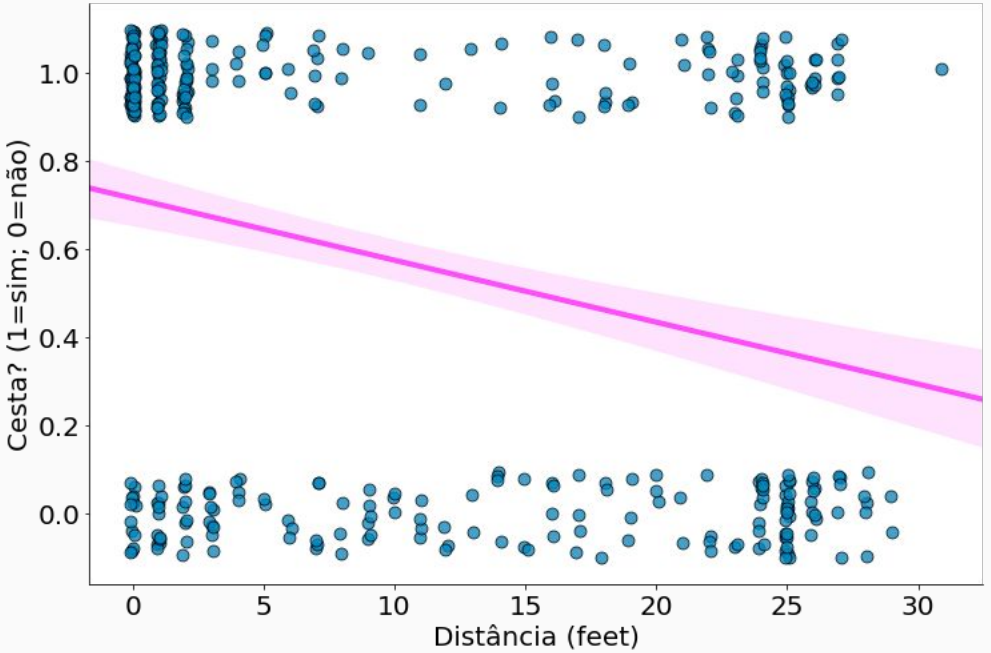
Gráfico de Dispersão

Parece que temos duas concentrações de pontos

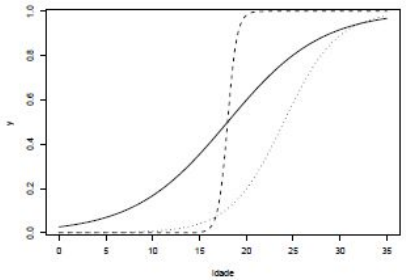


Regressão Linear

- Podemos executar uma regressão linear nos dados
- Vai capturar a tendência geral
- Neste caso, até funciona bem



Função logística

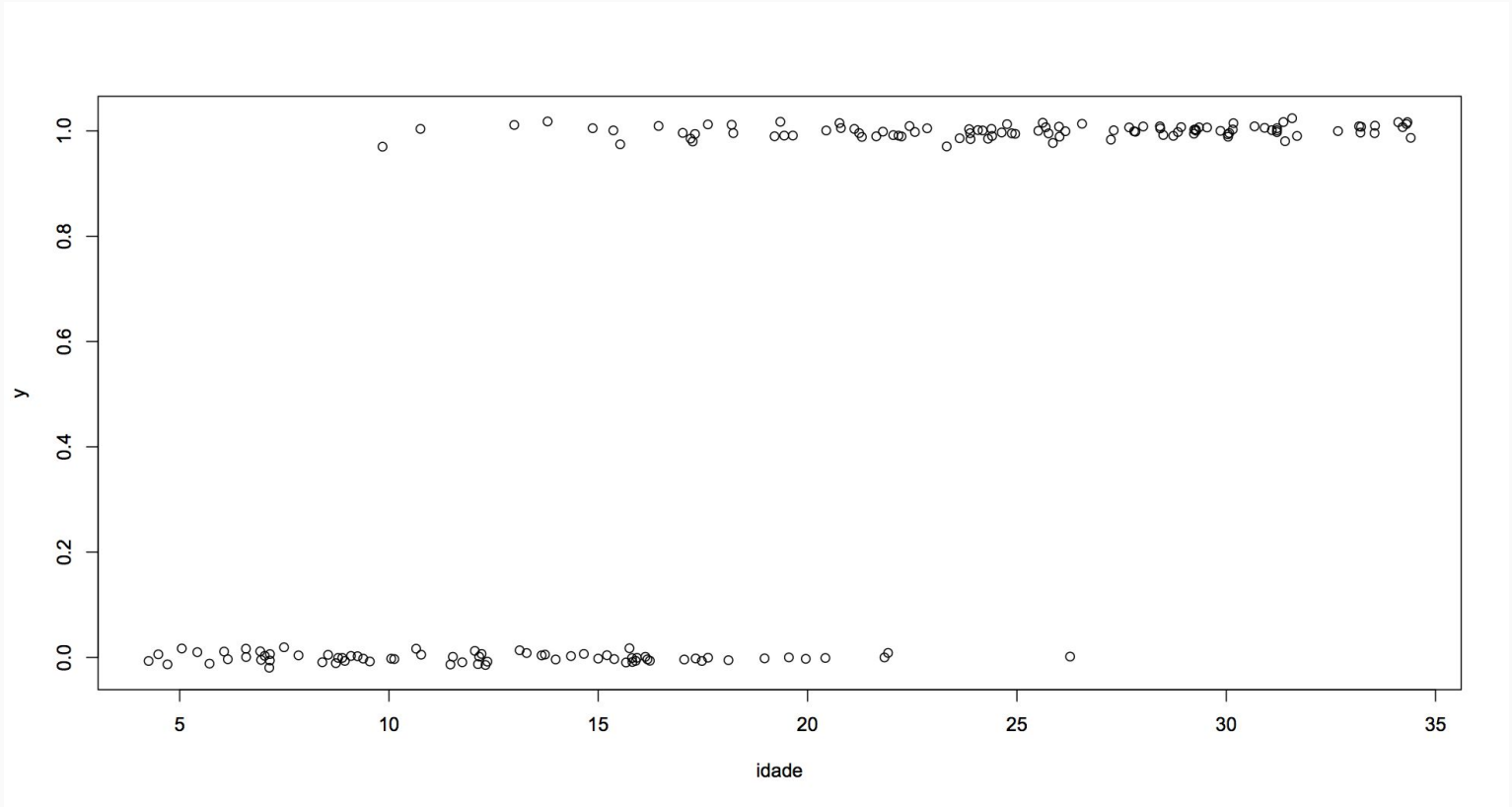


- O parâmetro β_0 está associado com o valor da probabilidade quando a idade for zero. Ele controla onde a curva logística vai se posicionar no eixo horizontal.
- O parâmetro $\beta_1/2$ é a inclinação da reta tangente à curva no ponto $\hat{x}$ em que $p(\hat{x}) = 1/2$.
- Mudar apenas $beta_0$ significa deslocar rigidamente a curva no eixo horizontal.
- Mudar apenas β_1 significa acelerar ou retardar a passagem do estágio de não execucção quase certa para o estágio de execução quase certa

Como escolher a melhor curva logística para ajustar aos dados?

- Várias perguntas:
  - Como obter os coeficientes de uma curva (regressão) logística?
  - Como escolher a "melhor" curva logística? "Melhor" em que sentido?
  - Como avaliar se o modelo logístico é um bom classificador?
  - Como generalizar o modelo se tivermos várias features?
    - E se a probabilidade depender também da escolaridade da mãe, do sexo da criança, ...

173 crianças

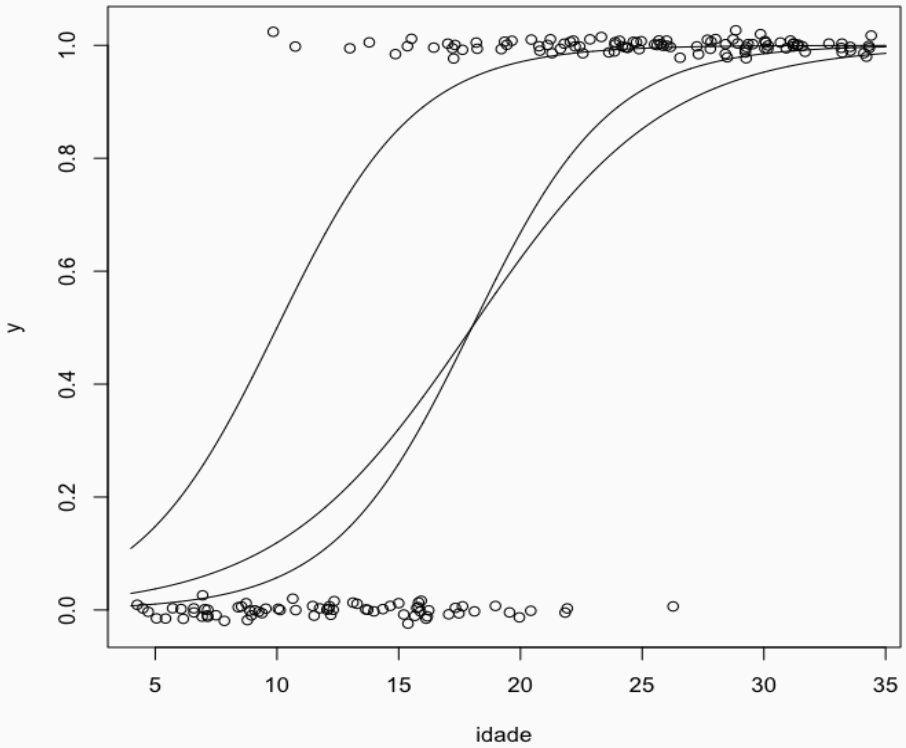


Função logística

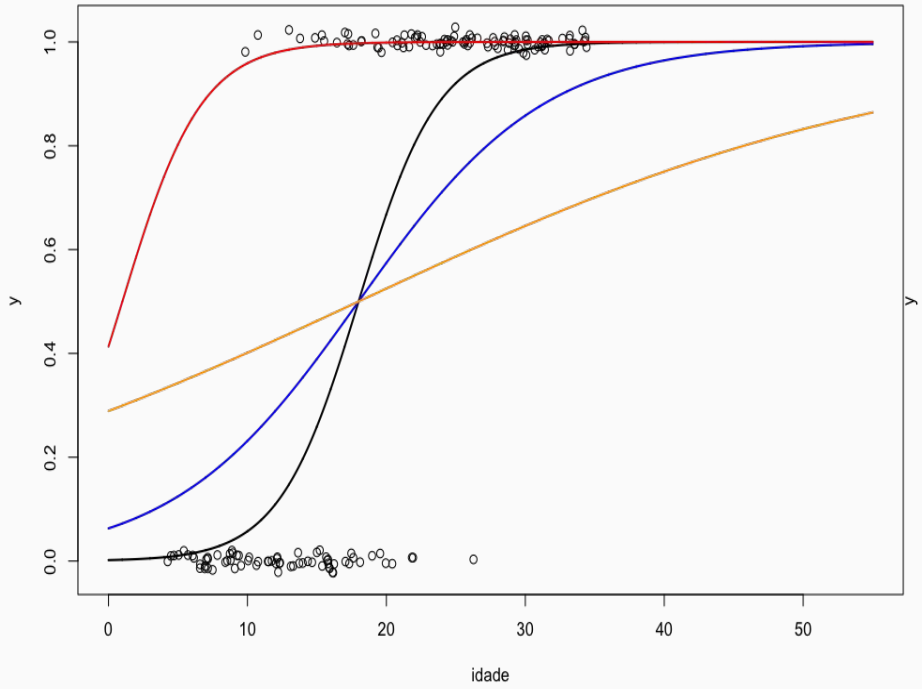
- A probabilidade de uma criança com idade x realizar a tarefa é
- $$\sigma(x) = \frac{1}{1+e^{-(w_0+w_1 x)}}$$

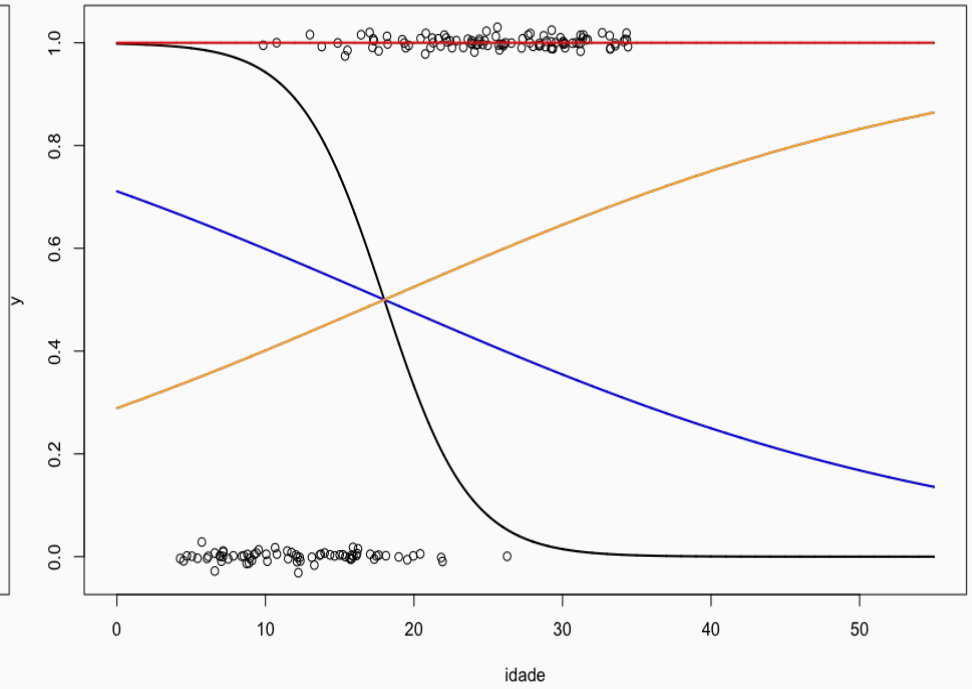
- Como escolher w_0 e w_1 compatíveis com os dados?

- Ideia: escolha w_0 e w_1 de tal forma que os dados realmente observados possam ser gerados pelo modelo.



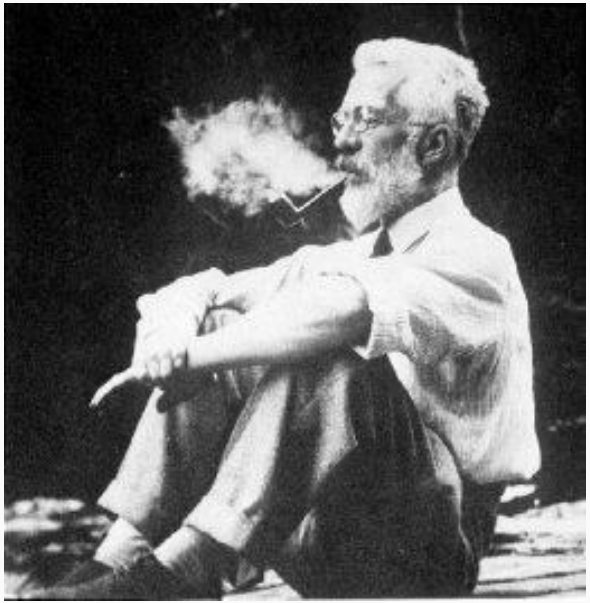
Diferentes parâmetros, diferentes curvas.

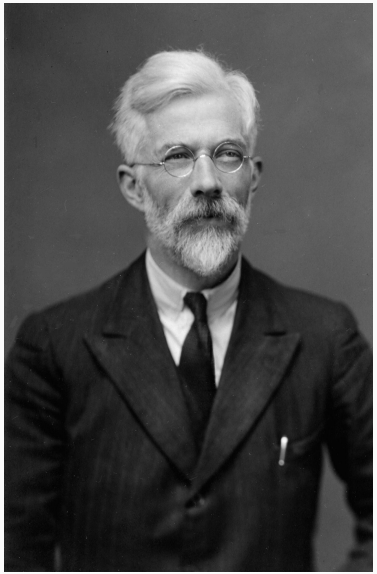




Ideia: Algumas das curvas são "compatíveis" com os dados.
Algumas curvas são verossímeis como modelo gerador dos dados observados.

- Método de máxima verossimilhança → para estimar parâmetros ou coeficientes com dados estatísticos
- Foi criado por Sir Ronald Fisher (1890 - 1962), o maior estatístico que já existiu.





E a luz se fez em 1922

- Fisher foi uma espécie de Isaac Newton da estatística, responsável pelos principais conceitos e resultados da inferência estatística, usados até hoje.
- Suas ideias principais foram publicadas de uma só vez, num artigo de 1922, *On the mathematical foundations of theoretical statistics*.
- Alguns dos principais conceitos e resultados usados até hoje:
  - verossimilhança,
  - suficiência
  - vício e eficiência de estimação
- são desse artigo maravilhoso (ele tinha 32 anos de idade).

IX. *On the Mathematical Foundations of Theoretical Statistics.*

By R. A. Fisher, M.A., *Fellow of Gonville and Caius College, Cambridge, Chief Statistician, Rothamsted Experimental Station, Harpenden.*

Communicated by Dr. E. J. Russell, F.R.S.

Received June 25,—Read November 17, 1921.

Contents.

Definitions.

Centre of Location.—That abscissa of a frequency curve for which the sampling errors of optimum location are uncorrelated with those of optimum scaling. (9.)

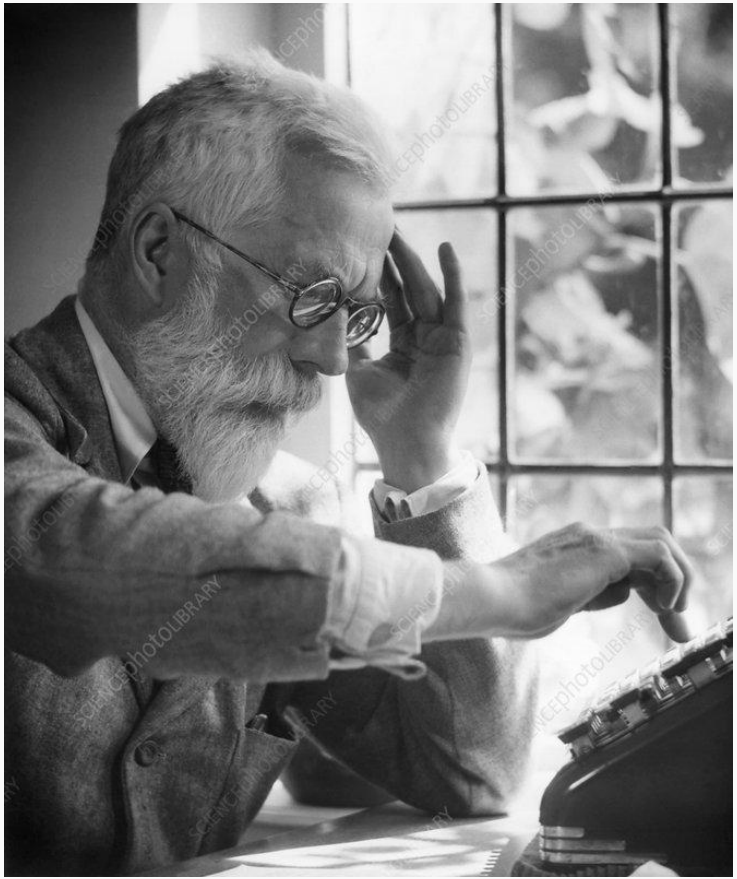
Consistency.—A statistic satisfies the criterion of consistency, if, when it is calculated from the whole population, it is equal to the required parameter. (4.)

Distribution.—Problems of distribution are those in which it is required to calculate the distribution of one, or the simultaneous distribution of a number, of functions of quantities distributed in a known manner. (3.)

Efficiency.—The efficiency of a statistic is the ratio (usually expressed as a percentage) which its intrinsic accuracy bears to that of the most efficient statistic possible. It

Mais um pouco de Fisher

- Fisher foi também um maiores geneticistas que já existiu
  - junto com Sewall Wright e Haldane, é responsável por juntar de forma coerente a teoria da evolução de Darwin e a teoria genética de Mendel (um quebra-cabeça complicado em 1920)
- Criador de:
  - teoria e prática do planejamento de experimentos (aleatorização, blocagem, quadrados-latinos, etc)
  - Análise de regressão linear (p-valores)
  - PCA
  - Análise discriminante
  - Teoria de valores extremos, etc etc etc etc etc



Verossimilhança = Likelihood

- Vimos algumas curvas logísticas "extremas".
- Dificilmente elas poderiam ter gerado os dados das crianças.
- Fisher: estas curvas extremas não são ***verossímeis***.
    - vero: verdadeiro, real, autêntico;
    - símil: semelhante, similar.
- algo é verossímil se parece verdadeiro,
    - se não repugna à verdade,
    - se é semelhante à verdade,
    - se é coerente o suficiente para se passar por verdade.
- Ao dizer que algo é verossímil, ***não*** dizemos que é verdadeiro.
- Verossímil = *parece verdadeiro* pois está de acordo com todas as evidências disponíveis

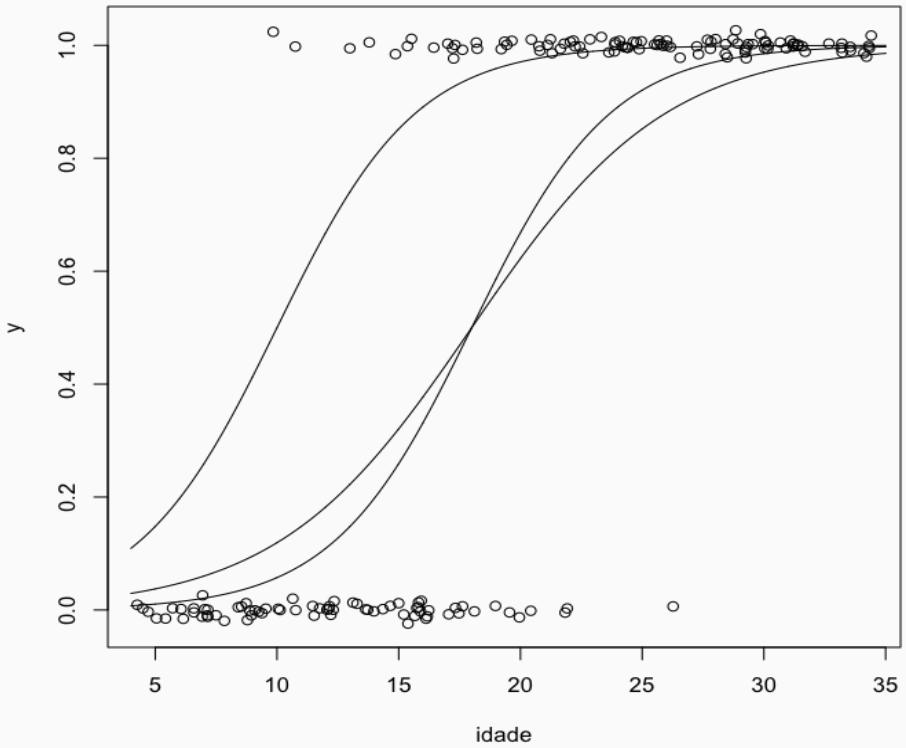
A verossimilhança do modelo logístico

- A probabilidade de uma criança com idade x realizar a tarefa é

$$\sigma(x) = \frac{1}{1+e^{-(w_0+w_1 x)}}$$

- Vamos fixar w_0 e w_1 → fixar uma curva

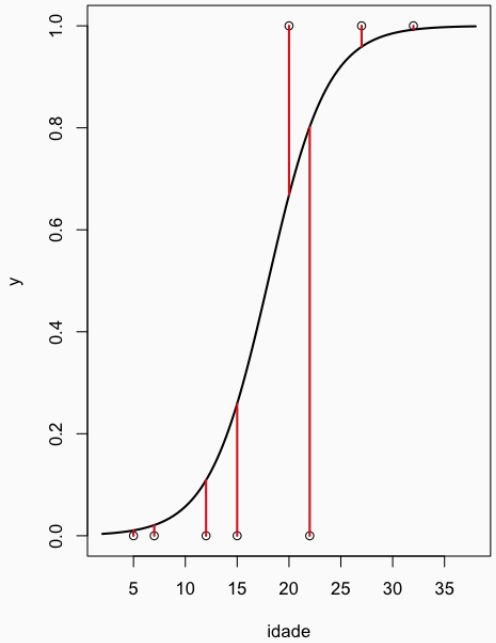
- Para esta curva fixada, obtenha a probabilidade de gerar os sucessos e fracassos realmente observados.

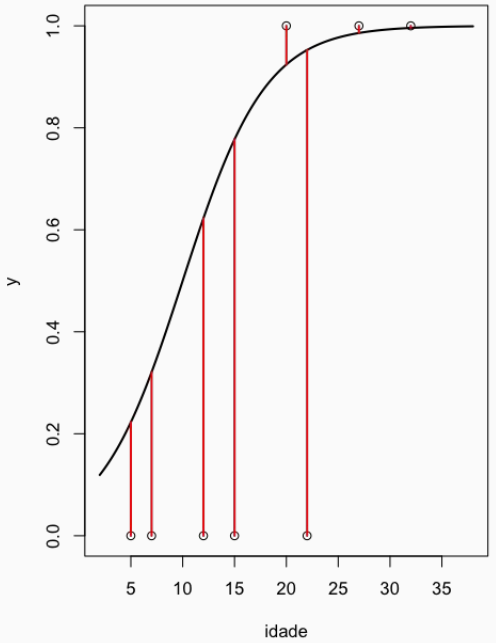


Duas curvas e suas probabilidades

- Para cada curva possível:
  - calcular a probabilidade de gerar os valores 0 ou 1 *realmente observados*
  - Multiplicar estas probabilidades (regra de indep de eventos: as crianças agem independentemente)
  - Obter a probabilidade para cada curva
  - Para qual curva esta probabilidade é máxima?

- Fazer exemplo no quadro comparando duas curvas com 5 pontos.





A função de verossimilhança

- Temos 5 crianças com idades x iguais a 5, 12, 22, 25, 30
- Os y's correspondentes são 0, 1, 0, 1, 1
- Se w_0 = -6.3 e w_1 = 0.35, obtenha a probabilidade de gerar os y's acima com o modelo logístico
- Para cada criança e para estas escolhas de w_0 e w_1, esta probabilidade é

$$\sigma(x) = \frac{1}{1+e^{-(-6.3+0.35x)}}$$

- Vamos refazer este cálculo obtendo esta probabilidade com diferentes valores de w_0 e w_1
- Esta probabilidade será uma função de w_0 e w_1
- $$L(w_0, w_1) = \mathbb{P}(Y_1 = 0, Y_2 = 1, Y_3 = 0, Y_4 = 1, Y_5 = 1 | w_0, w_1)$$

Obtendo a verossimilhança para $w_0 = -6.3$ e $w_1 = 0.35$

- Sejam $w_0 = -6.3$ e $w_1 = 0.35$
- Vamos obter
- $L(-6.3, 0.35) = \mathbb{P}(Y_1 = 0, Y_2 = 1, Y_3 = 0, Y_4 = 1, Y_5 = 1 | w_0 = -6.3, w_1 = 0.35)$
- O resultado de uma criança (sucesso ou fracasso) não afeta o resultados das demais crianças. São eventos independentes.

$$L(-6.3, 0.35) = \mathbb{P}(Y_1 = 0 | w_0 = -6.3, w_1 = 0.35)\mathbb{P}(Y_2 = 1 | w_0 = -6.3, w_1 = 0.35) \times$$
$$\times \mathbb{P}(Y_3 = 0 | w_0 = -6.3, w_1 = 0.35)\mathbb{P}(Y_4 = 1 | w_0 = -6.3, w_1 = 0.35)\mathbb{P}(Y_5 = 1 | w_0 = -6.3, w_1 = 0.35)$$

- Precisamos calcular cada uma das 5 probabilidades na expressão acima

Obtendo a verossimilhança para $w_0 = -6.3$ e $w_1 = 0.35$

- Queremos $L(-6.3, 0.35) = \mathbb{P}(Y_1 = 0, Y_2 = 1, Y_3 = 0, Y_4 = 1, Y_5 = 1 | w_0 = -6.3, w_1 = 0.35)$
- Isto é igual a

$$L(-6.3, 0.35) = \mathbb{P}(Y_1 = 0 | w_0 = -6.3, w_1 = 0.35)\mathbb{P}(Y_2 = 1 | w_0 = -6.3, w_1 = 0.35) \times$$
$$\times \mathbb{P}(Y_3 = 0 | w_0 = -6.3, w_1 = 0.35)\mathbb{P}(Y_4 = 1 | w_0 = -6.3, w_1 = 0.35)\mathbb{P}(Y_5 = 1 | w_0 = -6.3, w_1 = 0.35)$$

- Temos $\mathbb{P}(Y = 1 | w_0 = -6.3, w_1 = 0.35) = \frac{1}{1+e^{-(-6.3+0.35x)}}$
- e $\mathbb{P}(Y = 0 | w_0 = -6.3, w_1 = 0.35) = 1 - \frac{1}{1+e^{-(-6.3+0.35x)}} = \frac{1}{1+e^{-6.3+0.35x}}$
- A diferença entre as duas probabilidades acima está no expoente da exponencial

Obtendo a verossimilhança para $w_0 = -6.3$ e $w_1 = 0.35$

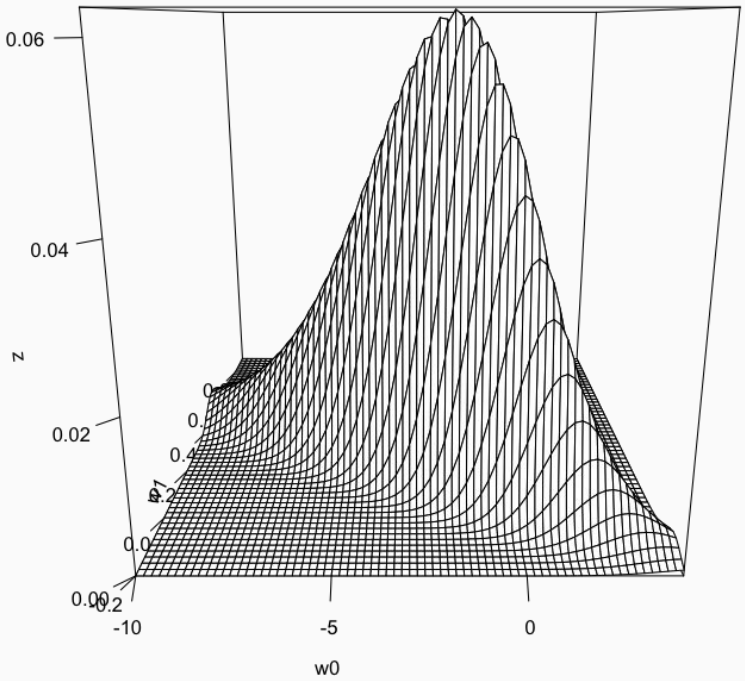
- Temos 5 crianças com idades x iguais a 5, 12, 22, 25, 30
- A verossimilhança para $w_0 = -6.3$ e $w_1 = 0.35$ é portanto igual ao produto

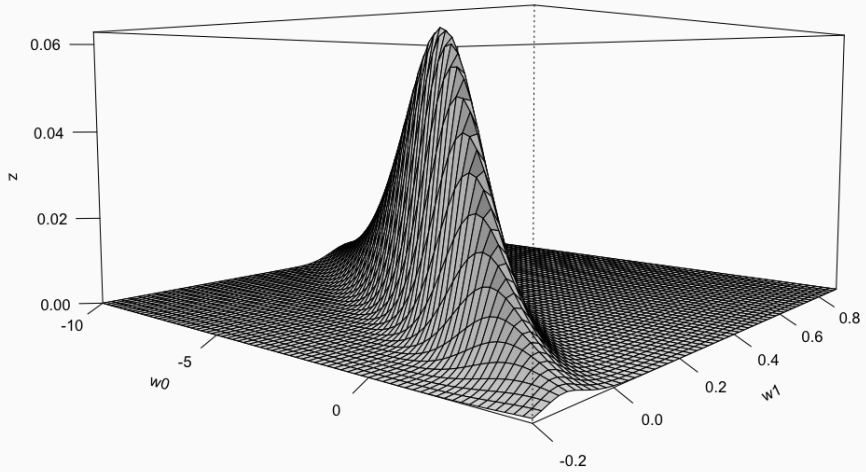
$$
\begin{aligned}
L(-6.3, 0.35) &= (1 - \sigma(5))\, \sigma(12)\, (1 - \sigma(22))\, \sigma(25)\, \sigma(30) \\
&= \frac{1}{1 + e^{(-6.3 + 0.35 * 5)}} \frac{1}{1 + e^{-(-6.3 + 0.35 * 12)}} \frac{1}{1 + e^{(-6.3 + 0.35 * 22)}} \frac{1}{1 + e^{-(-6.3 + 0.35 * 25)}} \frac{1}{1 + e^{-(-6.3 + 0.35 * 30)}} \\
&= 0.01936855
\end{aligned}
$$

- Escrevendo esta expressão como função genérica dos coeficientes w_0 e w_1 temos a função de verossimilhança

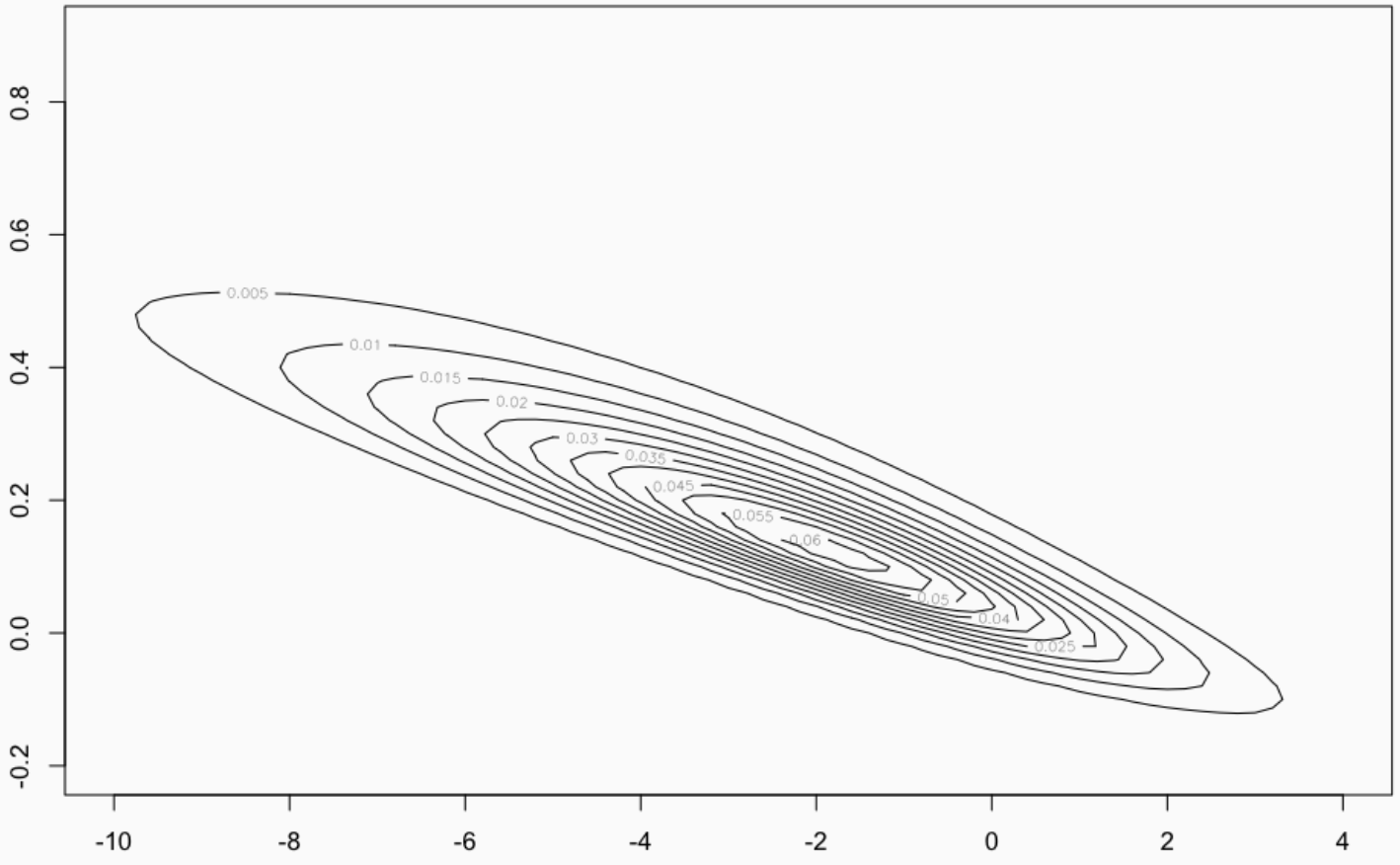
$$
\begin{aligned}
L(w_0, w_1) &= (1 - \sigma(5))\, \sigma(12)\, (1 - \sigma(22))\, \sigma(25)\, \sigma(30) \\
&= \frac{1}{1 + e^{(w_0 + w_1 * 5)}} \frac{1}{1 + e^{-(w_0 + w_1 * 12)}} \frac{1}{1 + e^{(w_0 + w_1 * 22)}} \frac{1}{1 + e^{-(w_0 + w_1 * 25)}} \frac{1}{1 + e^{-(w_0 + w_1 * 30)}}
\end{aligned}
$$

Função de verossimilhança $L(w_0, w_1)$





Curvas de nível da função de verossimilhança $L(w_0, w_1)$



MLE = Maximum Likelihood Estimator

- O MLE é o valor dos coeficientes (w_0, w_1) que maximiza a função de verossimilhança $L(w_0, w_1)$

- Notação: $$(\hat{w}_0, \hat{w}_1) = \arg\max_{(w_0, w_1)} L(w_0, w_1)$$

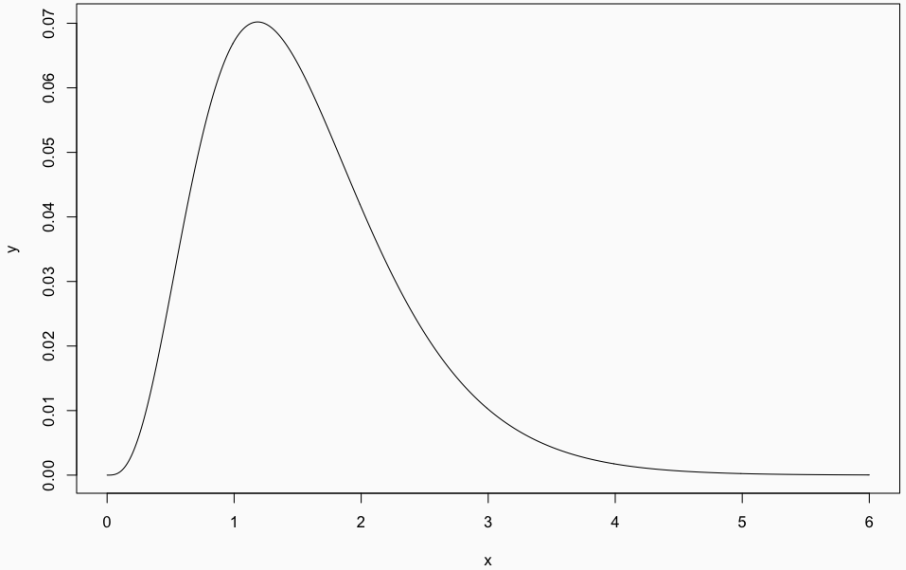
- Assim, $(\hat{w}_0, \hat{w}_1)$ é o valor dos coeficientes que torna máxima a probabilidade de observar a sequência de dados que realmente observamos
- Pelas curvas de nível do exemplo, vemos que $(\hat{w}_0, \hat{w}_1) \approx (-2, 0.1)$

Obtendo o MLE

- Precisamos de um algoritmo numérico para maximizar $L(w_0, w_1)$
- Método eficiente: método de Newton (ou Newton-Raphson)
- Como funciona?
- Caso uni-dimensional primeiro
- Queremos encontrar o ponto x* tal que f(x*) é o máximo da função f(x)
- Dizemos que x* é o ponto de máximo da função f(x): x* = arg max f(x)
- Como encontrar x*?
    - Derive f(x) obtendo f'(x)
    - Iguale a zero e "resolva" para x $\rightarrow$ f'(x) = 0 (encontrar a RAIZ desta equação)

Exemplo

- Queremos encontrar o ponto de máximo de $f(x) = x^{3.2} e^{-2.7x}$ para x > 0



Exemplo

- Obtemos a derivada f'(x)

$$f'(x) = 3.2\, x^{2.2} e^{-2.7x} - 2.7\, x^{3.2} e^{-2.7x}$$

- Iguale f'(x) = 0 e tente "isolar" x. Neste caso, é fácil:

$$\begin{aligned} 3.2\, x^{2.2} e^{-2.7x} &= 2.7\, x^{3.2} e^{-2.7x} \\ 3.2\, x^{2.2} &= 2.7\, x^{3.2} \\ \frac{3.2}{2.7} &= \frac{x^{3.2}}{x^{2.7}} \\ 1.185185 &= x^{0.5} \\ 1.404664 &= x \end{aligned}$$

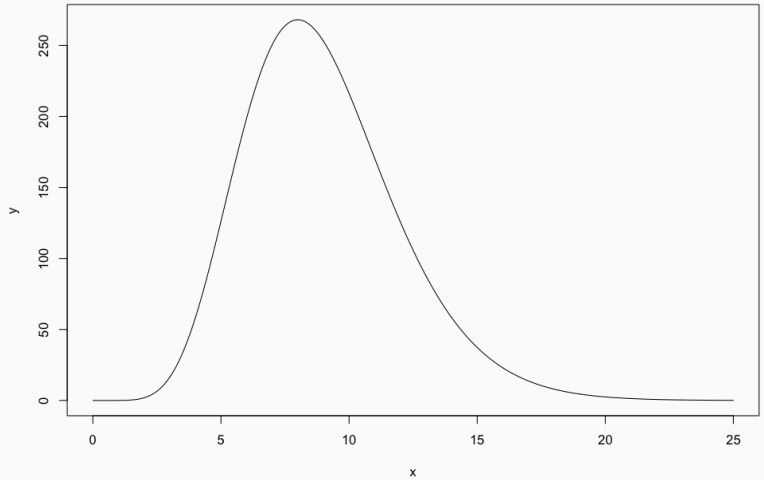
Exemplo

- Na maioria das vezes não conseguiremos isolar x:

$$f(x) = \frac{x^8}{21(e^x - 1)^{11}}$$

- com derivada

$$f'(x) = \frac{8\,x^7}{21(e^x - 1)^{11}} - \frac{231\,x^8(e^x - 1)e^x}{441\,(e^x - 1)^{22}}$$



- Não tem "isolar" x para obter o ponto de máximo

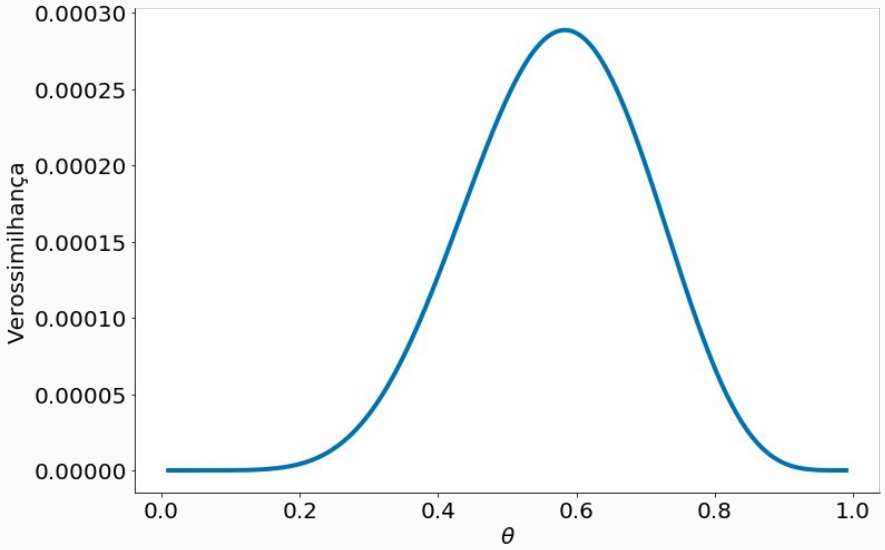
Um primeiro passo: tomar log(f(x))

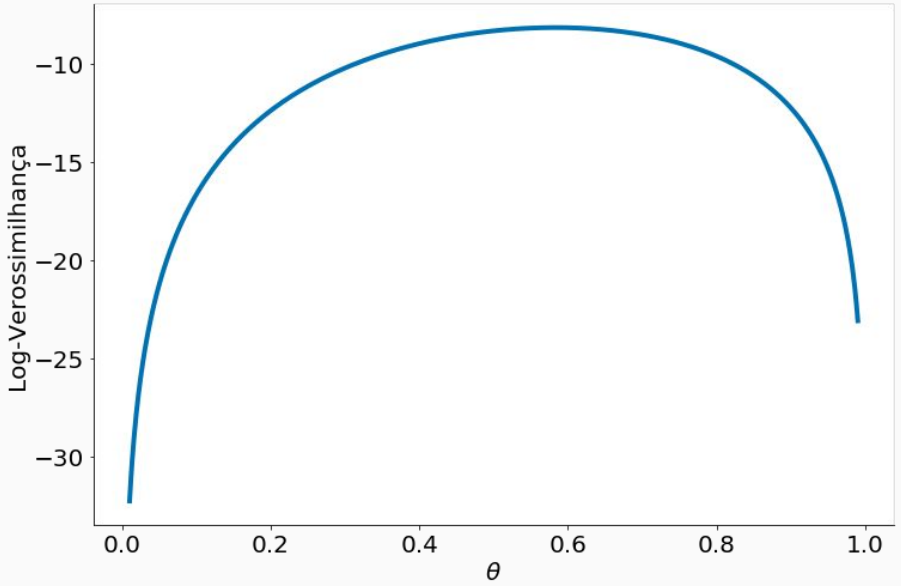
- Likelihood = probabilidade de vários dados
- Usualmente (quase sempre) ela será um PRODUTO de várias funções
- Considere o que é mais fácil derivar:
  - $f(x) = h(x) * g(x) * k(x)$
  - $f(x) = h(x) + g(x) + k(x)$
- Derivada de produtos será uma longa expressão:
  - $f'(x) = h'(x)* g(x) * k(x) + h(x) * g'(x) * k(x) + h(x) * g(x) * k'(x)$
  - $f'(x) = h'(x) + g'(x) + k'(x)$

Primeiro passo: tomar log

- Log(h(x) * g(x) * k(x)) = Log(h(x)) + Log(g(x)) + Log(k(x)) ← derivada + simples
- Mas faz sentido?? Queremos max $L(w_0, w_1)$ mas obtemos max $\log(L(w_0, w_1))$
- Na verdade, <u>não queremos</u> max $L(w_0, w_1)$
- Queremos ... arg max $L(w_0, w_1)$
- E .... arg max $L(w_0, w_1)$ = arg max $\log(L(w_0, w_1))$
- Por quê?
  - Porque log é função monótona: se x < y então log(x) < log(y)
  - Assim, se f(x) < f(x*) para todo x != x* então log(f(x)) < log(f(x*))
    - Se x* maximiza f(x) então x* também maximiza log(f(x))

Exemplos


0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
Verossimilhança
0.0
0.2
0.4
0.6
0.8
1.0
θ

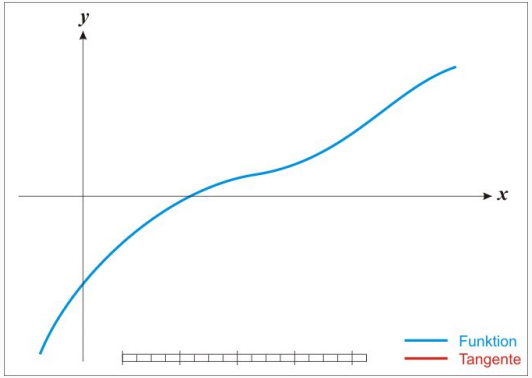

−10
−15
−20
−25
−30
Log-Verossimilhança
0.0
0.2
0.4
0.6
0.8
1.0
θ

Em suma, tome logs

- Em conclusão, $(\hat{w}_0, \hat{w}_1) = \underset{(w_0,w_1)}{\arg\max}\, L(w_0, w_1) = \underset{(w_0,w_1)}{\arg\max}\, \log(L(w_0, w_1))$
- Uma vantagem adicional: estabilidade numérica.
  - probabilidades estão 0 e 1.
  - Multiplicar muitas probabilidades → underflow (< precisão da máquina)
  - Tomar logs diminui substancialmente este problema.
- Em suma, vamos calcular o MLE buscando o máximo do LOG da função de verossimilhança
- Como fazer isto numericamente?

Achar o máximo de g(x) = achar raiz de g’(x)

- Achar o máximo de g(x) → pontos onde g'(x) = 0
- Chame g'(x) = f(x)
- Queremos achar as raízes da equação f(x) = 0
- Explicação intuitiva: como Newton deve ter pensado??
- Animação: https://en.wikipedia.org/wiki/Newton%27s_method
- Valor inicial: x_0



- Iterar até convergir: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$
- Regras de parada: $|x_{n+1} - x_n| < \varepsilon$ ou $\frac{|x_{n+1} - x_n|}{|x_n|} < \varepsilon$

$f(x) = 3x + 2 = 0$

$x\_1 = 2.3$ <- chute inicial

Como é a função f(x) em torno de $x\_1 = 2.3$?

Aproximar pela reta tangente

reta tangente que passa pela curva e em $x\_1 = 2.3$

$= f'(2.3)\, x + b = 0 \quad \rightarrow x = -f'(2.3)/b$

$3x = -2$

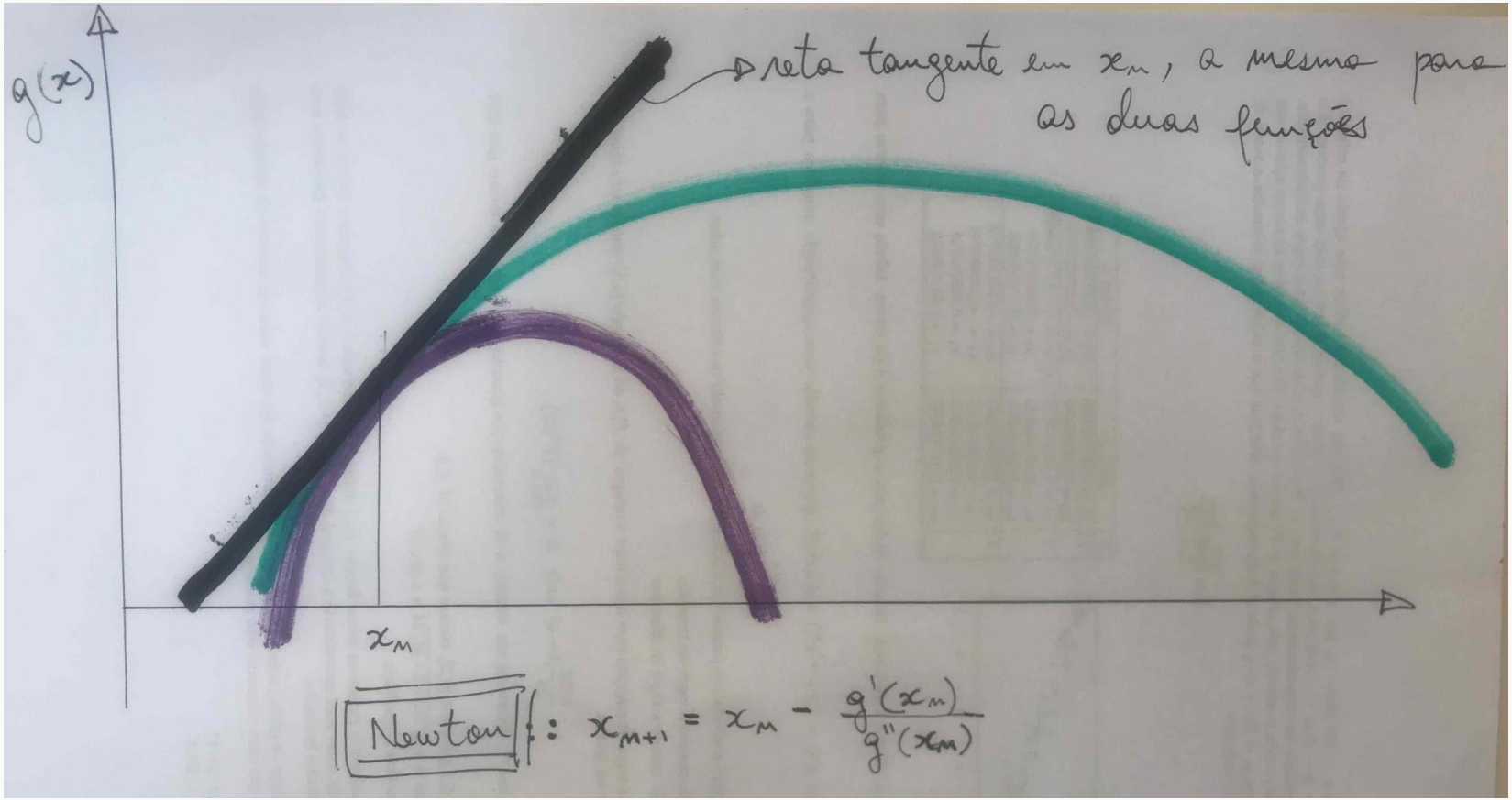
$x = -2/3$

Achar o máximo de g(x) = achar raiz de f(x)

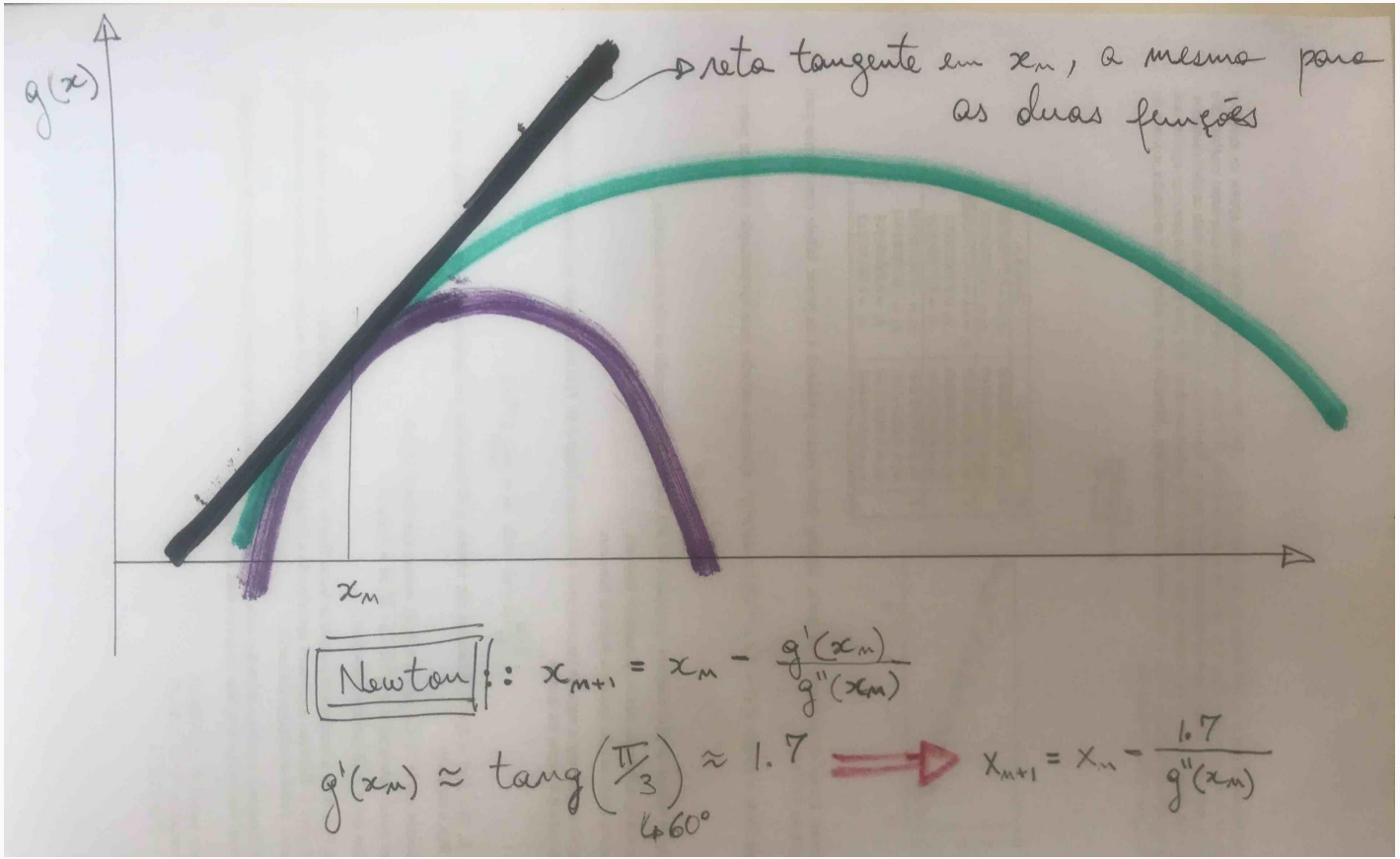
- Como g'(x) = f(x) ...
- a regra de iteração $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$
- significa $x_{n+1} = x_n - \frac{g'(x_n)}{g''(x_n)}$

- Vamos ver intuitivamente, o papel de cada termo na fórmula acima:
  - estando em x_n, para que lado andar? para a direita ou para a esquerda?
  - Decidindo para que lado andar, quanto devemos andar?
    - resposta depende de g'
    - e depende também de g"

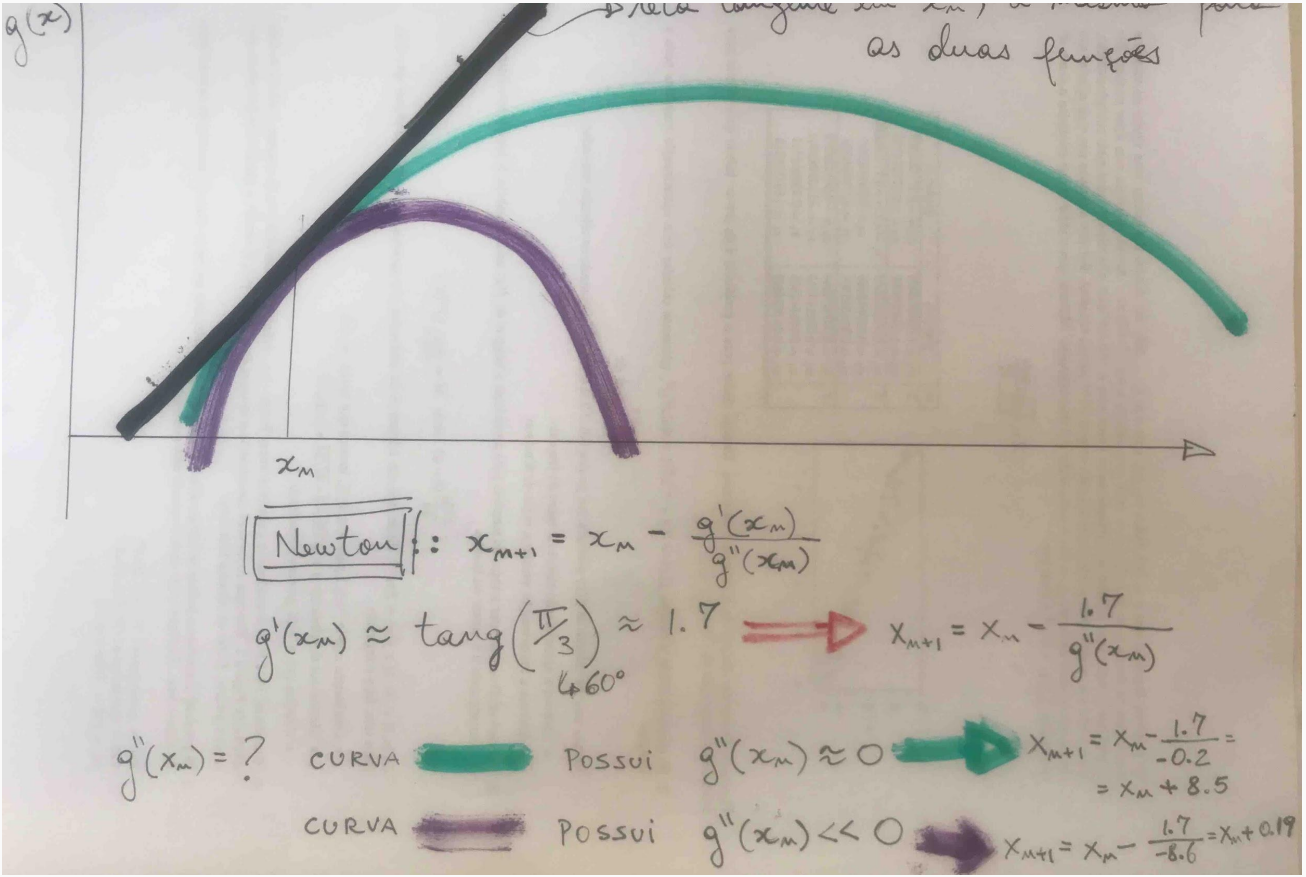
Explicação intuitiva do método de Newton



Explicação intuitiva do método de Newton

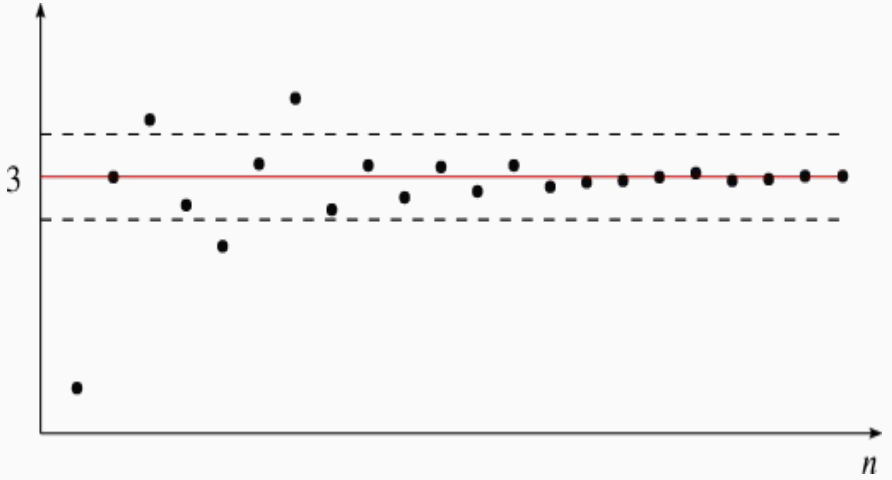


Explicação intuitiva do método de Newton



Convergência de método de Newton

- Grosseiramente, quando converge, o faz rapidamente
- Mas … nem sempre converge
- Existem algumas condições que garantem convergência mas elas em geral não são válida em DL



Generalizando para n features

- Queremos achar o máximo de uma função com mais de uma variável.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n \quad \text{um vetor-coluna } n \times 1$$

- Temos uma função

$$g : \mathbb{R}^n \mapsto \mathbb{R}$$
$$\mathbf{x} \to g(\mathbf{x})$$

Exemplos

$$g(w_0, w_1, w_2) = (w_0^2 + w_1^2 + w_2^2 - 2w_1 w_2) e^{-3w_0^2 + w_1^2 - 2w_2^2 + 0.4 w_0 w_1 w_2}$$

$$g(w_0, w_1, w_2) = \log(w_0^2 + w_1^2 + w_2^2 - 2w_1 w_2) - 3w_0^2 + w_1^2 - 2w_2^2 + 0.4 w_0 w_1 w_2$$

$$g(w_0, w_1, \ldots, w_n) = \frac{1}{1 + e^{-(w_0 + 3.27 w_1 + \ldots - 5.91 x_n)}}$$

$$g(w_0, w_1, \ldots, w_n) = \log\left(\frac{1}{1 + e^{-(w_0 + 3.27 w_1 + \ldots - 5.91 x_n)}}\right) + \log\left(\frac{1}{1 + e^{-(w_0 - 1.29 w_1 + \ldots + 0.22 x_n)}}\right) + \log\left(1 - \frac{1}{1 + e^{-(w_0 - 2.01 w_1 + \ldots + 0.73 x_n)}}\right)$$

Como achar o ponto de máximo da verossimilhança L(w)?

$$\hat{\mathbf{w}} = \begin{bmatrix} \hat{w}_0 \\ \hat{w}_1 \\ \vdots \\ \hat{w}_n \end{bmatrix} = \arg\max_{\mathbf{w}} L(\mathbf{w})$$

- Equação de iteração de Newton uni-dimensional:

$$w^{k+1} = w^k - \frac{L'(w^k)}{L''(w^k)} = w^k - \left[L''(w^k)\right]^{-1} L'(w^k)$$

- Caso multivariado: a mesma coisa, apenas matricial

Como achar o ponto de máximo da verossimilhança L(w)?

- Equação de iteração de Newton uni-dimensional:

$$w^{k+1} = w^k - \frac{L'(w^k)}{L''(w^k)} = w^k - \left[L''(w^k)\right]^{-1} L'(w^k)$$

- Caso multivariado: a mesma coisa, apenas matricial

$$\mathbf{w}^{k+1} = \begin{bmatrix} w_0^{k+1} \\ w_1^{k+1} \\ \vdots \\ w_n^{k+1} \end{bmatrix} = \begin{bmatrix} w_0^{k} \\ w_1^{k} \\ \vdots \\ w_n^{k} \end{bmatrix} - \Big[\underbrace{H(\mathbf{w}^k)}_{\text{matriz derivadas parciais de 2a ordem}} \Big]^{-1} \underbrace{\nabla L(\mathbf{w}^k)}_{\text{vetor de derivadas parciais}}$$

- Para atualizar w_j usamos ***TODAS*** as derivadas parciais (com respeito a todos os w_p, a menos que H seja matriz diagonal), em contraste com métodos de gradiente

Relembre o modelo de regressão logística

- Dados: pares de vetores (x_i, y_i)
- x_i = idade da criança i
- y_i = 0 ou 1
- Cada criança joga uma moeda para determinar seu sucesso ou fracasso (Y_i)
- A probabilidade de sucesso da criança i depende de sua idade x_i

$$P(Y_i = 1|x_i) = p(x_i) = \frac{1}{1+\exp(-(w_0+w_1 x_i))}$$

- Resultados das crianças são independentes: produto das probabilidades individuais
- Qual a probabilidade de vermos os dados que temos?

A função de log-verossimilhança

- Com m crianças:

$$
\begin{aligned}
L(\mathbf{w}) &= L(w_0, w_1) \\
&= \mathbb{P}(Y_1 = 0, Y_2 = 1, \ldots, Y_m = 1) \\
&= \mathbb{P}(Y_1 = 0)\, \mathbb{P}(Y_2 = 1)\, \ldots\, \mathbb{P}(Y_m = 1) \\
&= \prod_{i=1}^{m} \mathbb{P}(Y_i = y_i)
\end{aligned}
$$

- onde cada y_i (minúsculo) é igual a 0 ou 1
- Temos

$$
\mathbb{P}(Y_i = y_i) = \begin{cases} \sigma(x_i) & \text{se } y_i = 1 \\ 1 - \sigma(x_i) & \text{se } y_i = 0 \end{cases}
$$

- e $\sigma(x_i) = \frac{1}{1+e^{-(w_0 + w_1 x_i)}}$

Um truque importante

- Vimos que $\mathbb{P}(Y_i = y_i) = \begin{cases} \sigma(x_i) & \text{se } y_i = 1 \\ 1 - \sigma(x_i) & \text{se } y_i = 0 \end{cases}$
- Podemos escrever esta expressão usando uma única linha:

$$\mathbb{P}(Y_i = y_i) = \sigma(x_i)^{y_i} (1 - \sigma(x_i))^{1-y_i}$$

- Você vai verificar isto na aula de exercícios
- Qual a vantagem? Tome log:
- $\log(\mathbb{P}(Y_i = y_i)) = y_i \ \log(\sigma(x_i)) + (1 - y_i) \ \log(1 - \sigma(x_i))$

Log-verossimilhança

- Voltando para a amostra com os m indivíduos, obter a LOG-verossimilhança:

$$
\begin{aligned}
\ell(\mathbf{w}) &= \log(\ L(w_0, w_1)\ ) \\
&= \log\left(\prod_{i=1}^{m} \mathbb{P}(Y_i = y_i)\right) \\
&= \log\left(\prod_{i=1}^{m} \sigma(x_i)^{y_i}\,(1-\sigma(x_i))^{1-y_i}\right) \\
&= \sum_{i=1}^{m} \log\left(\sigma(x_i)^{y_i}\,(1-\sigma(x_i))^{1-y_i}\right) \\
&= \sum_{i=1}^{m} \left(y_i\ \log(\sigma(x_i)) + (1-y_i)\ \log(1-\sigma(x_i))\right)
\end{aligned}
$$

- sendo que $\sigma(x_i) = \frac{1}{1+e^{-(w_0+w_1 x_i)}}$

Equação de Newton: gradiente

- Precisamos das derivadas parciais com relação a w_0 e w_1
- Com $$\sigma(x_i) = \sigma_i = 1/(1 + e^{-(w_0 + w_1 x_i)})$$
- temos

$$\nabla \ell(\mathbf{w}) = \begin{bmatrix} \frac{\partial \log L}{\partial w_0} \\ \frac{\partial \log L}{\partial w_1} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} (y_i - \sigma_i) \\ \sum_{i=1}^{m} (y_i - \sigma_i) x_i) \end{bmatrix} = m \begin{bmatrix} \bar{y} - \bar{\sigma} \\ \overline{xy} - \overline{\sigma x} \end{bmatrix} = m \begin{bmatrix} \overline{y - \sigma} \\ \overline{(y - \sigma) x} \end{bmatrix}$$

- onde os $\sigma(x_i) = \sigma_i$ são avaliados (calculados) com o valor corrente dos pesos e são médias aritméticas $\bar{y}$, $\bar{\sigma}$, $\overline{xy}$ e $\overline{\sigma x}$

Dedução passo a passo do gradiente

$$
\begin{aligned}
\frac{\partial \log L}{\partial w_0} &= \frac{\partial}{\partial w_0}\left[\sum_{i=1}^{m} y_i \log(\sigma(x_i)) + (1-y_i)\log(1-\sigma(x_i))\right] \\
&= \sum_{i=1}^{m}\left[y_i \frac{\partial \log(\sigma(x_i))}{\partial w_0} + (1-y_i)\frac{\partial \log(1-\sigma(x_i))}{\partial w_0}\right] \\
&= \sum_{i=1}^{m}\left[y_i \frac{1}{\sigma(x_i)}\frac{\partial \sigma(x_i)}{\partial w_0} + (1-y_i)\frac{1}{1-\sigma(x_i)}\frac{\partial(-\sigma(x_i))}{\partial w_0}\right] \\
&= \sum_{i=1}^{m}\left[\frac{y_i}{\sigma(x_i)} - \frac{1-y_i}{1-\sigma(x_i)}\right]\left[\frac{\partial \sigma(x_i)}{\partial w_0}\right]
\end{aligned}
$$

Dedução passo a passo do gradiente

$$
\begin{aligned}
\frac{\partial \sigma(x_i)}{\partial w_0} &= \frac{\partial}{\partial w_0} \frac{1}{1 + e^{-(w_0 + w_1 x_i)}} \\
&= \frac{\partial}{\partial w_0} \left(1 + e^{-(w_0 + w_1 x_i)}\right)^{-1} \\
&= (-1)\left(1 + e^{-(w_0 + w_1 x_i)}\right)^{-2} \frac{\partial e^{-(w_0 + w_1 x_i)}}{\partial w_0} \\
&= \frac{-1}{(1 + e^{-(w_0 + w_1 x_i)})^2} e^{-(w_0 + w_1 x_i)} \frac{\partial (-(w_0 + w_1 x_i))}{\partial w_0} \\
&= \frac{-1}{(1 + e^{-(w_0 + w_1 x_i)})^2} e^{-(w_0 + w_1 x_i)} (-1) \\
&= \frac{1}{1 + e^{-(w_0 + w_1 x_i)}} \frac{e^{-(w_0 + w_1 x_i)}}{1 + e^{-(w_0 + w_1 x_i)}} \\
&= \sigma(x_i)(1 - \sigma(x_i))
\end{aligned}
$$

Dedução passo a passo do gradiente

$$\frac{\partial \log L}{\partial w_0} = \sum_{i=1}^{m} \left[\frac{y_i}{\sigma(x_i)} - \frac{1 - y_i}{1 - \sigma(x_i)} \right] \left[\sigma(x_i)(1 - \sigma(x_i)) \right]$$

$$= \sum_{i=1}^{m} \left[y_i (1 - \sigma(x_i)) - (1 - y_i)\sigma(x_i) \right]$$

$$= \sum_{i=1}^{m} \left[y_i - y_i \sigma(x_i)) - \sigma(x_i) + y_i \sigma(x_i) \right]$$

$$= \sum_{i=1}^{m} \left[y_i - \sigma(x_i) \right]$$

Dedução passo a passo do gradiente

$$\frac{\partial \log L}{\partial w_1} = \sum_{i=1}^{m} \left[\frac{y_i}{\sigma(x_i)} - \frac{1-y_i}{1-\sigma(x_i)} \right] \left[\frac{\partial \sigma(x_i)}{\partial w_1} \right]$$

$$\begin{aligned}
\frac{\partial \sigma(x_i)}{\partial w_1} &= \frac{\partial}{\partial w_0} \frac{1}{1+e^{-(w_0+w_1 x_i)}} \\
&= \frac{-1}{\left(1+e^{-(w_0+w_1 x_i)}\right)^{-2}} e^{-(w_0+w_1 x_i)} \frac{\partial(-(w_0+w_1 x_i))}{\partial w_1} \\
&= -\sigma(x_i)(1-\sigma(x_i))\,(-x_i) \\
&= \sigma(x_i)(1-\sigma(x_i))\,x_i
\end{aligned}$$

Dedução passo a passo do gradiente

$$\frac{\partial \log L}{\partial w_1} = \sum_{i=1}^{m} \left[\frac{y_i}{\sigma(x_i)} - \frac{1-y_i}{1-\sigma(x_i)} \right] \left[\frac{\partial \sigma(x_i)}{\partial w_1} \right]$$

$$= \sum_{i=1}^{m} \left[\frac{y_i}{\sigma(x_i)} - \frac{1-y_i}{1-\sigma(x_i)} \right] \left[\sigma(x_i)(1-\sigma(x_i))\, x_i \right]$$

$$= \sum_{i=1}^{m} \left[y_i - \sigma(x_i) \right] \left(x_i \right)$$

Vetorizando o gradiente

$$\frac{\partial \ell}{\partial \mathbf{w}} = \underbrace{\frac{\partial \log L}{\partial \mathbf{w}}}_{2\times 1}$$

$$= \begin{bmatrix} \partial \ell / \partial w_0 \\ \partial \ell / \partial w_1 \end{bmatrix} = \begin{bmatrix} \sum_i (y_i - \sigma(x_i))\, 1 \\ \sum_i (y_i - \sigma(x_i))\, x_i \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_m \end{bmatrix} \begin{bmatrix} y_1 - \sigma(x_1) \\ y_2 - \sigma(x_2) \\ \vdots \\ y_m - \sigma(x_m) \end{bmatrix}$$

Equação de Newton: Hessiano

$$\frac{\partial^2 \ell}{\partial w_0^2} = \frac{\partial}{\partial w_0}\frac{\partial \ell}{\partial w_0}$$

$$= \frac{\partial}{\partial w_0}\left(\sum_i [y_i - \sigma(x_i)]\right)$$

$$= \sum_i \frac{\partial}{\partial w_0}[y_i - \sigma(x_i)] = -\sum_i \frac{\partial \sigma(x_i)}{\partial w_0}$$

$$= -\sum_i \sigma(x_i)(1-\sigma(x_i)) = -n\,\frac{1}{n}\sum_i \sigma(x_i)(1-\sigma(x_i))$$

$$= -n\overline{\sigma(1-\sigma)}$$

Dedução passo a passo do Hessiano

- De modo similar, obtemos os demais elementos da matriz Hessiana.

$$H = -n \begin{bmatrix} \overline{\sigma(1-\sigma)} & \overline{\sigma(1-\sigma)x} \\ \overline{\sigma(1-\sigma)x} & \overline{\sigma(1-\sigma)x^2} \end{bmatrix}$$

- onde os elementos acima são médias aritméticas sobre os exemplos

Equação de iteração de Newton

- De volta ao procedimento de maximização:

$$\mathbf{w}^{k+1} = \begin{bmatrix} w_0^{k+1} \\ w_1^{k+1} \end{bmatrix} = \begin{bmatrix} w_0^{k} \\ w_1^{k} \end{bmatrix} - \Big[\underbrace{H(\mathbf{w}^k)}_{\text{matriz derivadas parciais de 2a ordem}} \Big]^{-1} \underbrace{\nabla L(\mathbf{w}^k)}_{\text{vetor de derivadas parciais}}$$

$$\mathbf{w}^{k+1} = \begin{bmatrix} w_0^{k+1} \\ w_1^{k+1} \end{bmatrix} = \begin{bmatrix} w_0^{k} \\ w_1^{k} \end{bmatrix} + \frac{1}{n} \begin{bmatrix} \overline{\sigma(1-\sigma)} & \overline{\sigma(1-\sigma)x} \\ \overline{\sigma(1-\sigma)x} & \overline{\sigma(1-\sigma)x^2} \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_m \end{bmatrix} \begin{bmatrix} y_1 - \sigma(x_1) \\ y_2 - \sigma(x_2) \\ \vdots \\ y_m - \sigma(x_m) \end{bmatrix}$$

- Para atualizar w_1, usamos a derivada parcial com respeito a w_1 E TAMBÉM w_0 (a menos que H seja matriz diagonal, e geralmente ela não é diagonal).

Flexibilidade da regressão logística

- Regressão logística é menos limitada do que parece.
- Os inputs-features podem ser:
  - Quaisquer características (features) dos dados
  - Transformações das features x originais tais como, por exemplo, log(x)
  - Uma expansão de base, por exemplo, x**2 e x**3
  - Indicadores de categorias (features categóricas)
  - Interações entre duas features tal como, por exemplo, x2 * x3
- A simplicidade e flexibilidade da regressão logística a tornam uma das técnicas de classificação estatística mais importantes e mais amplamente utilizada.

Regressão logística com várias features

- A chance de sucesso da criança não depende APENAS de sua idade.
- Vai depender também de:
  - sexo: feature X_2 = 0 (masc) ou 1 (fem)
  - escolaridade da mãe: feature X_3 = no. de anos de estudo formal
  - renda per capita da família: feature X_4 = renda mensal em 1000 reais
- Coletamos as features de cada criança num vetor **x** (em negrito):
  - $x = (x_1, x_2, x_3, x_4)$
- Como fazer um modelo em que a chance de sucesso depende de todas estas características simultaneamente?

- Modelo logístico incorpora todas as features de forma LINEAR.
- Para cada criança, crie um escore z:
  - Cada feature da criança é multiplicada por um peso w
  - O peso da feature está associado à importância da feature:
    - features importantes terão |w| grande
    - features pouco importantes terão seu peso |w| pequeno
    - features totalmente irrelevantes devem ter |w| aprox zero
  - Depois de ponderar cada feature da criança, somamos para obter o escore z

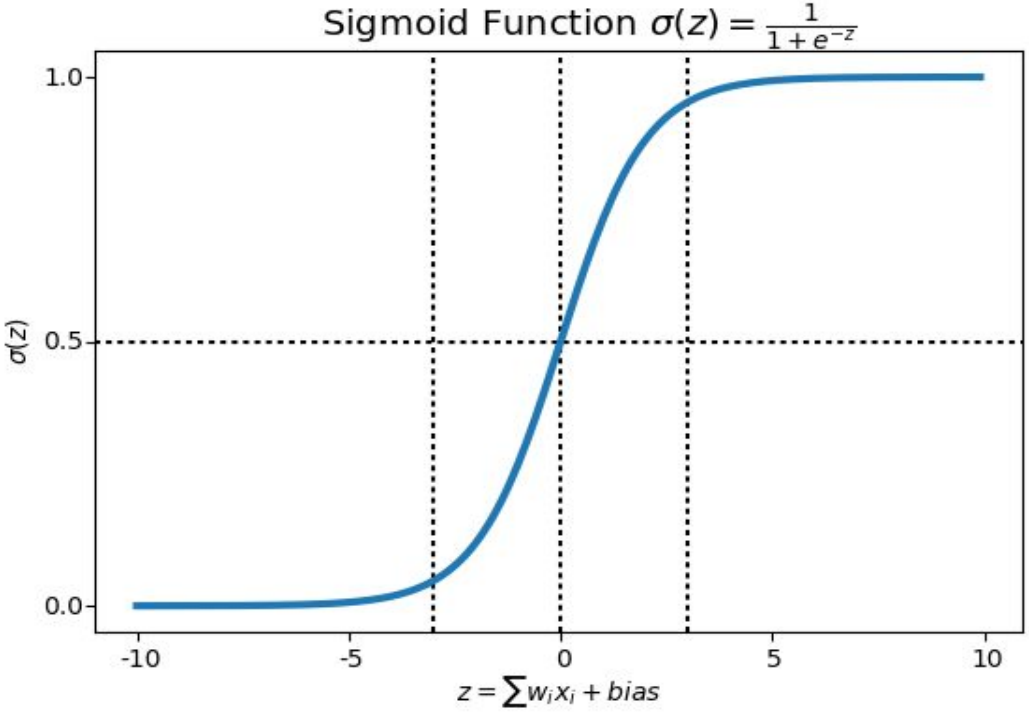
$$z = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4$$

Regressão logística com várias features

- Calcule $z = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4$
- para cada criança
- Queremos que a probabilidade de sucesso seja uma função do escore:
  - um alto valor de z leva a uma probabilidade alta (aprox 1)
  - um valor baixo de z leva a uma probabilidade baixa (aprox 0)
- Reduzimos a complexidade da análise a uma forma manejável, simples.
- O escore z embute a influência de todas as features ao mesmo tempo.

  $\mathbb{P}(Y = 1) = \sigma(z) = \frac{1}{1+e^{-z}}$
- Dois indivíduos com features diferentes *MAS COM O MESMO ESCORE z* terão a mesma probabilidade de sucesso.

Representação gráfica



$$\mathbb{P}(Y = 1) = \sigma(z) = \frac{1}{1+e^{-z}}$$

$$z = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4$$

Aprendizagem a partir dos dados

- Precisamos responder várias perguntas:
  - 1) Este modelo logístico representa bem os dados observados?
  - 2) Se ele representar bem os dados, como aprender os pesos "corretos" a partir de dados observados (= amostra de treinamento)
  - 3) Não queremos apenas aprender com os dados. Queremos a "melhor representação" possível. Qual a "melhor maneira" de aprender os pesos?
  - 4) Podemos fazer algo melhor que usar a regressão logística?
- Vamos responder (2) e (3) no resto dessa aula. Amanhã, veremos (1) e (4).

Olhando os escores de toda a amostra de treinamento

- Imagine que temos n=4 features e m crianças.
- Calculamos os escores z de todas elas numa única operação matricial:

$$\begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & x_{14} \\ 1 & x_{21} & x_{22} & x_{23} & x_{24} \\ & & \vdots & & \\ & & \vdots & & \\ 1 & x_{m1} & x_{m2} & x_{m3} & x_{m4} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} \mathbf{w}'\mathbf{x}^{(1)} \\ \mathbf{w}'\mathbf{x}^{(2)} \\ \vdots \\ \vdots \\ \mathbf{w}'\mathbf{x}^{(m)} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ \vdots \\ z_m \end{bmatrix}$$

- ***Um único*** vetor de pesos w é aplicado a cada uma das m crianças

Calcule agora as probabilidades de sucesso

- Depois de obter os z's obtenha as probabilidades:

$$\begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & x_{14} \\ 1 & x_{21} & x_{22} & x_{23} & x_{24} \\ & & \vdots & & \\ & & \vdots & & \\ 1 & x_{m1} & x_{m2} & x_{m3} & x_{m4} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} \mathbf{w}'\mathbf{x}^{(1)} \\ \mathbf{w}'\mathbf{x}^{(2)} \\ \vdots \\ \vdots \\ \mathbf{w}'\mathbf{x}^{(m)} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ \vdots \\ z_m \end{bmatrix} \rightarrow \begin{bmatrix} \sigma(z_1) \\ \sigma(z_2) \\ \vdots \\ \vdots \\ \sigma(z_m) \end{bmatrix} = \begin{bmatrix} 1/(1+e^{-z_1}) \\ 1/(1+e^{-z_2}) \\ \vdots \\ \vdots \\ 1/(1+e^{-z_m}) \end{bmatrix}$$

- Como obter os pesos w?
  - Do mesmo modo que antes: maximize a log-verossimilhança
  - Fórmulas são as mesmas de antes

Equação de iteração de Newton

- De volta ao procedimento de maximização:

$$\mathbf{w}^{k+1} = \begin{bmatrix} w_0^{k+1} \\ w_1^{k+1} \end{bmatrix} = \begin{bmatrix} w_0^{k} \\ w_1^{k} \end{bmatrix} - \left[\underbrace{H(\mathbf{w}^k)}_{\text{matriz derivadas parciais de 2a ordem}} \right]^{-1} \underbrace{\nabla L(\mathbf{w}^k)}_{\text{vetor de derivadas parciais}}$$

$$\mathbf{w}^{k+1} = \begin{bmatrix} w_0^{k+1} \\ w_1^{k+1} \end{bmatrix} = \begin{bmatrix} w_0^{k} \\ w_1^{k} \end{bmatrix} + \frac{1}{m} \begin{bmatrix} \overline{\sigma(1-\sigma)} & \overline{\sigma(1-\sigma)x} \\ \overline{\sigma(1-\sigma)x} & \overline{\sigma(1-\sigma)x^2} \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_m \end{bmatrix} \begin{bmatrix} y_1 - \sigma(x_1) \\ y_2 - \sigma(x_2) \\ \vdots \\ y_m - \sigma(x_m) \end{bmatrix}$$

- Para atualizar w_1, usamos a derivada parcial com respeito a w_1 E TAMBÉM w_0 (a menos que H seja matriz diagonal, e geralmente ela não é diagonal).

Log-verossimilhança

$$
\begin{aligned}
\ell(\mathbf{w}) &= \log(\ L(w_0, w_1, w_2, w_3, w_4)\ ) \\
&= \log\left(\prod_{i=1}^{m} \mathbb{P}(Y_i = y_i)\right) \\
&= \log\left(\prod_{i=1}^{m} \sigma(\mathbf{x}^{(i)})^{y_i} (1 - \sigma(\mathbf{x}^{(i)}))^{1-y_i}\right) \\
&= \sum_{i=1}^{m} \log\left(\sigma(\mathbf{x}^{(i)})^{y_i} (1 - \sigma(\mathbf{x}^{(i)}))^{1-y_i}\right) \\
&= \sum_{i=1}^{m} \left(y_i\ \log(\sigma(\mathbf{x}^{(i)})) + (1 - y_i)\ \log(1 - \sigma(\mathbf{x}^{(i)}))\right)
\end{aligned}
$$

- sendo que $\sigma(\mathbf{x}^{(i)}) = \frac{1}{1+e^{-z_i}} = \frac{1}{1+e^{-(w_0+w_1x_{i1}+w_2x_{i2}+w_3x_{i3}+w_4x_{i4})}}$

Equação de Newton: gradiente

- Precisamos das derivadas parciais com relação a cada componente de w
- Temos

$$\nabla \ell(\mathbf{w}) = \begin{bmatrix} \frac{\partial \log L}{\partial w_0} \\ \frac{\partial \log L}{\partial w_1} \\ \frac{\partial \log L}{\partial w_2} \\ \frac{\partial \log L}{\partial w_3} \\ \frac{\partial \log L}{\partial w_4} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n}(y_i - \sigma_i) \\ \sum_{i=1}^{n}(y_i - \sigma_i)x_{i1} \\ \sum_{i=1}^{n}(y_i - \sigma_i)x_{i2} \\ \sum_{i=1}^{n}(y_i - \sigma_i)x_{i3} \\ \sum_{i=1}^{n}(y_i - \sigma_i)x_{i4} \end{bmatrix} = n \begin{bmatrix} \overline{y - \sigma} \\ \overline{(y - \sigma)x_1} \\ \overline{(y - \sigma)x_2} \\ \overline{(y - \sigma)x_3} \\ \overline{(y - \sigma)x_4} \end{bmatrix}$$

Mais uma forma de expressar o gradiente

- Notação matricial

$$\nabla \ell(\mathbf{w}) = \begin{bmatrix} \frac{\partial \log L}{\partial w_0} \\ \frac{\partial \log L}{\partial w_1} \\ \frac{\partial \log L}{\partial w_2} \\ \frac{\partial \log L}{\partial w_3} \\ \frac{\partial \log L}{\partial w_4} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n}(y_i - \sigma_i) \\ \sum_{i=1}^{n}(y_i - \sigma_i)x_{i1} \\ \sum_{i=1}^{n}(y_i - \sigma_i)x_{i2} \\ \sum_{i=1}^{n}(y_i - \sigma_i)x_{i3} \\ \sum_{i=1}^{n}(y_i - \sigma_i)x_{i4} \end{bmatrix} = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \dots \quad \dots & \mathbf{x}^{(m)} \\ | & | & & | \end{bmatrix}}_{5 \times m} \underbrace{\begin{bmatrix} y_1 - \sigma_1 \\ y_2 - \sigma_2 \\ \vdots \\ y_m - \sigma_m \end{bmatrix}}_{m \times 1}$$

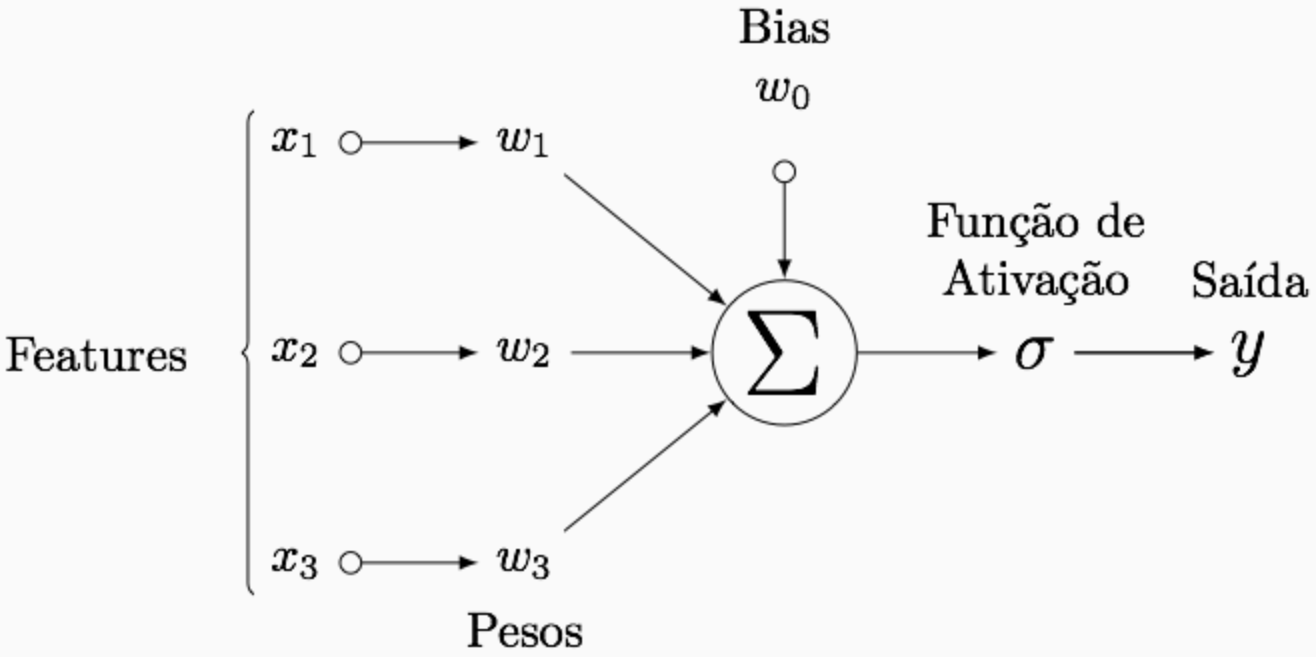
Equação de iteração de Newton

- De volta ao procedimento de maximização:

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \underbrace{\left[\underbrace{H(\mathbf{w}^k)}_{5 \text{ x } 5,\ 2\text{a ordem}} \right]^{-1}}_{} \underbrace{\nabla L(\mathbf{w}^k)}_{\text{vetor } 5 \text{ x } 1}$$

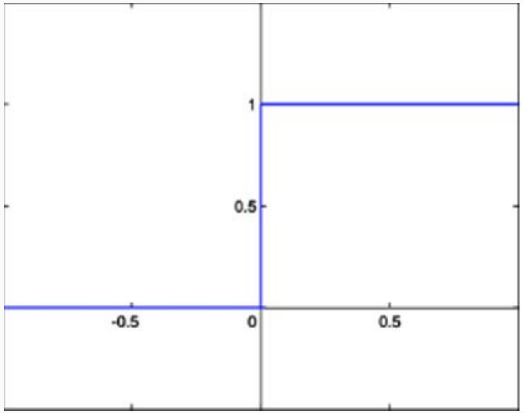
$$\mathbf{w}^{k+1} = \mathbf{w}^k + \frac{1}{m} \begin{bmatrix} \overline{\sigma(1-\sigma)} & \overline{\sigma(1-\sigma)x_1} & \dots & \overline{\sigma(1-\sigma)x_4} \\ \overline{\sigma(1-\sigma)x_1} & \overline{\sigma(1-\sigma)x_2^2} & \dots & \overline{\sigma(1-\sigma)x_2x_4} \\ & & \vdots & \\ \overline{\sigma(1-\sigma)x_4} & \overline{\sigma(1-\sigma)x_1x_4} & \dots & \overline{\sigma(1-\sigma)x_4^2} \end{bmatrix}^{-1} \begin{bmatrix} | & | & & | \\ \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \dots \dots & \mathbf{x}^{(m)} \\ | & | & & | \end{bmatrix} \begin{bmatrix} y_1 - \sigma_1 \\ y_2 - \sigma_2 \\ \vdots \\ y_m - \sigma_m \end{bmatrix}$$

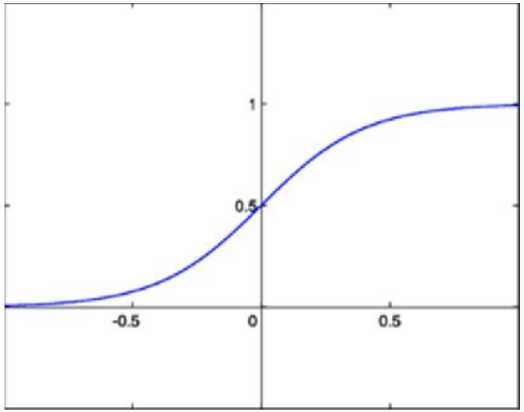
Regressão logística como rede neural com uma camada



Perceptron x logística

- Perceptron: threshold "hard": se $z = w_0 + w_1 x_1 + \ldots + w_n x_n > 0 \rightarrow \mathbb{P}(\text{class1}) = 1$
- Modelo logístico: threshold "soft": se $z = w_0 + w_1 x_1 + \ldots + w_n x_n > 0 \rightarrow \mathbb{P}(\text{class1}) > 1/2$
  - perceptron gera dados com classes linearmente separáveis
  - logística gera dados não-linearmente separáveis:
    - podemos ter $\mathbb{P}(\text{class1}) \approx 1$ mas ainda assim observar a classe 0

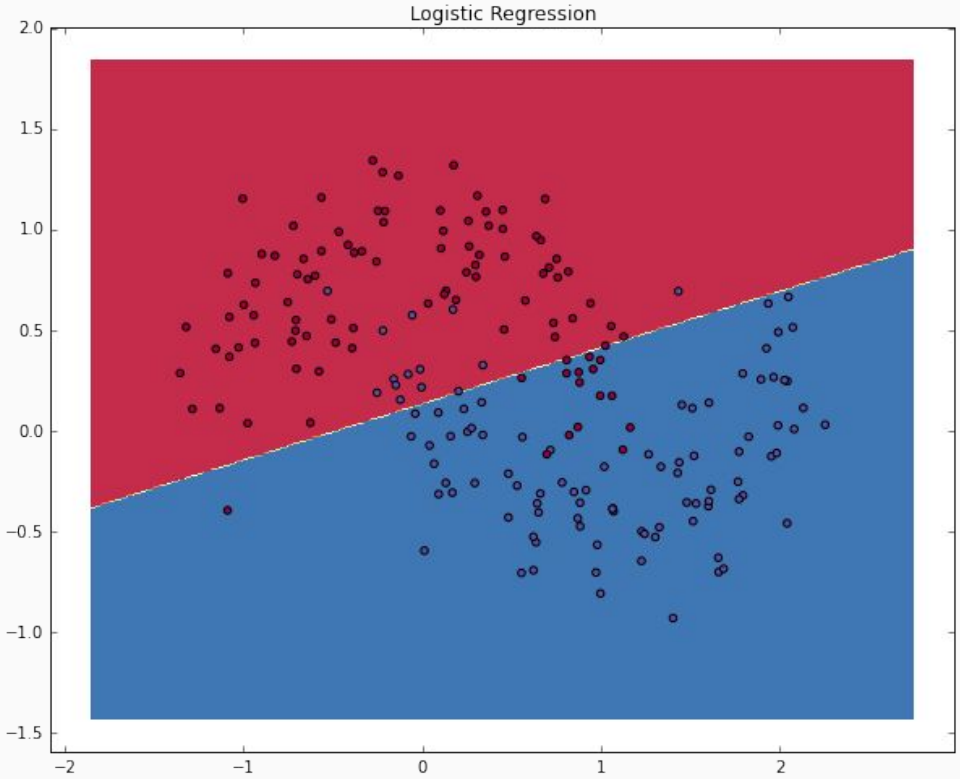




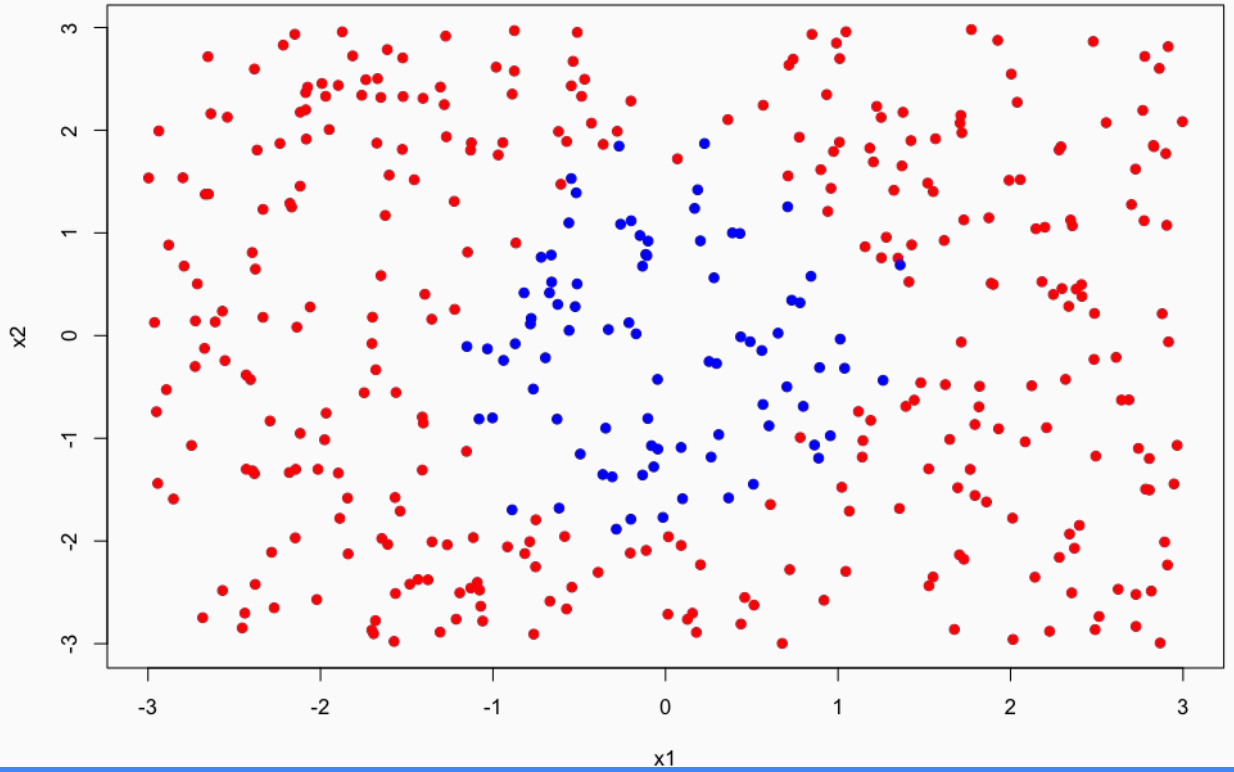
Usando a regressão logística para classificar

- Imagine que temos apenas duas features, x_1 e x_2
- Achamos os pesos w_0, w_1 e w_2 por máxima verossimilhança
- Temos então $\mathbb{P}(Y = 1|x_1, x_2) = \sigma(x_1, x_2) = \frac{1}{1+e^{w_0+w_1x_1+w_2x_2}}$
- Considere os pontos do plano (x_1, x_2) tais que esta probab = ½
- Quem são estes pontos? (Exercício)
- São os pontos tais que $w_0 + w_1x_1 + w_2x_2 = 0$
- Esta é a equação de uma reta no plano (x_1, x_2)
- Ela determina uma fronteira de decisão:
  - de um lado, probab de sucesso é > ½
  - do outro lado, é menor que ½

Decision boundary



E quando a real fronteira de decisão não for linear?

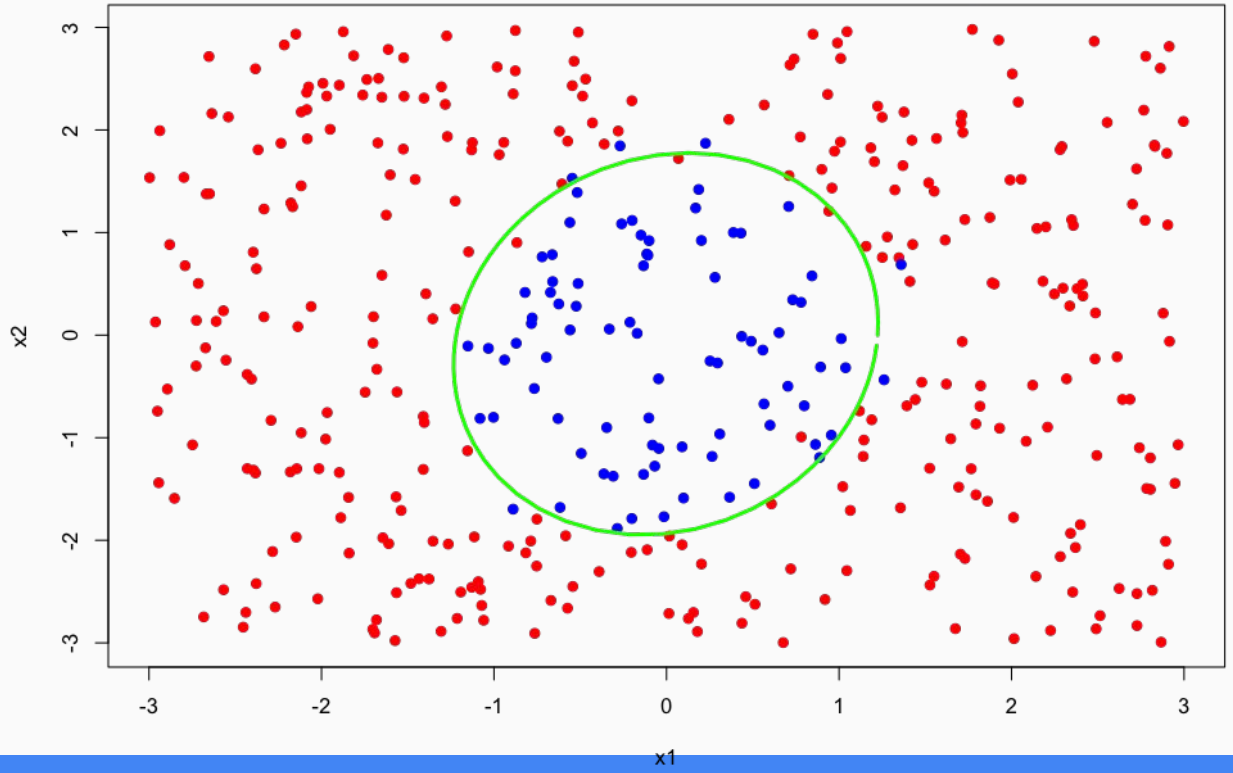


Modelo generativo usado e ajuste de regressão logística

$$z_i = 7 - 0.1x_{i1} - 0.15x_{i2} - 4.4x_{i1}^2 - 2.2x_{i2}^2 + 0.5x_{i1}x_{i2}$$

```
> summary(fit1)

Call:
glm(formula = y ~ matx, family = binomial("logit"))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.34601  -0.00377   0.00000   0.00000   2.48558

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  9.59718    1.92892   4.975 6.51e-07 ***
matxx1       0.03001    0.42974   0.070   0.9443
matxx2      -0.47726    0.26839  -1.778   0.0754 .
matx        -6.43045    1.29979  -4.947 7.52e-07 ***
matx        -2.80773    0.56631  -4.958 7.13e-07 ***
matx         0.93236    0.47525   1.962   0.0498 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 411.165  on 399  degrees of freedom
Residual deviance:  54.943  on 394  degrees of freedom
AIC: 66.943

Number of Fisher Scoring iterations: 11
```

Resultado do ajuste: fronteira de decisão

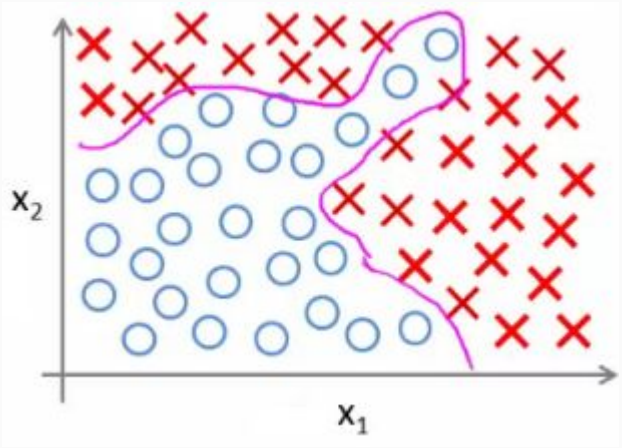


Flexibilidade da regressão logística

- Este exemplo mostra que a regressão logística possui grande flexibilidade
- Features podem ser criadas a partir de features básicas:
  - potências de features básicas: x2 = x1**2 (renda ao quadrado, ao cubo)
  - transformações não-lineares de features básicas: x2 = g(x1) (tal como log(renda) ou sqrt(renda))
  - termos de interações entre features: x3 = x1*x2 (tal como x3 = sexo*renda)
- A probabilidade de sucesso é uma função de uma COMBINAÇÃO LINEAR das features (básicas ou derivadas):
- $$z_i = w_0 + w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i1}^2 + w_4 x_{i1} x_{i2} + w_5 \log(x_{i1}) + w_6 x_{i2} \sqrt{x_{i3}}$$
  $$\mathbb{P}(Y_i = 1 | \mathbf{x}_i) = \frac{1}{1+e^{-z_i}}$$

Importante mensagem:

- Para aprender uma decision boundary não-linear com regressão logística → precisamos de muitos termos não lineares das features "básicas"
- Por exemplo, com duas features x1 e x2, podemos buscar os pesos w com

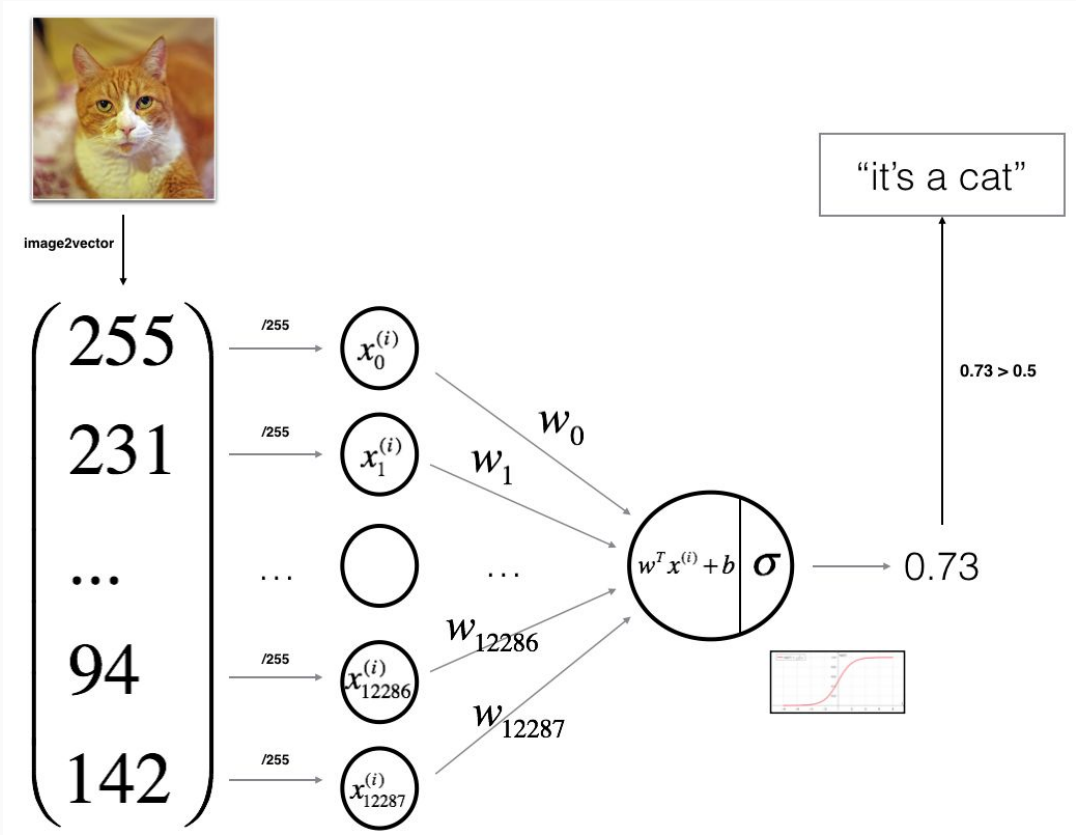


$$\mathbb{P}(Y=1|x_1,x_2)=\frac{1}{1+e^{-(w_0+w_1x_1+w_2x_2+w_3x_1^2+w_4x_2^2+w_5x_1x_2+w_6x_1^3+w_7x_1^2x_2+w_8x_1x_2^2)}}$$

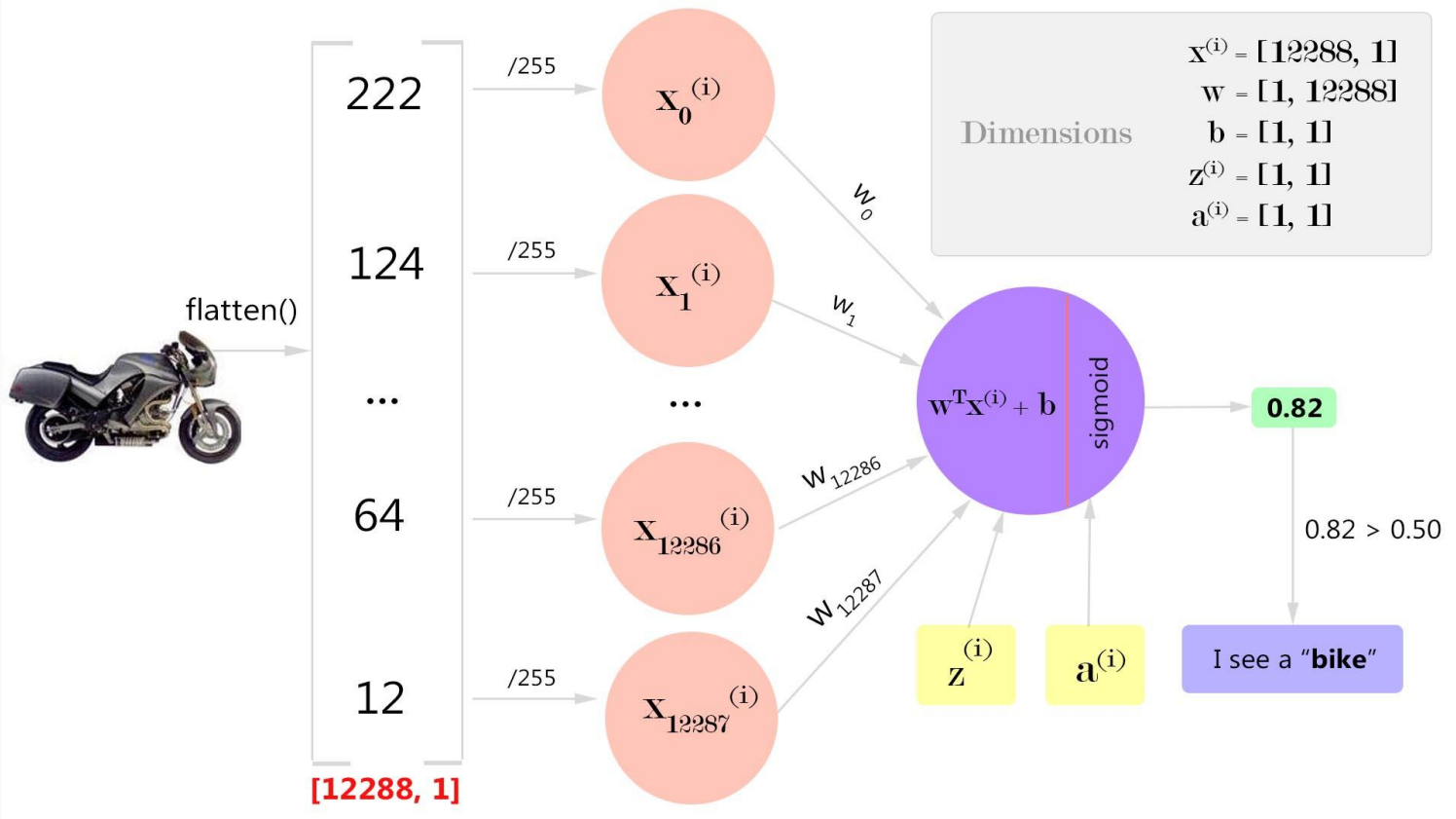
Regressão logística para imagem?

- Podemos usar regressão logística para classificar imagens em dois grupos.
- Por exemplo, gatos x não-gatos
- Provavelmente não teremos uma bom resultado
- Mas como isto pode ser feito, mesmo que gerando um resultado pobre em termos de acertos na classificação?
- Transformamos cada imagem num grande vetor de features.
- As features são as intensidades de "cores"nos pixels das imagens.
- Isto é,
    - cada pixel → uma feature.

Logística para imagem?



Logística para imagem?



Métricas para avaliar a regra de classificação

- A classificação feita pela nossa regra de decisão (baseda na regressão logística não é perfeita.
- Ela comete vários erros: indivíduoa que de fato são diabéticos não possuem as características x_1 e x_2 típicas de um diabético.
- Em consequência, a nossa regra de decisão (que olha apenas os regressores em $\boldsymbol{x}$) aloca estes indivíduos à classe 0 (não diabéticos).
- Estes são os *falso-negativos* (o diagnóstico é falsamente negativo).
- Analogamente, vários não-diabéticos possuem características típicas de diabéticos e são então alocados pela regra de decisão logística à categora 1 (diabéticos).
- Estes são os *falso-positivos* (o diagnóstico é falsamente positivo).
- Claro, existe o conceito de *verdadeiro-positivo* e *verdadeiro-negativo*.

Falso-positivos e Falso-negativos

- Idealmente, queremos poucos falso-positivos e poucos falso-negativos (ou muitos verdadeiro-positivos e muitos verdadeiro-negativos).
- Isto será obtido se tivermos uma pequena probabilidade de ter um falso-positivo (FP) e um falso-negativo (FN).

$$\mathbb{P}(FP) = \mathbb{P}(\text{classificado como } + | \text{é -}) = \frac{\mathbb{P}(\text{classif } + \text{ e é -})}{\mathbb{P}(\text{é -})}$$

e

$$\mathbb{P}(FN) = \mathbb{P}(\text{classificado como } - | \text{é } +) = \frac{\mathbb{P}(\text{classif - e é } +)}{\mathbb{P}(\text{é -})}$$

- No caso de verdadeiro-positivos , temos

$$\mathbb{P}(VP) = \mathbb{P}(\text{classificado como } + | \text{é } +) = \frac{\mathbb{P}(\text{classif } + \text{ e é } +)}{\mathbb{P}(\text{é } +)}$$

Recall ou revocação ou sensibilidade

- No caso de verdadeiro-positivos , temos

$$\mathbb{P}(VP) = \mathbb{P}(\text{classificado como } + | \text{é } +) = \frac{\mathbb{P}(\text{classif } + \text{ e é } +)}{\mathbb{P}(\text{é } +)}$$

- Esta probabilidade (estimada) é chamada de RECALL (revocação) em aprendizado de máquina ou de sensibilidade ou sensitividade em estudos epidemiológicos.
- Recall alto significa que o algoritmo retornou a maioria dos resultados relevantes.

Verdadeiro-negativos ou especificidade

- Quanto aos verdadeiro-negativos,

$$\mathbb{P}(VN) = \mathbb{P}(\text{classificado como -}|\text{é -}) = \frac{\mathbb{P}(\text{classif - e é -})}{\mathbb{P}(\text{é -})}$$

- Esta medida é chamada de *especificidade*.
- A idéia é que o algoritmo é específico para o que ele se propõe classficar.
- Se o item não é $+$, ele não retorna $+$.
- Veja que $\mathbb{P}(VN) + \mathbb{P}(FP) = 1$ pois um indivíduo que é negativo, será classificado ou como negativo (corretamente) ou como positivo (falsamente).
- Do mesmo modo, $\mathbb{P}(VP) + \mathbb{P}(FP) = 1$.

Estimando falso-positivos e falso-negativos

- Estimamos estas quantidades a partir dos dados comparando a verdadeira classe dos exemplos com a classe alocada a eles pela regressão logística.

	Diag -	Diag +
é -	429	71
é +	145	123

- Assim, o RECALL é estimado como

$$\mathbb{P}(VP) \approx \frac{123/768}{(145+123)/768} = \frac{123}{145+123} = 0.47$$

- Estamos acertando no diagnóstico de aprox metade dos verdadeiramente diabéticos.
- $\mathbb{P}(VN) \approx 429/(429+71) = 0.86$: acertamos mais frequentemente no diagnóstico dos verdadeiramente não-diabéticos.

Precisão, recall e especificidade

- Em aprendizado de máquina, uma métrica muito comum inverte os eventos usados na definição do RECALL.
- Temos RECALL igual a $\mathbb{P}(VP) = \mathbb{P}(\text{classificado como } +|\text{é } +)$.
- A PRECISÃO de um algoritmo de classificação é dada por

$$\text{Precisão} = \mathbb{P}(\text{é } +|\text{classificado como } +)$$

.

- Alta precisão indica que um algoritmo retornou mais resultados relevantes que irrelevantes.
- A partir da tabela anterior, podemos estimar a precisão como $123/(123 + 71) = 0.63$.
- Mais uma métrica, especificidade ($\mathbb{P}(VN) = \mathbb{P}(\text{classif -}|\text{e' -})$), estimada como $429/(429 + 71) = 0.86$.

Deep Learning

Semana 01 - Aula 03

Renato Assunção - DCC - UFMG

Roteiro desta aula

- Retomando classificação com duas classes
  - Com várias classes: máxima verossimilhança e logística
  - Problema de regressão: aptos e características; máxima verossimilhança
  - Função de custo = - log-verossimilhança
- Redes neurais com múltiplas camadas
- Longa explicação da notação e conceitos
- Exemplos com redes rasas (poucas camadas e poucos neurônios)
- Um digressão sobre porque as redes neurais funcionam:
  - Teorema de aproximação universal

Recapitulando...

- Vimos regressão logística e perceptron: algoritmos para classificação supervisionada em duas classes:
  - Temos dados estatísticos: coleção de exemplos ou casos
  - Duas classes rotuladas como Y= 0 ou 1
  - Para cada exemplo: inputs ou features num vetor x
  - Os mesmos inputs devem ser medidos em cada exemplo
- Objetivo: usar os dados para obter uma boa representação de $\mathbb{P}(Y = 1|\mathbf{x})$
- Esta probabilidade condicional é uma função matemática dos inputs x.
- Dados os inputs x, obtemos $\mathbb{P}(Y = 1|\mathbf{x})$

Como usar esta probabilidade?

- De posse da função $\mathbb{P}(Y = 1|\mathbf{x})$, fazemos classificações de novos exemplos onde Y não é conhecido.
  - Recebemos x
  - Calculamos $\mathbb{P}(Y = 1|\mathbf{x})$
  - Se for aprox 1, classifique na categoria 1.
    - É uma predição que pode ou não se confirmar.
  - Se for aprox 0, classique na categoria 0.
  - Se for aprox ½
    - neste exemplo, o input x não fornece informação suficiente para predizer a resposta Y. É uma caso em que a classificação tem grande incerteza.

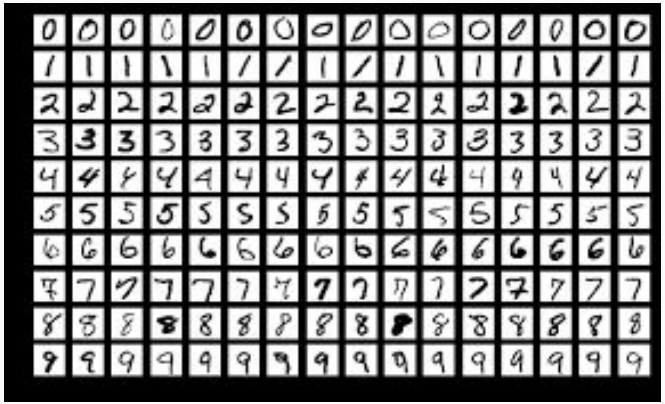
Expandindo um pouco os problemas

- Existem dois problemas adicionais intimamente relacionados com a classificação supervisionada em duas classes:
  - Classificação em K > 2 categorias
  - Regressão
- Estes dois problemas diferem do anterior pela estrutura da resposta Y
- Classificação em K > 2 categorias:
  - Y tem mais que duas classes: Y = 0, 1, 2, ..., K-1
  - O resto é igual
- Regressão:
  - Y é uma variável contínua, o resto é igual

Classificação multi-classes

Classificação com k categorias

- Exemplo canônico: MNIST
- Dados são 70 mil imagens de dígitos manuscritos: cada imagem, um único dígito.
- Y = 0, 1, 2, ..., 9 (resposta é o dígito exibido na imagem)
- Input: o tom de cinza (0-255) em cada pixel da imagem, empilhados como vetor



Criadores do MNIST

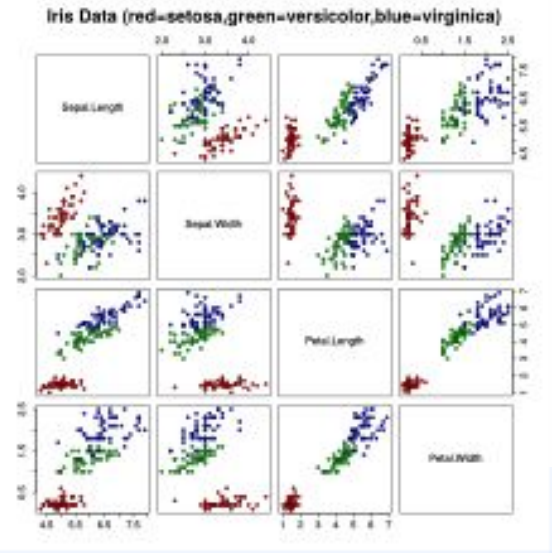
- Yann Le Cunn, Corinna Cortes, Christopher Burges
- Le Cunn é Silver Professor do Instituto Courant de Ciências Matemáticas da New York University e Chief AI Scientist no Facebook.
- Muito reconhecido por seu trabalho pioneiro em reconhecimento óptico de caracteres e visão computacional.
- Um dos principais criadores das redes neurais convolucionais (CNN), tópico de sexta.
- Ganhador do Turing Award de 2019 com Hinton e Bengio

Segundo exemplo canônico

- Iris dataset
- Dados de 150 flores
- Três categorias (espécies) de flores:
  - Iris setosa, Iris virginica, Iris versicolor
- Inputs: 4 variáveis:
  - comprimento e largura da pétala
  - Comprimento e largura da sépala
- Objetivo: criar um modelo para distinguir as espécies umas das outras com base nos 4 inputs.

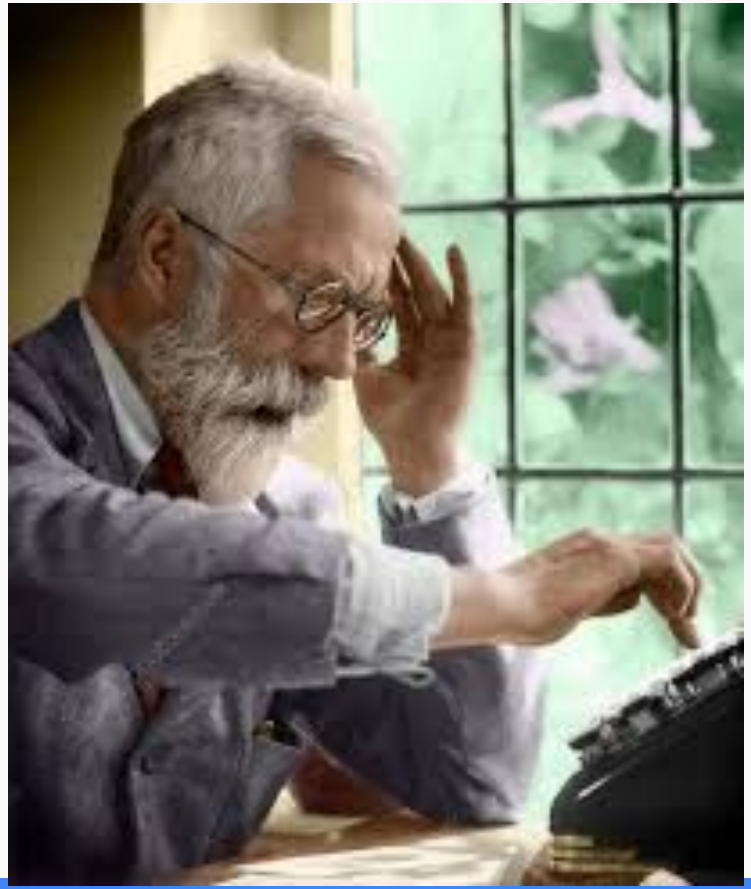
The Iris Dataset

Collected by Ronald Fisher in 1936



Sir Ronald Aylmer Fisher

- Iris dataset: usado pelo estatístico britânico Sir Ronald Fisher em seu artigo de 1936, *The use of multiple measurements in taxonomic problems,* como um exemplo de análise discriminante linear.
- Fisher foi também um dos maiores geneticistas da história, responsável por unir Darwin e Mendel de forma coerente.



Log-verossimilhança com DUAS classes

- m exemplos, duas classes: LOG-verossimilhança:

$$
\begin{aligned}
\ell(\mathbf{w}) &= \log(\ L(w_0, w_1, \ldots, w_n)\ ) \\
&= \log\left(\prod_{i=1}^{m} \mathbb{P}(Y_i = y_i)\right) \\
&= \log\left(\prod_{i=1}^{m} \sigma(\mathbf{x}_i)^{y_i}(1-\sigma(\mathbf{x}_i))^{1-y_i}\right) \\
&= \sum_{i=1}^{m} \log\left(\sigma(\mathbf{x}_i)^{y_i}(1-\sigma(\mathbf{x}_i))^{1-y_i}\right) \\
&= \sum_{i=1}^{m} \left(y_i\ \log(\sigma(\mathbf{x}_i)) + (1-y_i)\ \log(1-\sigma(\mathbf{x}_i))\right)
\end{aligned}
$$

- sendo que $\sigma(x_i) = \mathbb{P}(Y_i = 1 | \mathbf{x}_i) = \dfrac{1}{1+e^{-(w_0+w_1 x_{1i}+\ldots+w_n x_{ni})}}$

Olhando um único exemplo com DUAS classes

- Log-verossimilhança com m exemplos: soma sobre exemplos individuais
- Vamos olhar um único exemplo, o exemplo i
- Temos $\sigma(x_i) = \mathbb{P}(Y_i = 1|\mathbf{x}_i) = \frac{1}{1+e^{-(w_0+w_1x_{1i}+\ldots+w_nx_{ni})}}$
- A log-verossimilhança é a soma de m termos:

$$\ell(\mathbf{w}) = \sum_{i=1}^{m} \left(y_i \ \log(\sigma(\mathbf{x}_i)) + (1 - y_i) \ \log(1 - \sigma(\mathbf{x}_i)) \right)$$

$$y_i \ \log(\sigma(\mathbf{x}_i)) + (1 - y_i) \ \log(1 - \sigma(\mathbf{x}_i)) = \begin{cases} \log(\sigma(\mathbf{x}_i), & \text{se } y_i = 1 \\ \log(1 - \sigma(\mathbf{x}_i)), & \text{se } y_i = 0 \end{cases}$$

Com K classes o natural seria ter cada termo somando o log(probab) da classe realmente observada no exemplo i

A log-verossimilhança será exatamente isto. Vamos ver...

Estrutura estocástica para o caso multi-classe

- Em cada exemplo, a resposta Y é um rótulo indicando sua classe

$$Y_i = \begin{cases} 1, & \text{com probab } \sigma_1 \\ 2, & \text{com probab } \sigma_2 \\ \vdots & \\ K, & \text{com probab } \sigma_K \end{cases}$$

As probabilidades de cada classe

- Com duas classes, tínhamos apenas uma probabilidade de sucesso

$$\mathbb{P}(Y=1|\mathbf{x}) = \sigma(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}'\mathbf{x}}}$$

- A outra probabilidade era $1-\sigma(\mathbf{x})$
- A probabilidade de sucesso é função dos inputs x: diferentes x, diferentes sigmas
- Precisamos especificar agora K probabilidades, todas dependendo dos inputs x:

$$\mathbb{P}(Y=k|\mathbf{x}) = \sigma_k(\mathbf{x})$$

- tais que

$$\sigma_1(\mathbf{x}) + \ldots + \sigma_K(\mathbf{x}) = 1$$

A verossimilhança

- Suponha que, de alguma forma, especificamos $\sigma_1(\mathbf{x}), \ldots, \sigma_K(\mathbf{x})$
- Qual a chance de observar uma certa sequência de classes?
- Por exemplo, com K=3 classes, e 5 exemplos

$$
\begin{aligned}
L &= \mathbb{P}(Y_1 = 3|\mathbf{x}_1)\mathbb{P}(Y_2 = 1|\mathbf{x}_2)\mathbb{P}(Y_3 = 2|\mathbf{x}_3)\mathbb{P}(Y_4 = 1|\mathbf{x}_4)\mathbb{P}(Y_5 = 3|\mathbf{x}_5) \\
&= \sigma_3(\mathbf{x}_1)\sigma_1(\mathbf{x}_2)\sigma_2(\mathbf{x}_3)\sigma_1(\mathbf{x}_4)\sigma_3(\mathbf{x}_5) \\
&= \prod_{i=1}^{m}\prod_{k=1}^{3}\sigma_k(\mathbf{x}_i)^{I[y_i=k]}
\end{aligned}
$$

A LOG-verossimilhança

- Basta tomar o log agora:
- Produtos viram somas:

$$\ell = \log\left[\prod_{i=1}^{m}\prod_{k=1}^{3}\sigma_k(\mathbf{x}_i)^{I[y_i=k]}\right]$$

$$= \sum_{i=1}^{m}\sum_{k=1}^{3}\left[\log\sigma_k(\mathbf{x}_i)^{I[y_i=k]}\right]$$

$$= \sum_{i=1}^{m}\sum_{k=1}^{3}\left[I[y_i = k]\ \log\sigma_k(\mathbf{x}_i)\right]$$

- Nosso exemplo fica então

$$\ell = \log(\sigma_3(\mathbf{x}_1)) + \log(\sigma_1(\mathbf{x}_2)) + \log(\sigma_2(\mathbf{x}_3)) + \log(\sigma_1(\mathbf{x}_4)) + \log(\sigma_3(\mathbf{x}_5))$$

- Isto é, cada exemplo contribui com o log da probab da sua classe observada

As probabilidades das classes: de duas para K classes

- No caso de duas classes:

$$\sigma(\mathbf{x}_i) = \mathbb{P}(Y_i = 1|\mathbf{x}_i) = \frac{1}{1+e^{-(w_0+w_1x_{1i}+\ldots+w_nx_{ni})}}$$

- A probabilidade da classe 0 é obtida por subtração
- Podemos escrever

$$\frac{1}{1+e^{-(w_0+w_1x_{1i}+\ldots+w_nx_{ni})}} = \frac{e^{(w_0+w_1x_{1i}+\ldots+w_nx_{ni})}}{1+e^{(w_0+w_1x_{1i}+\ldots+w_nx_{ni})}} \propto e^{(w_0+w_1x_{1i}+\ldots+w_nx_{ni})} = e^{\mathbf{w}'\mathbf{x}_i}$$

- Para o caso multi-classes, especificamos um vetor de pesos para cada uma das K classes: $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K$

Especificando as probabilidades das classes: softmax

- Para o caso multi-classes, especificamos um vetor de pesos para cada uma das K classes: $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K$
- As probabilidades $\sigma_k(\mathbf{x}_i) = \mathbb{P}(Y_i = k|\mathbf{x}_i) \propto e^{\mathbf{w}_k' \mathbf{x}_i}$
- Elas devem somar 1. Basta normalizarmos agora (modelo softmax):

$$\sigma_k(\mathbf{x}_i) = \mathbb{P}(Y_i = k|\mathbf{x}_i) = \frac{e^{\mathbf{w}_k' \mathbf{x}_i}}{\sum_{j=1}^{K} e^{\mathbf{w}_j' \mathbf{x}_i}}$$

Resultado final: log-verossimlhança para multi-classe

- Combinando a log-verossimilhança de antes com esta expressão para as probabilidades das classes, temos

$$
\begin{aligned}
\ell &= \ell(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K) \\
&= \sum_{i=1}^{m} \sum_{k=1}^{K} \left[I[y_i = k] \ \log \sigma_k(\mathbf{x}_i) \right] \\
&= \sum_{i=1}^{m} \sum_{k=1}^{K} \left[I[y_i = k] \ \log \left(\frac{e^{\mathbf{w}_k' \mathbf{x}_i}}{\sum_{j=1}^{K} e^{\mathbf{w}_j' \mathbf{x}_i}} \right) \right]
\end{aligned}
$$

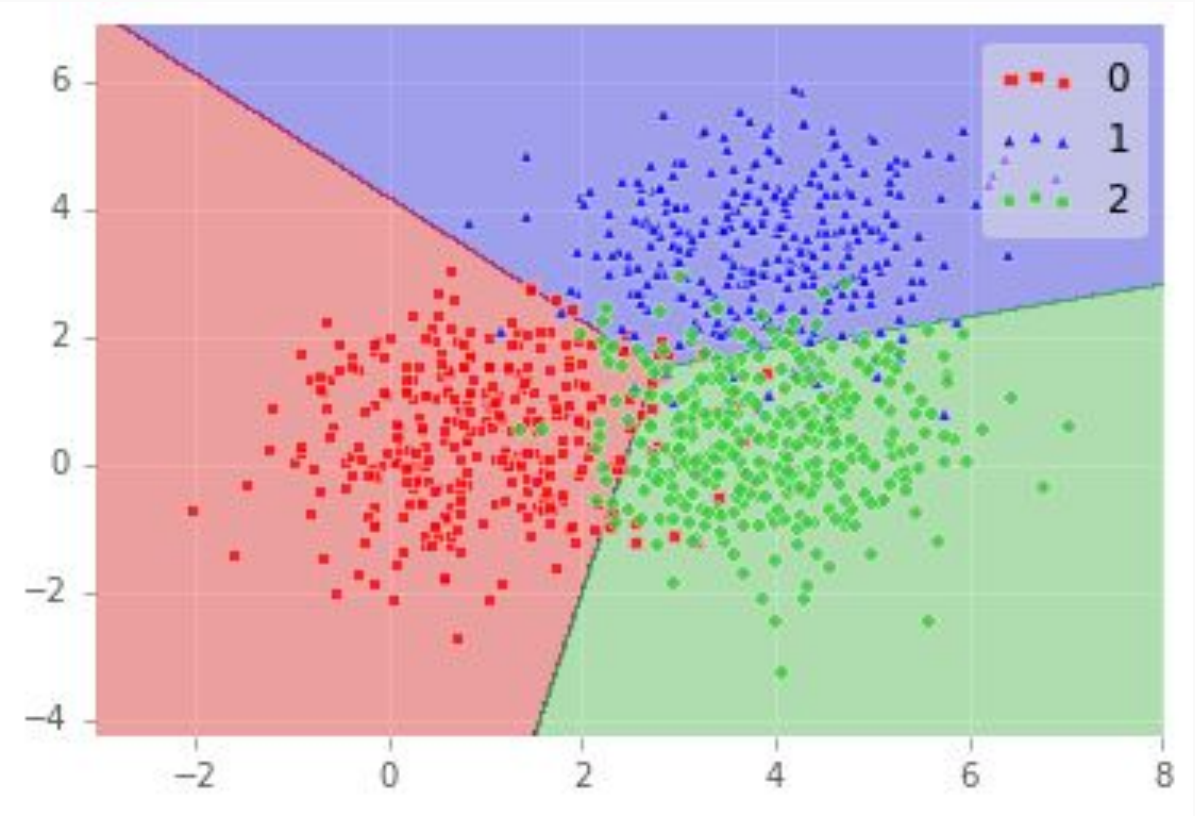
Como obter o estimador de máxima verossimilhança?

- Método numérico de Newton

$$\mathbf{w}^{t+1} = \begin{bmatrix} w_0^{t+1} \\ w_1^{t+1} \\ \vdots \\ w_n^{t+1} \end{bmatrix} = \begin{bmatrix} w_0^{t} \\ w_1^{t} \\ \vdots \\ w_n^{t} \end{bmatrix} - \Big[\underbrace{J^2(\mathbf{w}^t)}_{\text{matriz der. parciais 2a ordem de J}} \Big]^{-1} \underbrace{\nabla J(\mathbf{w}^t)}_{\text{vetor gradiente de J}}$$

- Vetor gradiente da log-verossimilhança.
- Se temos K classes e p inputs, teremos vetor gradiente de dimensão K*(p+1)-dim
- Matriz hessiana de derivadas parciais de segunda ordem: K*(p+1) x K*(p+1)

A regra de decisão e os decision boundaries



Função custo e gradiente descendente

Maximizar a log-verossimilhança ou minimizar o custo

- Em machine learning, é mais comum falar em minimizar uma função custo.
- A função custo é o NEGATIVO da log-verossimilhança

$$J(w_0, w_1, \ldots, w_n) = -\ell(w_0, w_1, \ldots, w_n) = -\log L(w_0, w_1, \ldots, w_n)$$

- Usamos o método de Newton como antes.
- Newton acha mínimos e máximos.
- Nossos modelos de redes neurais terão muitos e muitos parâmetros.
- O cálculo do Hessiano (matriz de derivadas parciais de 2a ordem) será proibitivo.
- Vamos adotar outro método, mais simples e talvez com convergência mais lenta.

Esqueça o hessiano, use apenas o gradiente

- Ao invés de usar

$$\mathbf{w}^{k+1} = \begin{bmatrix} w_0^{k+1} \\ w_1^{k+1} \\ \vdots \\ w_n^{k+1} \end{bmatrix} = \begin{bmatrix} w_0^{k} \\ w_1^{k} \\ \vdots \\ w_n^{k} \end{bmatrix} - \begin{bmatrix} \underbrace{J^2(\mathbf{w}^k)}_{\text{matriz de parciais 2a ordem de J}} \end{bmatrix}^{-1} \underbrace{\nabla J(\mathbf{w}^k)}_{\text{vetor gradiente de J}}$$

- usamos

$$\mathbf{w}^{k+1} = \begin{bmatrix} w_0^{k+1} \\ w_1^{k+1} \\ \vdots \\ w_n^{k+1} \end{bmatrix} = \begin{bmatrix} w_0^{k} \\ w_1^{k} \\ \vdots \\ w_n^{k} \end{bmatrix} - \alpha \underbrace{\nabla J(\mathbf{w}^k)}_{\text{vetor gradiente de J}}$$

α é um escalar positivo e pequeno.

Por exemplo, é comum usar $\alpha = 0.01$

Resumo do problema supervisionado

- Amostra de m exemplos de dados:
  - Uma resposta Y
  - Inputs-features num vetor x
- Objetivo: representar o valor esperado de Y através de uma função dos inputs
  - Dado x, quanto é $\mathbb{E}(Y|x)$?
    - No caso binário: $\mathbb{E}(Y|x) = \mathbb{P}(Y = 1|x)$
    - No caso multi-classe: $\mathbb{P}(Y = k|x)$
    - No caso contínuo: $\mathbb{E}(Y|x) = \mu(x)$
- Esta função de x depende de parâmetros-pesos $\mathbf{w} = (w_0, w_1, \ldots, w_n)$
- Obtenha a log-verossimilhança dos pesos: a probabilidade de gerar os dados da amostra para cada possível valor dos pesos.
- Função de custo: $J(w_0, w_1, \ldots, w_n)$ = - Log-verossmilhança $\rightarrow$ Minimize a função com Newton /GD