

Regressão Não Paramétrica

Prof. Dr. Leandro Balby Marinho



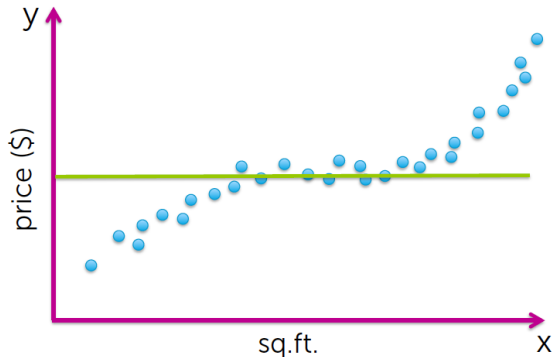
Aprendizagem de Máquina

Roteiro

1. Algoritmos dos Vizinhos mais Próximos

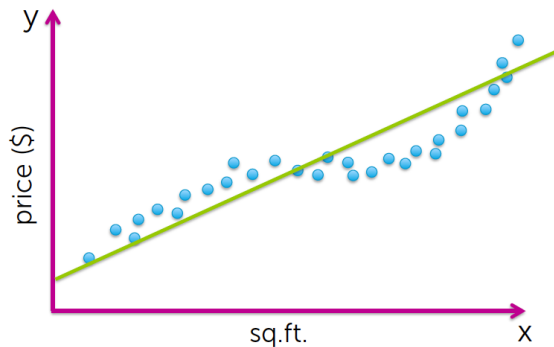
3. Regressão Kernel

Introdução



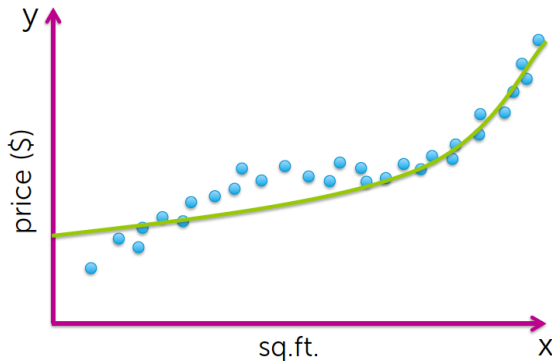
Regressão Linear tem flexibilidade limitada.

Introdução



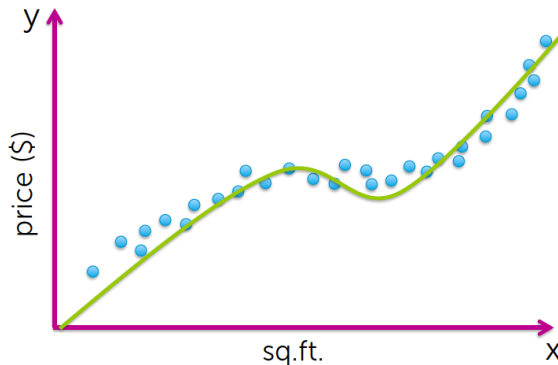
Regressão Linear tem flexibilidade limitada.

Introdução



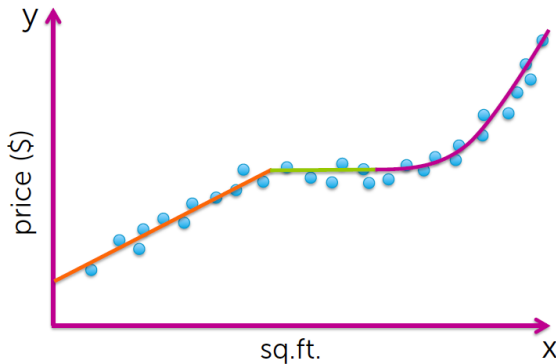
Regressão Linear tem flexibilidade limitada.

Introdução



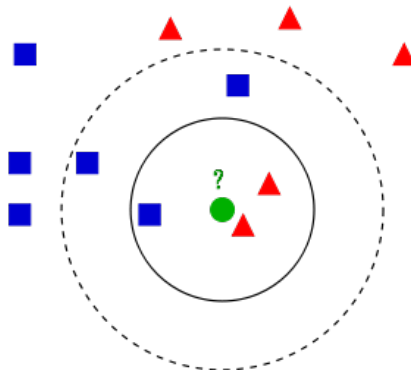
Regressão Linear tem flexibilidade limitada.

Introdução



Regressão Linear tem flexibilidade limitada.

Explorando localidade



Como prever o valor da instância de teste destacada em verde?

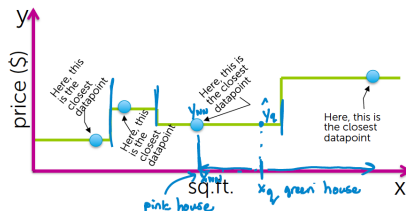
Regressão com 1-NN

- ▶ $\mathcal{D}^{train} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$
- ▶ Query: $\mathbf{x}^{(q)}$

1. Ache o ponto mais próximo a $\mathbf{x}^{(q)}$:

$$\mathbf{x}_{NN} = \min_i \text{distance}(\mathbf{x}^{(q)}, \mathbf{x}^{(i)})$$

2. Predição: $\hat{y}^{(q)} = y_{NN}$



Visualizando 1-NN em múltiplas dimensões

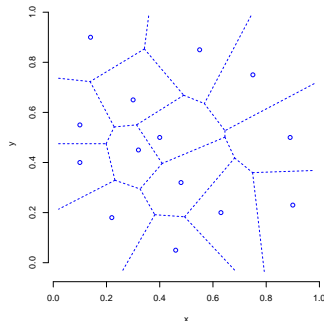


Diagrama de Voronoi.

- ▶ Divide o espaço em N regiões com um ponto em cada uma.
- ▶ Qualquer ponto em uma região é mais próximo ao ponto da região do que qualquer outro ponto de outras regiões.

Medidas de Distância

Seja d uma **medida de distância** (também chamada de **métrica**) da forma

$$d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_0^+$$

onde \mathcal{X} é um conjunto qualquer e d satisfaz:

- ▶ **Positiva-definida:** $d(x, y) \geq 0$ e $d(x, y) = 0 \Leftrightarrow x = y$
- ▶ **Simétrica:** $d(x, y) = d(y, x)$
- ▶ **Desigualdade Triangular:** $d(x, z) \leq d(x, y) + d(y, z)$

para quaisquer $x, y, z \in \mathcal{X}$.

Métrica de Minkowski para Vetores

Métrica de Minkowski/ norma ℓ_r em $\mathcal{X} := \mathbb{R}^n$:

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n |\mathbf{x}_i - \mathbf{y}_i|^r \right)^{1/r}$$

com $r \in \mathbb{R}, r \geq 1$.

$r = 1$ (**distância de Manhattan**, norma ℓ_1)

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n |\mathbf{x}_i - \mathbf{y}_i| \right)$$

Também chamada de distância de Hamming em vetores binários.

Métrica de Minkowski para Vetores

Métrica de Minkowski/ norma ℓ_r em $\mathcal{X} := \mathbb{R}^n$:

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n |\mathbf{x}_i - \mathbf{y}_i|^r \right)^{1/r}$$

com $r \in \mathbb{R}, r \geq 1$.

$r = 2$ (**distância Euclidiana**, norma ℓ_2)

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n (\mathbf{x}_i - \mathbf{y}_i)^2 \right)^{1/2}$$

Métrica de Minkowski para Vetores

Métrica de Minkowski/ norma ℓ_r em $\mathcal{X} := \mathbb{R}^n$:

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n |\mathbf{x}_i - \mathbf{y}_i|^r \right)^{1/r}$$

com $r \in \mathbb{R}, r \geq 1$.

$r = \infty$ (norma ℓ_∞)

$$d(\mathbf{x}, \mathbf{y}) = \max_{i=1}^n |\mathbf{x}_i - \mathbf{y}_i|$$

Exemplo 1

Calcule as distâncias ℓ_1, ℓ_2 para os dois vetores em \mathbb{R}^3 abaixo.

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 4 \\ 2 \\ 5 \end{bmatrix}$$

Exemplo 1

Calcule as distâncias ℓ_1, ℓ_2 para os dois vetores em \mathbb{R}^3 abaixo.

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 4 \\ 2 \\ 5 \end{bmatrix}$$

$$d_{\ell_1} = |1 - 4| + |2 - 2| + |3 - 5| = 3 + 0 + 2 = 5$$

$$d_{\ell_2} = \sqrt{(1 - 4)^2 + (2 - 2)^2 + (3 - 5)^2} = \sqrt{9 + 0 + 4} = 3.6$$

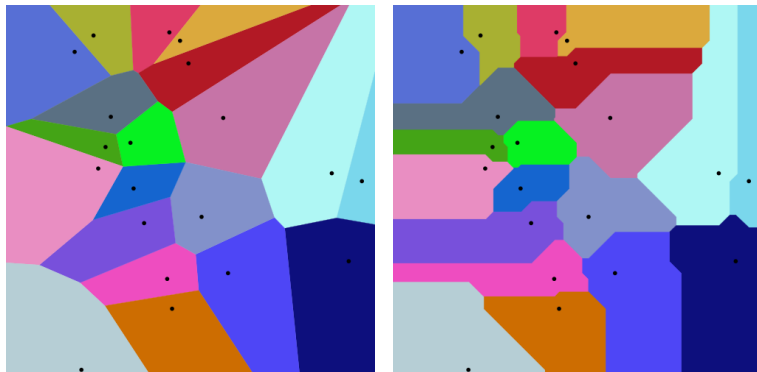
Distância Euclidiana Ponderada

Podemos dar mais importância a alguns atributos nos cálculos de distâncias:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{a_1(\mathbf{x}_1 - \mathbf{y}_1)^2 + a_2(\mathbf{x}_2 - \mathbf{y}_2)^2 + \dots + a_d(\mathbf{x}_d - \mathbf{y}_d)^2}$$

Onde a_i é peso associado ao i -ésimo atributo.

Métricas diferentes implica regiões de predição diferentes



Normas ℓ_2 (esquerda) e ℓ_1 (direita)

Algoritmo do Vizinho mais Próximo

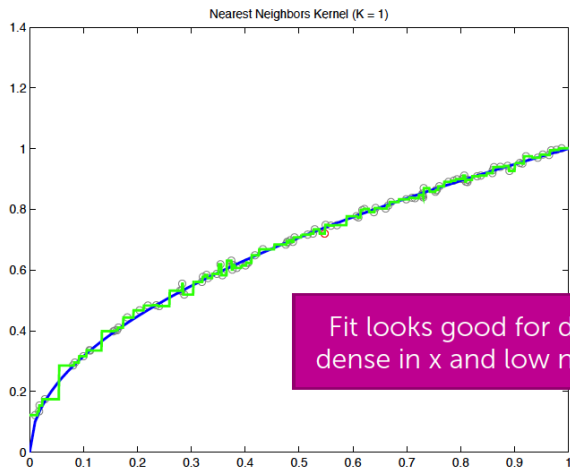
- ▶ Não há uma fase construção do modelo.
- ▶ O modelo é construído para cada instância de teste (**lazy learner**).
- ▶ Simples e fácil de implementar.
- ▶ Parâmetro: número dos k melhores vizinhos.
- ▶ Usado em muitas aplicações (e.g. sistemas de recomendação).

Algoritmo 1-NN

1-NN($\mathbf{x}^{(q)}$)

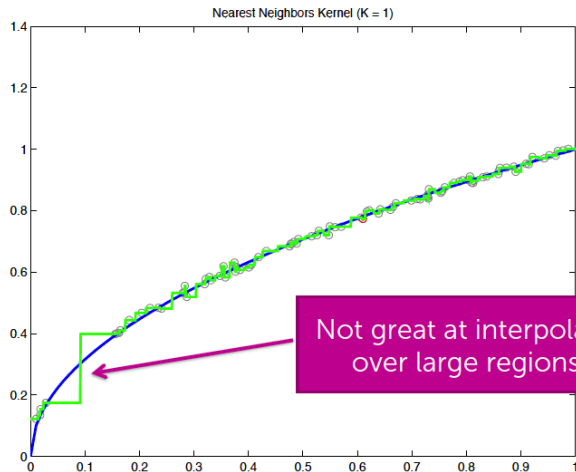
```
1  init Dist2NN =  $\infty$ ,  $\mathbf{x}_{NN} = \emptyset$ 
2  for  $i = 1$  to  $N$ 
3       $\delta = d(\mathbf{x}^{(i)}, \mathbf{x}^{(q)})$ 
4      if  $\delta < \mathbf{Dist2NN}$ 
5           $\mathbf{x}_{NN} = \mathbf{x}^{(i)}$ 
6          Dist2NN =  $\delta$ 
7  return  $\mathbf{x}_{NN}$ 
```

1-NN na prática

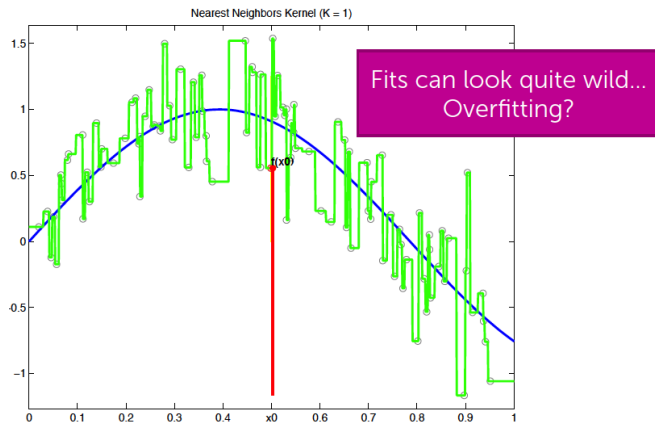


Fit looks good for data
dense in x and low noise

1-NN na prática



1-NN na prática



Regressão com K-NN

- ▶ $\mathcal{D}^{train} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$
 - ▶ Query: $\mathbf{x}^{(q)}$
1. Ache os K vizinhos de $\mathbf{x}^{(q)}$: $(\mathbf{x}_{NN_1}, \mathbf{x}_{NN_2}, \dots, \mathbf{x}_{NN_K})$
 - ▶ Tal que para qualquer $\mathbf{x}^{(i)}$ não contido no conjunto:
 $d(\mathbf{x}^{(i)}, \mathbf{x}^{(q)}) \geq d(\mathbf{x}_{NN_K}, \mathbf{x}^{(q)})$
 2. Predição: $\hat{y}^{(q)} = \frac{1}{K}(y_{NN_1} + y_{NN_2} + \dots + y_{NN_K})$

Regressão com K-NN

► $\mathcal{D}^{train} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$

► Query: $\mathbf{x}^{(q)}$

1. Ache os K vizinhos de $\mathbf{x}^{(q)}$: $(\mathbf{x}_{NN_1}, \mathbf{x}_{NN_2}, \dots, \mathbf{x}_{NN_K})$

- Tal que para qualquer $\mathbf{x}^{(i)}$ não contido no conjunto:
 $d(\mathbf{x}^{(i)}, \mathbf{x}^{(q)}) \geq d(\mathbf{x}_{NN_K}, \mathbf{x}^{(q)})$

2. Predição: $\hat{y}^{(q)} = \frac{1}{K}(y_{NN_1} + y_{NN_2} + \dots + y_{NN_K})$

Mais confiável se compararmos com um conjunto maior de vizinhos.

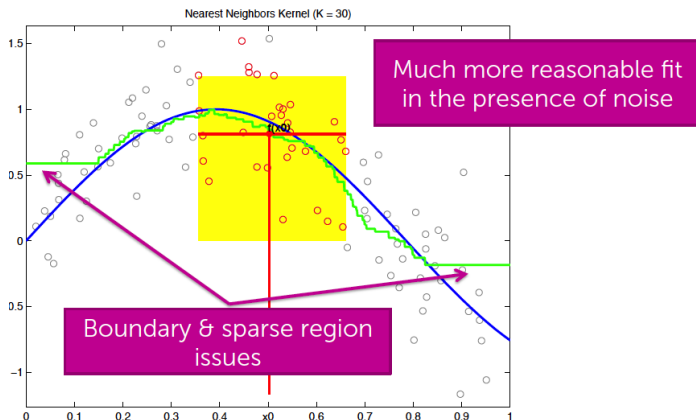
Algoritmo K-NN

$K\text{-NN}(\mathbf{x}^{(q)}, K)$

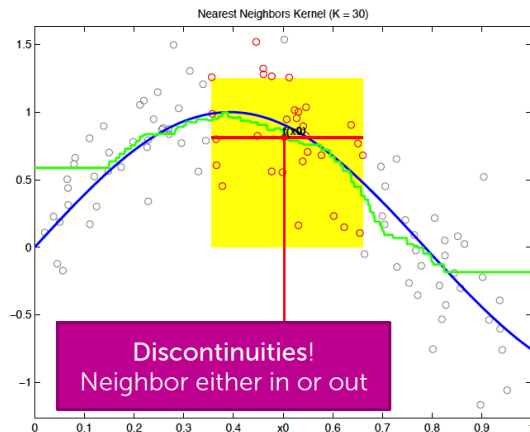
```
1  init Dist2KNN = sort( $\delta$ ),  $\mathbf{K}_{NN}$  = sort( $\mathbf{x}_{NN_1}, \dots, \mathbf{x}_{NN_K}$ )
2  for  $i = K + 1$  to  $N$ 
3       $\delta = d(\mathbf{x}^{(i)}, \mathbf{x}^{(q)})$ 
4      if  $\delta < \mathbf{Dist2KNN}[K]$ 
5          find  $j$  such that  $\delta > \mathbf{Dist2KNN}[j - 1]$  and  $\delta < \mathbf{Dist2KNN}[j]$ 
6           $\mathbf{K}\text{-NN}[j + 1 : K] = \mathbf{K}\text{-NN}[j : K - 1]$ 
7           $\mathbf{Dist2KNN}[j + 1 : K] = \mathbf{Dist2KNN}[j : K - 1]$ 
8           $\mathbf{Dist2KNN}[j] = \delta$ 
9           $\mathbf{K}\text{-NN}[j] = \mathbf{x}^{(i)}$ 
10 return  $\mathbf{K}\text{-NN}$ 
```

Custo para uma predição: $O(N \log N)$

K-NN na prática



K-NN na prática



K-NN com pesos

Ideia: Dar mais importância para pontos mais similares à query.

Predição:

$$\hat{y}^{(q)} = \frac{c_{qNN_1}y_{NN_1} + c_{qNN_2}y_{NN_2} + c_{qNN_3}y_{NN_3} + \dots + c_{qNN_K}y_{NN_K}}{\sum_{j=1}^K c_{qNN_j}}$$

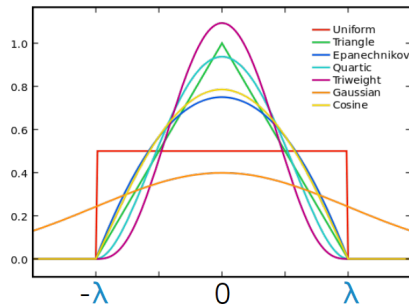
onde c_{qNN_j} é o peso do vizinho NN_j .

Como definir os pesos?

- ▶ Peso c_{qNN_j} deve ser pequeno quando $d(\mathbf{x}_{NN_j}, \mathbf{x}^{(q)})$ for grande.
- ▶ Peso c_{qNN_j} deve ser grande quando $d(\mathbf{x}_{NN_j}, \mathbf{x}^{(q)})$ for pequena.
- ▶ Método simples: $\frac{1}{d(\mathbf{x}^{(q)}, \mathbf{x}^{(j)})}$

Pesos com Kernel

Pesos agora são definidos por Kernels.



Kernel Gaussiano: $K_{\lambda}(d(\mathbf{x}^{(q)}, \mathbf{x}^{(j)})) = \exp(-d(\mathbf{x}^{(q)}, \mathbf{x}^{(j)})^2/\lambda)$

Roteiro

1. Algoritmos dos Vizinhos mais Próximos

3. Regressão Kernel

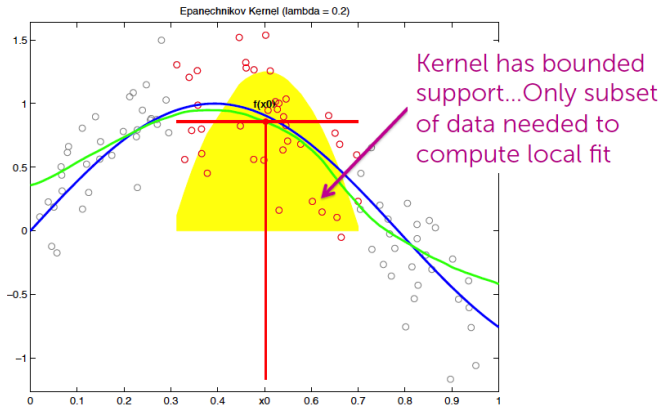
Regressão com Kernels

Em vez de ponderar somente os vizinhos, pondere todos os pontos.

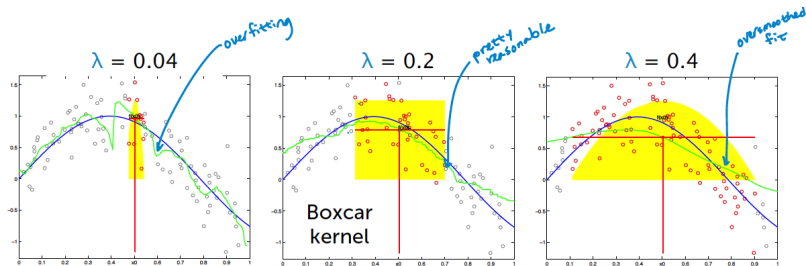
Predição:

$$\hat{y}^{(q)} = \frac{\sum_{i=1}^N c_{qi} y^{(i)}}{\sum_{i=1}^N c_{qi}} = \frac{\sum_{i=1}^N K_{\lambda}(d(\mathbf{x}^{(q)}, \mathbf{x}^{(i)})) y^{(i)}}{\sum_{i=1}^N K_{\lambda}(d(\mathbf{x}^{(q)}, \mathbf{x}^{(i)}))}$$

Regressão Kernel na Prática



A escolha de λ importa mais que a escolha do kernel



Abordagens não paramétricas

- ▶ Objetivos:
 - ▶ Flexibilidade
 - ▶ Poucas hipóteses sobre a função real $f(\mathbf{x})$.
 - ▶ complexidade aumenta com o número de observações.
- ▶ Muitas outras opções:
 - ▶ Splines, árvores, regressão ponderada localmente ...




Limite do KNN em relação aos dados

- ▶ Quando a quantidade de dados (sem ruído) tende a infinito o MSE do 1-NN tende a zero.
- ▶ Quando a quantidade de dados (com ruído) tende a infinito o MSE de NN tende a zero se deixarmos K crescer.

Número de atributos e tamanho dos dados

- ▶ Quanto mais atributos, mais dados para cobrir bem o espaço.
- ▶ Precisa de $N = O(\exp(d))$ observações para bom desempenho.
- ▶ Nesse aspecto modelos paramétricos podem ser úteis.

Referências

-  Pang-Ning Tan, Michael Steinbach, Vipin Kumar. Introduction to Data Mining. Primeira Edição. Addison Wesley, 2006.
-  Lars Schmidt-Thieme. Notas de aula em aprendizagem de máquina. Disponível em: http://www.ismll.uni-hildesheim.de/lehre/ml-11w/index_en.html
-  Brett Lantz. Machine Learning with R. Primeira Edição. Packt, 2013.