AGA0513 - eScience em Astronomia

Introdução a *Machine Learning*

Nina S. T. Hirata
(nina@ime.usp.br)

Departamento de Ciência da Computação
Instituto de Matemática e Estatística
(IME - USP)

Maio / 2017

DCC – IME/USP

Departamento de Ciência da Computação (IME)

Bacharelado em Ciência da Computação

MACxxx - Introdução à computação
(Poli, IF, IAG, IGc, FEA, IO, Moleculares, ICB)

Processamento e análise de imagens

Machine learning

Colaboração com o IAG

NAP LabCosmos (coord. Profa. Cláudia Oliveira)

- Separação de estrelas e quasares (ALHAMBRA)
- Classificação de imagens de galáxias
- Deblending de pixels em imagens

Pesquisas em imagens em machine learning

Aplicações:

- classificação de imagens de plâncton
- imagens de telescópio óptico
- imagens de documentos
- imagens de comics
- expressões matemáticas manuscritas
- imagens de retina (vasos sanguíneos)
- detecção de QR codes

Exemplos de problemas tratados com ML

– Reconhecer objetos, pessoas, etc, em imagens

– Reconhecer texto/escrita

– Reconhecer falas e sons

– diagnóstico médico

– qual anúncio publicar no meu site ?

– identificar prenúncios de um desastre natural

Evolução ao longo dos anos ...

Fascínio por máquinas inteligentes / ficção

Escritórios (documentos) / indústria (linha de produção)

Avanço em tecnologia de processamento de dados – computação eficiente

Avanço em tecnologia de sensores – abundância de dados

Veículos autônomos, tradutores simultâneos, jogadores (xadrez, go, etc)

Big data, eScience

- Volume
- Velocity
- Variety
- Veracity / Variability
- Value

Desafio: Extrair informação útil ou conhecimento a partir do volume de dados

Estatística e **machine learning** desempenham papel importante nesse processo

O que é *Machine learning*

1. **Contexto** (estudo em questão)

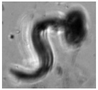
2. **Recorte** X dentro do contexto: observação + informação (conhecimento sobre a observação)
   - observações (<u>entrada</u> $\mathbf{x}$)
   - estado, identidade, ação (<u>saída</u> y) – conhecimento

3. **Suposição**: existe uma relação entre X e y

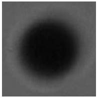
4. **Objetivo**: determinar a relação entre X e y, de tal forma que dado um elemento x qualquer de X eu consiga predizer a saída correspondente y

Recorte: classificação de imagens de plâncton

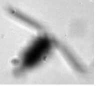
Espaço X: imagens de plâncton



Appendicularia

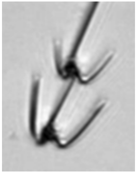


Bubble

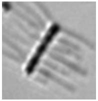


Calanoida

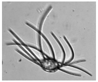
Dinoflagellate



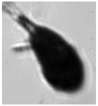
Dinoflagellate M



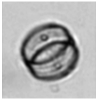
Chaetoceros



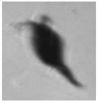
Cnidaria



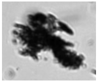
Copepoda



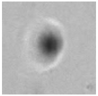
Coscinodiscus



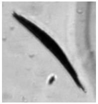
Cyclopoida



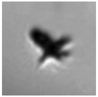
Detritus



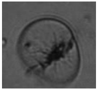
Detritus ball



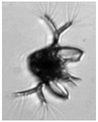
Filaments



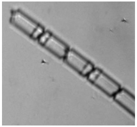
Nauplii



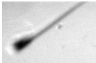
Noctiluca



Penilia



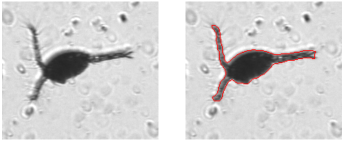
P. Dactyliosolen



Stalked c.

Conceitos

Extrair *features*: converter representação RAW para outra $\longrightarrow (x_1, x_2, \ldots, x_n)$



– informação de forma, cor, textura, ...

Modelar a relação:

– y dado $\Longrightarrow$ supervisionado (classificação, regressão)

– y não dado $\Longrightarrow$ não-supervisionado (clustering)

Supervised Classification

Entrada: *features* (atributos)

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$$

Saída:

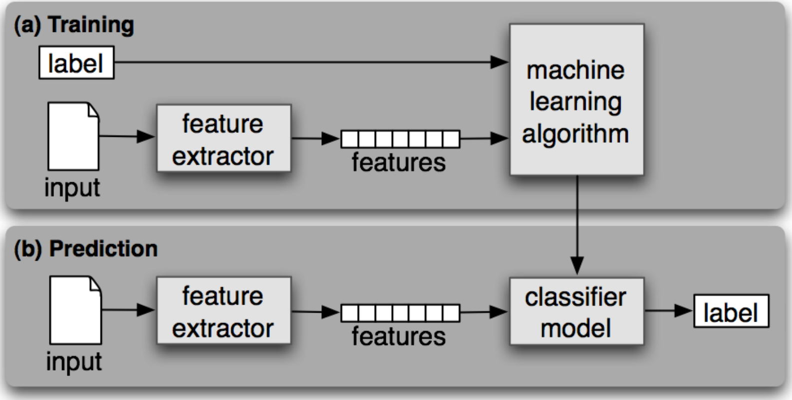
– Classificação: rótulo de classe $y \in Y = \{1, 2, \ldots, c\}$

– Regressão: um valor $y \in Y = \mathbb{R}$

Conjunto de exemplos de treinamento

$$S = \{(\mathbf{x}_i, y_i) : i = 1, 2, \ldots, N\}$$

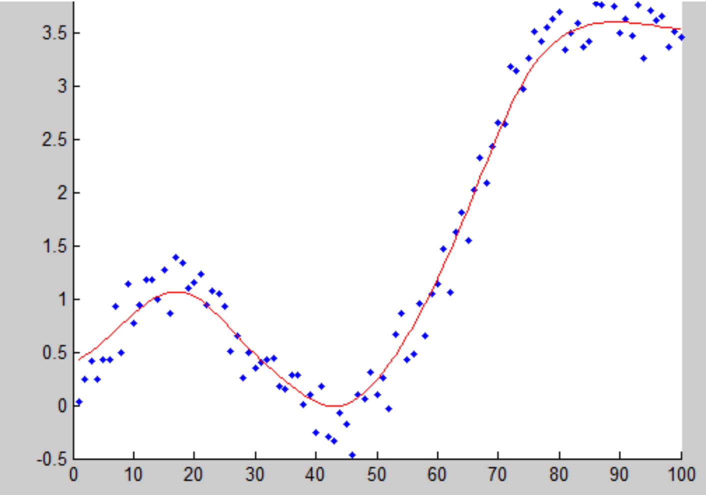
Treinamento / predição



http://www.nltk.org/book/ch06.html

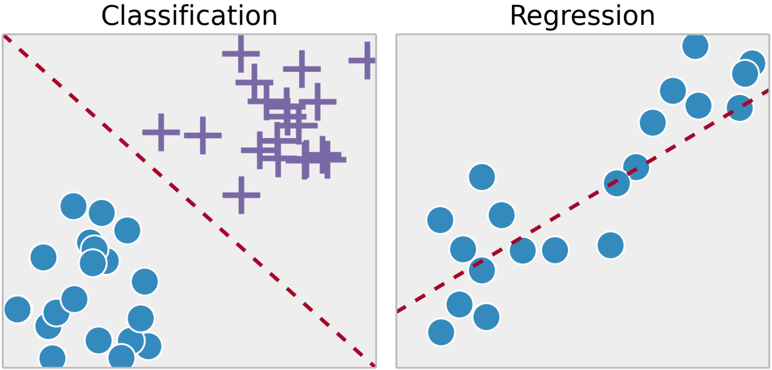
Regressão

$n = 1$



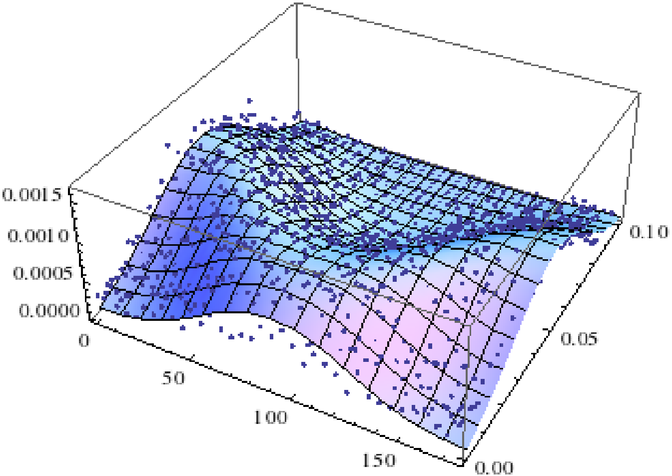
Classificação × regressão

Classificação: encontrar uma fronteira de decisão
Regressão: ajustar uma curva/superfície aos dados



$n = 2$ e $n = 1$, resp

Regressão



$n = 2$

Prática

1. abrir o arquivo `html/main.html` no browser
2. item 2.A
3. abrir o notebook `practice_basics.ipynb`
4. executar o notebook `practice_basics.ipynb` (mesmo conteúdo de 2.A)

Função linear de uma variável

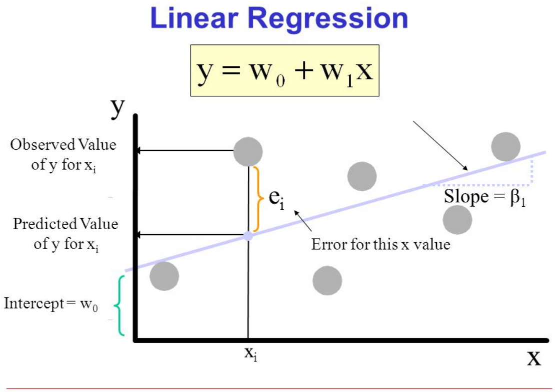
$x \in \mathbb{R}$

$$h(x) = w_0 + w_1 x_1$$

$\mathbf{x} = (1, x) \in \mathbb{R}^2$
$\mathbf{w} = (w_0, w_1)$

$$h(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

Regressão linear



Regressão linear

$$(x_1, x_2, \ldots, x_n) \Longrightarrow (1, x_1, x_2, \ldots, x_n) \in \mathbb{R}^{n+1}$$

$\mathbf{w} = (w_0, w_1, \ldots, w_n) \in \mathbb{R}^{n+1}$ vetor de pesos

$$h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n = \sum_{j=0}^{n} w_j x_j$$

$$h_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}^T \mathbf{w}$$

Queremos $h_{\mathbf{w}}(\mathbf{x}_i) \approx y_i$

Uma ideia simples:

1. chutar um vetor de pesos qualquer $\mathbf{w}$

2. verificar se a função linear se ajusta adequadamente

3. se sim, terminar

4. senão, alterar ligeriramente a função linear (i.e., $\mathbf{w}$) e voltar para o passo 2

Mas até quando repetir isso?

Se $z_i = h_{\mathbf{w}}(\mathbf{x}_i) = \mathbf{x}^T\mathbf{w}$

queremos que a diferença entre y_i e z_i seja a menor possível.

Essa diferença pode ser expressa, por exemplo, por

$$
\begin{aligned}
SSR_D(\mathbf{w}) &= \sum_{\mathbf{x}_i \in D} (y_i - z_i)^2 \\
&= \sum_{\mathbf{x}_i \in D} (y_i - \mathbf{x}^T\mathbf{w})^2
\end{aligned}
$$

Problema: queremos encontrar o ponto $\mathbf{w}$ que minimiza SSR_D

Método do gradiente descendente

Função custo

$$J_D(\mathbf{w}) = \frac{1}{2} \sum_{\mathbf{x}_i \in D} (y_i - z_i)^2$$

É uma função quadrática positiva; logo tem ponto de mínimo

Gradiente descendente: Método de otimização de função, que consiste em:

- chutar um valor para $\mathbf{w}$
- calcular o vetor gradiente de J_D em $\mathbf{w}$
- alterar $\mathbf{w}$ na direção oposta ao do vetor gradiente

Método do gradiente descendente

Vetor gradiente de J:

$$\nabla J(\mathbf{w}) = \left[\frac{\partial J}{\partial w_0}, \frac{\partial J}{\partial w_1}, \ldots, \frac{\partial J}{\partial w_n}\right]$$

$$\begin{aligned}
\frac{\partial J}{\partial w_j} &= \frac{\partial}{\partial w_j}\frac{1}{2}\sum_i (y_i - z_i)^2 \\
&= \frac{1}{2}\sum_i \frac{\partial}{\partial w_j}(y_i - z_i)^2 \\
&= \frac{1}{2}\sum_i 2(y_i - z_i)\frac{\partial}{\partial w_j}(y_i - z_i) \\
&= \sum_i (y_i - z_i)\frac{\partial}{\partial w_j}\big(y_i - (w_0 + w_1 x_{i1} + \ldots + w_j x_{ij} + \ldots + w_n x_{in})\big) \\
&= -\sum_i (y_i - z_i)\, x_{ij}
\end{aligned}$$

Método do gradiente descendente

Peso inicial: $\mathbf{w}(0)$

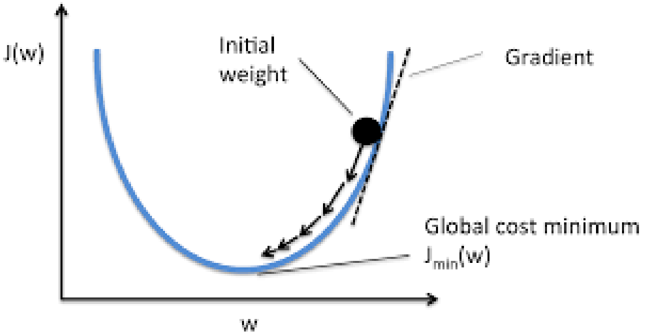
Regra de atualização:

$$\mathbf{w}(r+1) = \mathbf{w}(r) + \eta\Delta\mathbf{w}(r)$$

$$\Delta\mathbf{w}(r) = -\nabla J(\mathbf{w})$$

$$\Delta w_j(r) = \sum_i (y_i - z_i)\, x_{ij}$$

Método do gradiente descendente



Regressão linear: resumo

- Ajustar uma função linear aos dados pode ser modelado como um problema de minimizar uma função custo J_D expressa em termos da soma do quadrado dos resíduos (RSS).

- Para cada função linear, caracterizado por um vetor de pesos $\mathbf{w}$ específico, há um custo $J_D(\mathbf{w})$ associado

- O ponto de mínimo pode ser calculado usando-se o método do gradiente descendente

Prática

1. item 2.B

2. abrir e executar o notebook
   `practice_regression.ipynb`

Classificação

Problema que queremos resolver: dado um conjunto de amostras, encontrar uma fronteira de decisão que melhor separe as duas classes.

Pontos a serem considerados:

- que tipo de fronteira de decisão ?
- qual o significado de "melhor separação" ?
- quantas classes são ?
- etc

Classificação

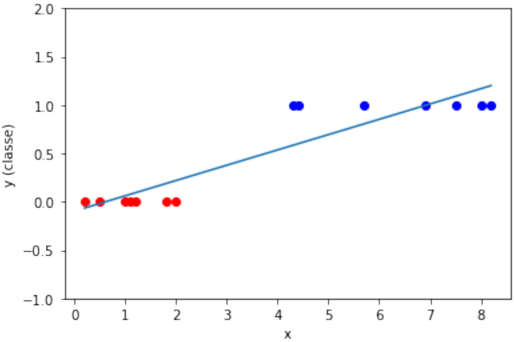
$y = 0$: exemplo negativo
$y = 1$: exemplo positivo

Podemos aplicar a regressão linear e decidir da seguinte forma ?

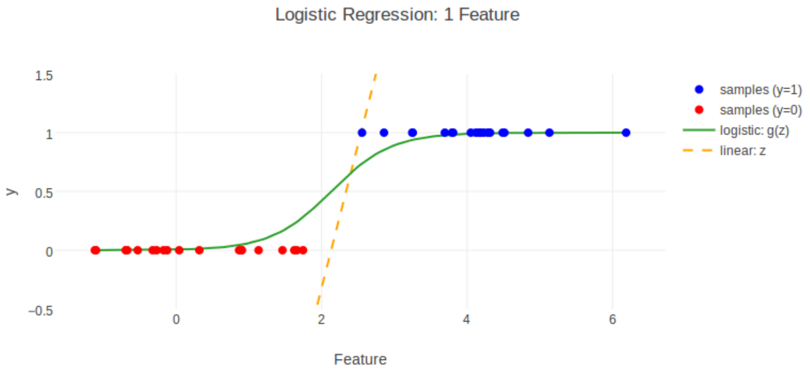
$$z = \begin{cases} 1, & \text{se } h_{\mathbf{w}}(\mathbf{x}) > 0, \\ 0, & \text{se } h_{\mathbf{w}}(\mathbf{x}) \leqslant 0. \end{cases}$$

Pergunta: é uma boa solução ?

Parece que não ...



Em problemas de classificação, podemos usar a regressão logística

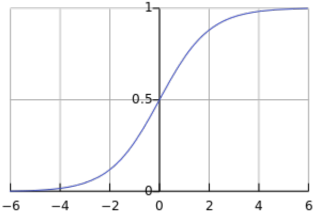


Logistic Regression: 1 Feature
1.5
1
0.5
0
−0.5
y
0
2
4
6
Feature
samples (y=1)
samples (y=0)
logistic: g(z)
linear: z

Função logística

Função loística (sigmoide):

$$s(x) = \frac{1}{1 + e^{-x}}$$



$$0 \leqslant s(x) \leqslant 1$$

Função custo

Regressão linear

$$J_D(\mathbf{w}) = \frac{1}{2} \sum_{\mathbf{x}_i \in D} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

Classificação: poderia ser essa ?

$$J_D(\mathbf{w}) = \frac{1}{2} \sum_{\mathbf{x}_i \in D} (y_i - s(\mathbf{w}^T \mathbf{x}_i))^2$$

Função logística como probabilidade

$$h_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}^T\mathbf{w} = \sum_{j=0}^{n} w_j x_j$$

$$g_0(\mathbf{x}) = s(h_{\mathbf{w}}(\mathbf{x})) = \frac{1}{1 + e^{-h_{\mathbf{w}}(\mathbf{x})}}$$

$$g_0(\mathbf{x}) = 1 - g_1(\mathbf{x})$$

$$h_{\mathbf{w}}(\mathbf{x}) = 0 \iff g_0(\mathbf{x}) = g_1(\mathbf{x}) = 0.5$$
$$h_{\mathbf{w}}(\mathbf{x}) < 0 \iff g_0(\mathbf{x}) > g_1(\mathbf{x})$$
$$h_{\mathbf{w}}(\mathbf{x}) > 0 \iff g_0(\mathbf{x}) < g_1(\mathbf{x})$$

Regressão logística

Função de verossimilhança

$$L(\mathbf{w}) = \prod_{i=1}^{N} P(y_i \,|\, \mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{N} [g_1(\mathbf{x}_i)]^{y_i} [1 - g_1(\mathbf{x}_i)]^{(1-y_i)}$$

Função de custo: O ponto de máximo da função L é o ponto de mínimo da função

$$J(\mathbf{w}) = -\ln L(\mathbf{w})$$

Assim, pelo método do gradiente descendente obtém-se:

$$\Delta w_j(r) = \sum_{i=1}^{N} (y_i - g_1(\mathbf{x}_i))\mathbf{x}_{ij}$$

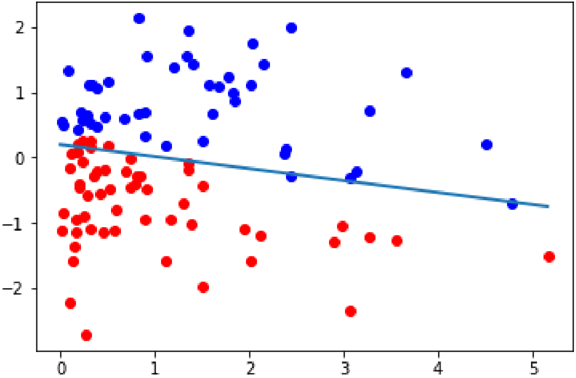
Regressão logística: resumo

- Usado para encontrar uma fronteira de decisão linear para problemas de classificação

- O valor $h_{\mathbf{w}}(\mathbf{x})$ é modulado pela sigmoide s de forma que fique no intervalo $[0, 1]$ antes do cálculo da distância até y

- O vetor de peso $\mathbf{w}$ pode ser obtido aplicando-se o método do gradiente descendente ao log da função de verossimilhança

- A fronteira de decisão resultante é linear



2
1
0
−1
−2
0
1
2
3
4
5

Fronteiras de decisão não lineares

1. Criar novas *features*

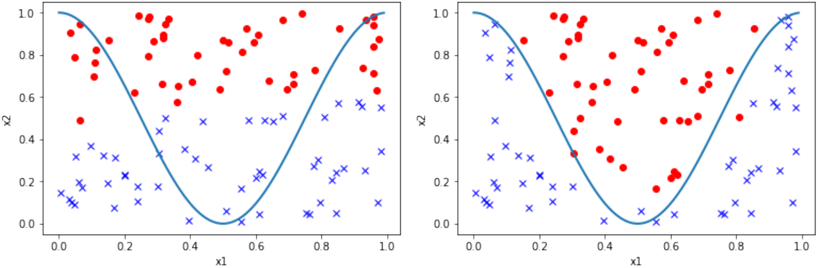
Exemplo: $(1, x_1, x_2) \Longrightarrow (1, x_1, x_2, x_1^2, x_1x_2, x_2^2)$

$n = 2$ passa para $n' = 5$

e aplicar a regressão logística

2. Usar modelos não-lineares (ex.: rede neural)

Fronteiras de decisão não lineares



linear × não linear (SVM)

Prática

1. item 2.C

2. abrir e executar o notebook
   `practice_classification1.ipynb`

3. abrir e executar o notebook
   `practice_classification2.ipynb`

Prática – sckitlearn

http://scikit-learn.org/

1. item 2.D e 2.E

2. abrir e executar o notebook
   `practice_scikitlearn.ipynb`

3. abrir e executar o notebook
   `mais_scikitlearn.ipynb`

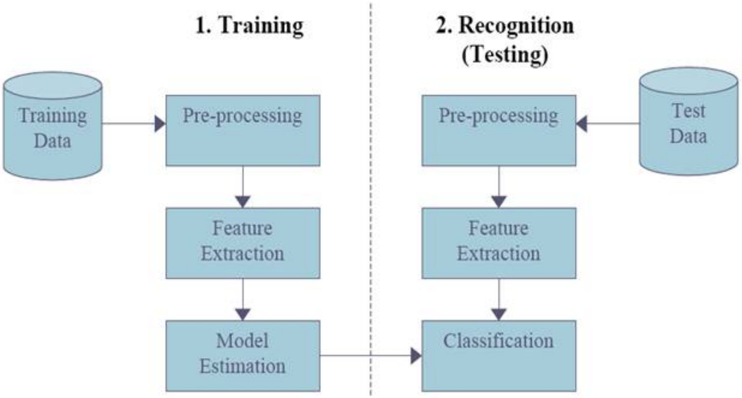


Figure 2: The pattern classification process