Mineração de Dados

Aula 4: Introdução a métodos não paramétricos

Rafael Izbicki

Aula passada

Em muitos problemas reais temos várias covariáveis.

Um modelo de regressão linear simples que usa todas elas pode ter performance preditiva muito ruim, pois com ele é necessário estimar muitos coeficientes, além de que muitas vezes certas covariáveis influenciam muito pouco na variável resposta.

Devemos então selecionar qual o melhor subconjunto de covariáveis para ser utilizado.

Como visto nas aulas anteriores, podemos, para cada modelo, estimar o risco $R(g)$, e buscar aquele com menor risco.

Quando há muitas covariáveis, esse procedimento leva tempo demais.

Aula passada

Em muitos problemas reais temos várias covariáveis.

Um modelo de regressão linear simples que usa todas elas pode ter performance preditiva muito ruim, pois com ele é necessário estimar muitos coeficientes, além de que muitas vezes certas covariáveis influenciam muito pouco na variável resposta.

Devemos então selecionar qual o melhor subconjunto de covariáveis para ser utilizado.

Como visto nas aulas anteriores, podemos, para cada modelo, estimar o risco $R(g)$, e buscar aquele com menor risco.

Quando há muitas covariáveis, esse procedimento leva tempo demais.

Aula passada

Em muitos problemas reais temos várias covariáveis.

Um modelo de regressão linear simples que usa todas elas pode ter performance preditiva muito ruim, pois com ele é necessário estimar muitos coeficientes, além de que muitas vezes certas covariáveis influenciam muito pouco na variável resposta.

Devemos então selecionar qual o melhor subconjunto de covariáveis para ser utilizado.

Como visto nas aulas anteriores, podemos, para cada modelo, estimar o risco $R(g)$, e buscar aquele com menor risco.

Quando há muitas covariáveis, esse procedimento leva tempo demais.

Aula passada

Em muitos problemas reais temos várias covariáveis.

Um modelo de regressão linear simples que usa todas elas pode ter performance preditiva muito ruim, pois com ele é necessário estimar muitos coeficientes, além de que muitas vezes certas covariáveis influenciam muito pouco na variável resposta.

Devemos então selecionar qual o melhor subconjunto de covariáveis para ser utilizado.

Como visto nas aulas anteriores, podemos, para cada modelo, estimar o risco $R(g)$, e buscar aquele com menor risco.

Quando há muitas covariáveis, esse procedimento leva tempo demais.

Aula passada

Em muitos problemas reais temos várias covariáveis.

Um modelo de regressão linear simples que usa todas elas pode ter performance preditiva muito ruim, pois com ele é necessário estimar muitos coeficientes, além de que muitas vezes certas covariáveis influenciam muito pouco na variável resposta.

Devemos então selecionar qual o melhor subconjunto de covariáveis para ser utilizado.

Como visto nas aulas anteriores, podemos, para cada modelo, estimar o risco $R(g)$, e buscar aquele com menor risco.

Quando há muitas covariáveis, esse procedimento leva tempo demais.

Resumindo

Vimos duas alternativas para contornar esse problema.

Método 1: Usamos uma heurística para buscar o melhor modelo. Aqui, estudamos o forward stepwise, no qual começamos avaliando o risco do modelo que só tem intercepto, e então passamos a tentar inserir apenar uma covariável por vez. Buscamos sempre a variável que minimiza o risco estimado.

Resumindo

Vimos duas alternativas para contornar esse problema.

Método 1: Usamos uma heurística para buscar o melhor modelo. Aqui, estudamos o forward stepwise, no qual começamos avaliando o risco do modelo que só tem intercepto, e então passamos a tentar inserir apenar uma covariável por vez. Buscamos sempre a variável que minimiza o risco estimado.

Resumindo

Método 2: Mudamos a penalização usada. Aqui vimos o lasso, que ao invés de buscar

$$\arg\min_{\beta,S} \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_1 - \ldots - \beta_p x_p)^2 + \lambda \sum_{j=1}^{p} \mathbb{I}(\beta_j \neq 0),$$

buscamos por

$$\arg\min_{\beta} \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_1 - \ldots - \beta_p x_p)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Balanço entre EQM baixo e coeficientes pequenos.

λ em geral é escolhido por validação cruzada.

Resumindo

Método 2: Mudamos a penalização usada. Aqui vimos o lasso, que ao invés de buscar

$$\arg\min_{\beta,S} \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_1 - \ldots - \beta_p x_p)^2 + \lambda \sum_{j=1}^{p} \mathbb{I}(\beta_j \neq 0),$$

buscamos por

$$\arg\min_{\beta} \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_1 - \ldots - \beta_p x_p)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Balanço entre EQM baixo e coeficientes pequenos.

λ em geral é escolhido por validação cruzada.

Resumindo

Método 2: Mudamos a penalização usada. Aqui vimos o lasso, que ao invés de buscar

$$\arg\min_{\beta,S} \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_1 - \ldots - \beta_p x_p)^2 + \lambda \sum_{j=1}^{p} \mathbb{I}(\beta_j \neq 0),$$

buscamos por

$$\arg\min_{\beta} \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_1 - \ldots - \beta_p x_p)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Balanço entre EQM baixo e coeficientes pequenos.

λ em geral é escolhido por validação cruzada.

- É extremamente rápido achar $\widehat{\beta}$ para todos os λ simultaneamente (algoritmo lars)

- A solução induzida por ele possui muitos zeros (i.e., o vetor $\widehat{\beta}$ é esparço). Assim, o modelo resultante é de fácil interpretação.

O balanço viés-variância

$$\mathbb{E}\left[(Y - \widehat{g}(\mathbf{X}))^2 \middle| \mathbf{X} = \mathbf{x}\right]$$

- $\mathbb{V}[Y|\mathbf{X} = \mathbf{x}]$ é a variância intrínseca da variável resposta, que não depende da função $\widehat{g}$, assim não pode ser reduzida.
- $(r(\mathbf{x}) - \mathbb{E}[\widehat{g}(\mathbf{x})])^2$ é o viés do estimador $\widehat{g}$
- $\mathbb{V}[\widehat{g}(\mathbf{x})]$ é sua variância

O balanço viés-variância

$$\mathbb{E}\left[(Y - \widehat{g}(\mathbf{X}))^2 \middle| \mathbf{X} = \mathbf{x}\right] = \mathbb{V}[Y|\mathbf{X} = \mathbf{x}] + (r(\mathbf{x}) - \mathbb{E}[\widehat{g}(\mathbf{x})])^2 + \mathbb{V}[\widehat{g}(\mathbf{x})]$$

- $\mathbb{V}[Y|\mathbf{X} = \mathbf{x}]$ é a variância intrínseca da variável resposta, que não depende da função $\widehat{g}$, assim não pode ser reduzida.
- $(r(\mathbf{x}) - \mathbb{E}[\widehat{g}(\mathbf{x})])^2$ é o viés do estimador $\widehat{g}$
- $\mathbb{V}[\widehat{g}(\mathbf{x})]$ é sua variância

O balanço viés-variância

$$\mathbb{E}\left[(Y - \widehat{g}(\mathbf{X}))^2 \middle| \mathbf{X} = \mathbf{x}\right] = \mathbb{V}[Y|\mathbf{X} = \mathbf{x}] + (r(\mathbf{x}) - \mathbb{E}[\widehat{g}(\mathbf{x})])^2 + \mathbb{V}[\widehat{g}(\mathbf{x})]$$

- $\mathbb{V}[Y|\mathbf{X} = \mathbf{x}]$ é a variância intrínseca da variável resposta, que não depende da função $\widehat{g}$, assim não pode ser reduzida.
- $(r(\mathbf{x}) - \mathbb{E}[\widehat{g}(\mathbf{x})])^2$ é o viés do estimador $\widehat{g}$
- $\mathbb{V}[\widehat{g}(\mathbf{x})]$ é sua variância

O balanço viés-variância

$$\mathbb{E}\left[(Y - \widehat{g}(\mathbf{X}))^2 \middle| \mathbf{X} = \mathbf{x}\right] = \mathbb{V}[Y|\mathbf{X} = \mathbf{x}] + (r(\mathbf{x}) - \mathbb{E}[\widehat{g}(\mathbf{x})])^2 + \mathbb{V}[\widehat{g}(\mathbf{x})]$$

- $\mathbb{V}[Y|\mathbf{X} = \mathbf{x}]$ é a variância intrínseca da variável resposta, que não depende da função $\widehat{g}$, assim não pode ser reduzida.
- $(r(\mathbf{x}) - \mathbb{E}[\widehat{g}(\mathbf{x})])^2$ é o viés do estimador $\widehat{g}$
- $\mathbb{V}[\widehat{g}(\mathbf{x})]$ é sua variância

O balanço viés-variância

$$\mathbb{E}\left[(Y - \widehat{g}(\mathbf{X}))^2 \middle| \mathbf{X} = \mathbf{x}\right] = \mathbb{V}[Y|\mathbf{X} = \mathbf{x}] + (r(\mathbf{x}) - \mathbb{E}[\widehat{g}(\mathbf{x})])^2 + \mathbb{V}[\widehat{g}(\mathbf{x})]$$

- $\mathbb{V}[Y|\mathbf{X} = \mathbf{x}]$ é a variância intrínseca da variável resposta, que não depende da função $\widehat{g}$, assim não pode ser reduzida.
- $(r(\mathbf{x}) - \mathbb{E}[\widehat{g}(\mathbf{x})])^2$ é o viés do estimador $\widehat{g}$
- $\mathbb{V}[\widehat{g}(\mathbf{x})]$ é sua variância

Nesta aula: Introdução a métodos não paramétricos

Até agora, nos restringimos a métodos *paramétricos* para estimar $r(x)$.

Em outras palavras, nossos estimadores sempre possuiam um número finito de parâmetros a serem estimados

Ex:

$$r(x) = \beta_0 + \beta_1 x$$

$$r(x) = \beta_0 + \beta_1 x + \beta_2 x^2$$

$$r(x) = \beta_0 + \beta_1 x_1 + \beta_1 x_2$$

Métodos paramétricos muitas vezes são muito restritivos e simplistas para criar boas funções de predição.

Ex: a relação entre duas variáveis não é bem descrita por uma relação linear

Até agora, nos restringimos a métodos *paramétricos* para estimar $r(x)$.

Em outras palavras, nossos estimadores sempre possuiam um número finito de parâmetros a serem estimados

Ex:

$$r(x) = \beta_0 + \beta_1 x$$

$$r(x) = \beta_0 + \beta_1 x + \beta_2 x^2$$

$$r(x) = \beta_0 + \beta_1 x_1 + \beta_1 x_2$$

Métodos paramétricos muitas vezes são muito restritivos e simplistas para criar boas funções de predição.

Ex: a relação entre duas variáveis não é bem descrita por uma relação linear

Até agora, nos restringimos a métodos *paramétricos* para estimar $r(x)$.

Em outras palavras, nossos estimadores sempre possuiam um número finito de parâmetros a serem estimados

Ex:

$$r(x) = \beta_0 + \beta_1 x$$

$$r(x) = \beta_0 + \beta_1 x + \beta_2 x^2$$

$$r(x) = \beta_0 + \beta_1 x_1 + \beta_1 x_2$$

Métodos paramétricos muitas vezes são muito restritivos e simplistas para criar boas funções de predição.

Ex: a relação entre duas variáveis não é bem descrita por uma relação linear

Até agora, nos restringimos a métodos *paramétricos* para estimar $r(x)$.

Em outras palavras, nossos estimadores sempre possuiam um número finito de parâmetros a serem estimados

Ex:

$$r(x) = \beta_0 + \beta_1 x$$

$$r(x) = \beta_0 + \beta_1 x + \beta_2 x^2$$

$$r(x) = \beta_0 + \beta_1 x_1 + \beta_1 x_2$$

Métodos paramétricos muitas vezes são muito restritivos e simplistas para criar boas funções de predição.

Ex: a relação entre duas variáveis não é bem descrita por uma relação linear

Até agora, nos restringimos a métodos *paramétricos* para estimar $r(x)$.

Em outras palavras, nossos estimadores sempre possuiam um número finito de parâmetros a serem estimados

Ex:

$$r(x) = \beta_0 + \beta_1 x$$

$$r(x) = \beta_0 + \beta_1 x + \beta_2 x^2$$

$$r(x) = \beta_0 + \beta_1 x_1 + \beta_1 x_2$$

Métodos paramétricos muitas vezes são muito restritivos e simplistas para criar boas funções de predição.

Ex: a relação entre duas variáveis não é bem descrita por uma relação linear

k Vizinhos Mais Próximos (KNN)

Método dos k Vizinhos Mais Próximos

Benedetti, 1977 e Stone, 1977; muito popular em aprendizado de máquina: KNN

Motivação muito diferente: médias locais

Método dos k Vizinhos Mais Próximos

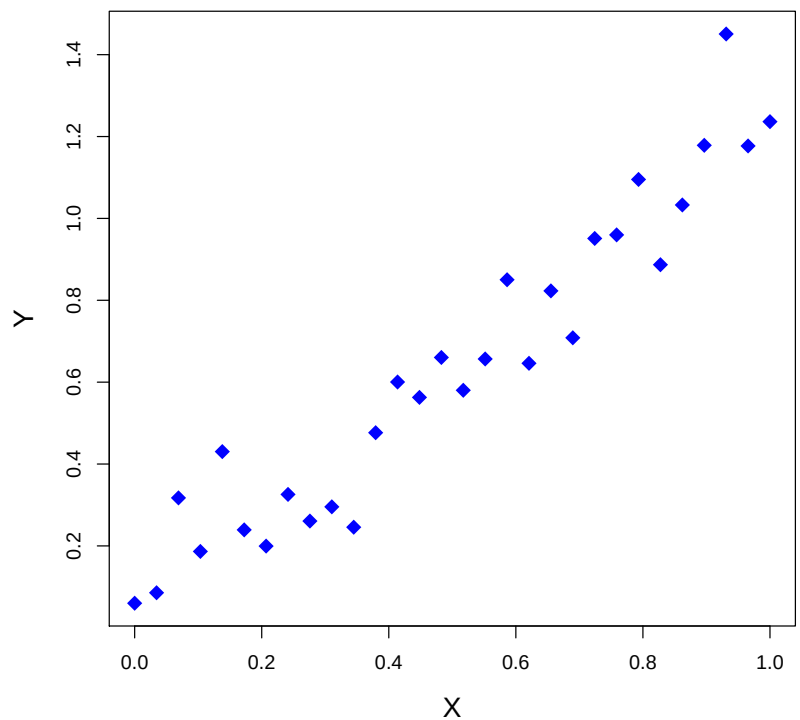
Benedetti, 1977 e Stone, 1977; muito popular em aprendizado de máquina: KNN

Motivação muito diferente: médias locais

Método dos k Vizinhos Mais Próximos

Benedetti, 1977 e Stone, 1977; muito popular em aprendizado de máquina: KNN

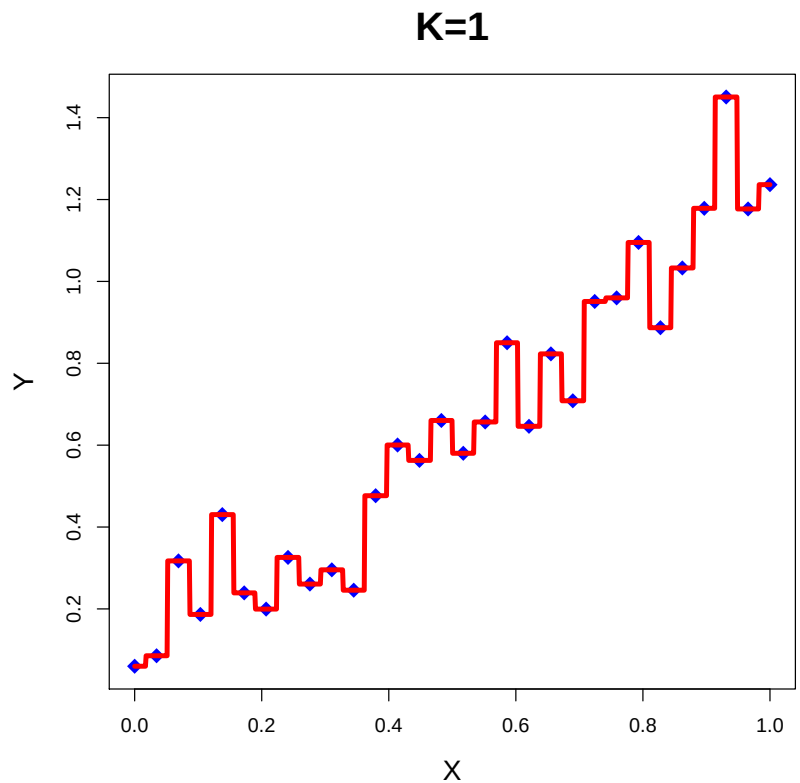
Motivação muito diferente: médias locais



Método dos k Vizinhos Mais Próximos

Benedetti, 1977 e Stone, 1977; muito popular em aprendizado de máquina: KNN

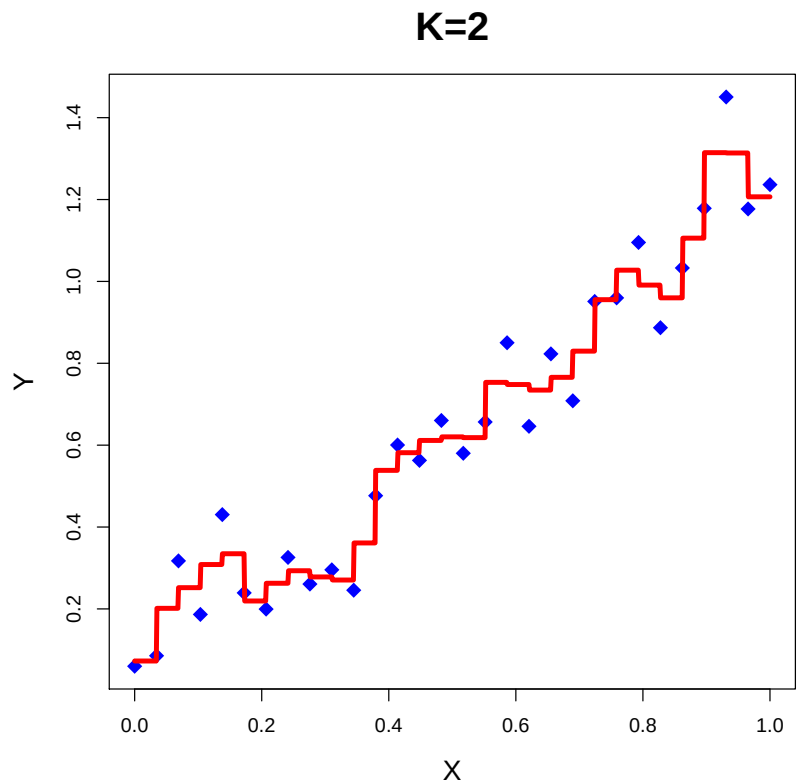
Motivação muito diferente: médias locais



Método dos k Vizinhos Mais Próximos

Benedetti, 1977 e Stone, 1977; muito popular em aprendizado de máquina: KNN

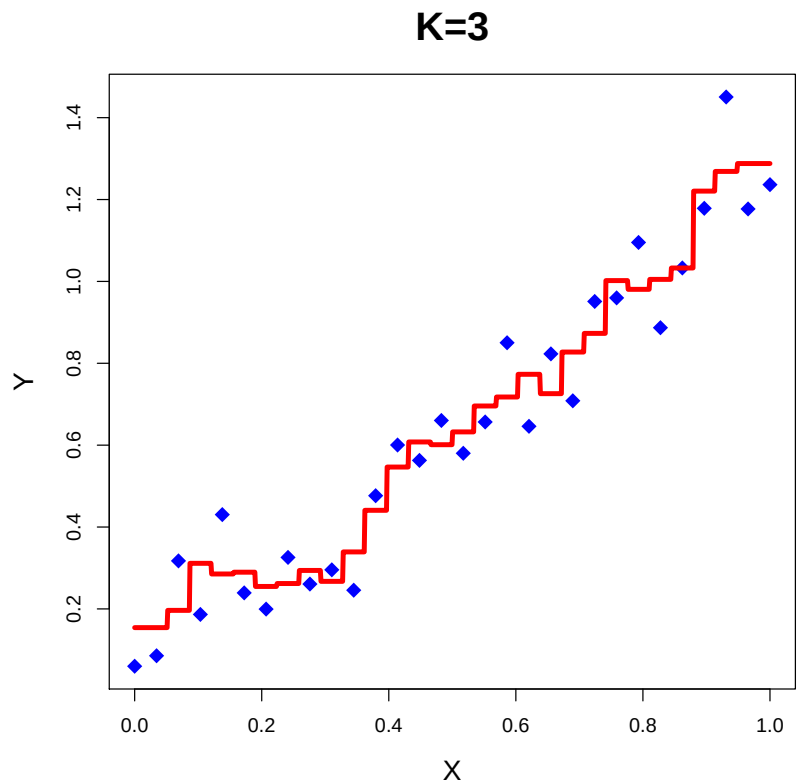
Motivação muito diferente: médias locais



Método dos k Vizinhos Mais Próximos

Benedetti, 1977 e Stone, 1977; muito popular em aprendizado de máquina: KNN

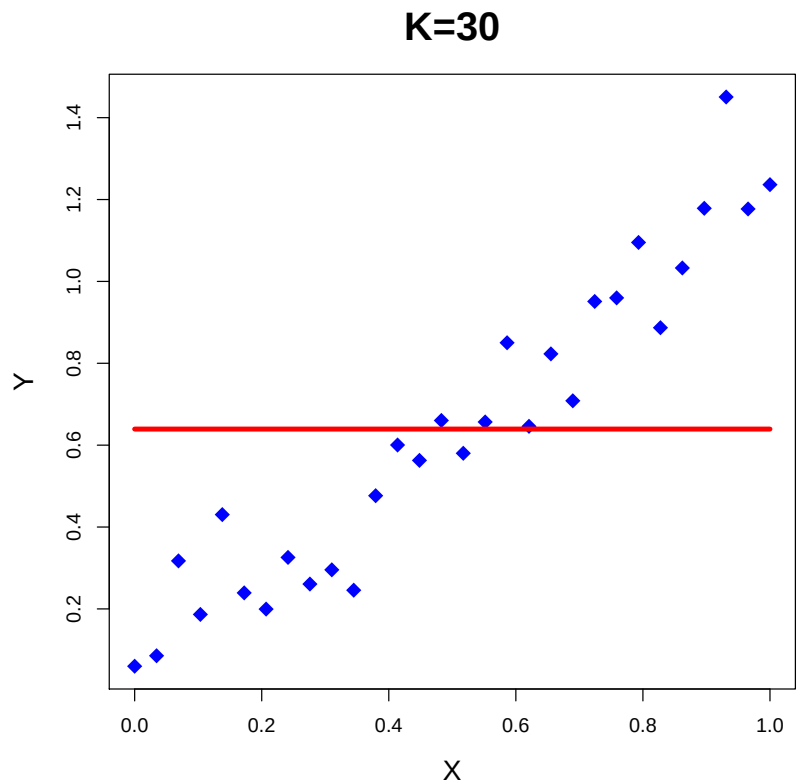
Motivação muito diferente: médias locais



Método dos k Vizinhos Mais Próximos

Benedetti, 1977 e Stone, 1977; muito popular em aprendizado de máquina: KNN

Motivação muito diferente: médias locais



Formalmente:

$$\widehat{r}(\mathbf{x}) = \frac{1}{K} \sum_{\mathbf{x}_i \in \mathcal{N}_\mathbf{x}} y_i,$$

onde $\mathcal{N}_\mathbf{x}$ é o conjunto das k observações mais próximas de $\mathbf{x}$, i.e.,

$$\mathcal{N}_\mathbf{x} = \left\{ i \in \{1, \ldots, n\} : d(\mathbf{x}_i, \mathbf{x}) \leq d_\mathbf{x}^k \right\}$$

$d_\mathbf{x}^k$: distância do k-ésimo vizinho mais próximo de $\mathbf{x}$ a $\mathbf{x}$.

Usualmente utilizamos a distância Euclidiana.

Formalmente:

$$\widehat{r}(\mathbf{x}) = \frac{1}{K} \sum_{\mathbf{x}_i \in \mathcal{N}_\mathbf{x}} y_i,$$

onde $\mathcal{N}_\mathbf{x}$ é o conjunto das k observações mais próximas de $\mathbf{x}$, i.e.,

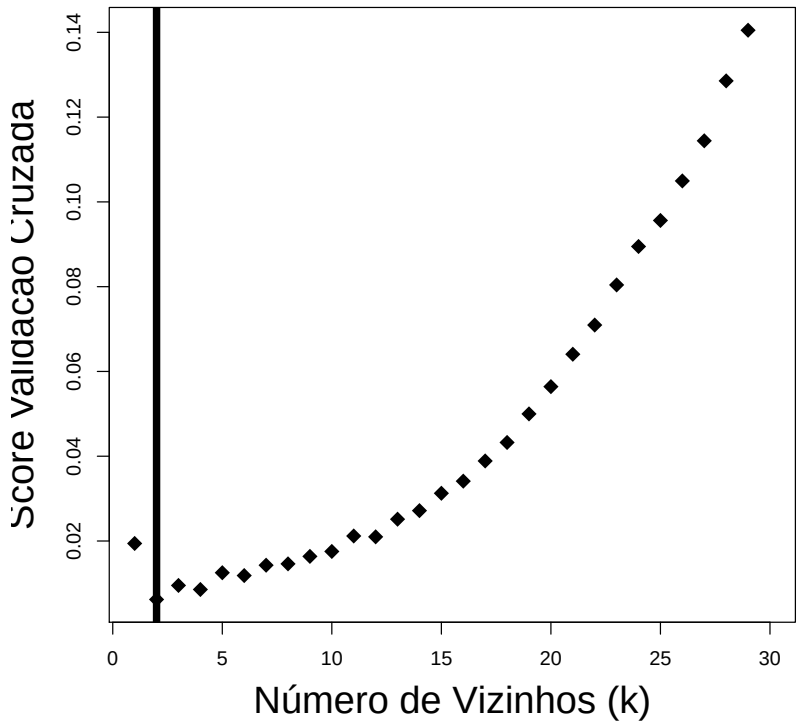
$$\mathcal{N}_\mathbf{x} = \left\{ i \in \{1, \ldots, n\} : d(\mathbf{x}_i, \mathbf{x}) \leq d_\mathbf{x}^k \right\}$$

$d_\mathbf{x}^k$: distância do k-ésimo vizinho mais próximo de $\mathbf{x}$ a $\mathbf{x}$.

Usualmente utilizamos a distância Euclidiana.

Como escolher k? Validação cruzada!

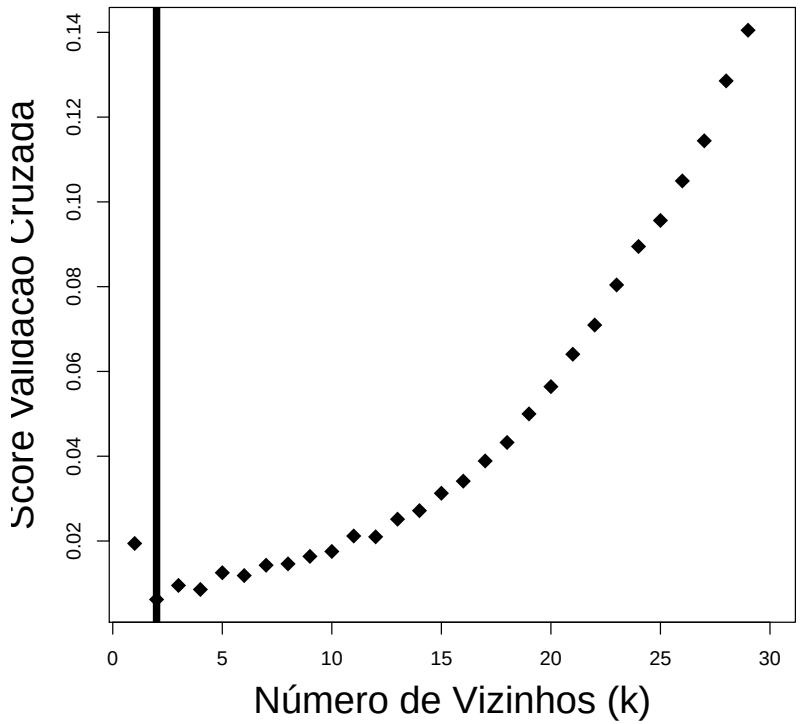
Qual se papel no balanço viés-variância?



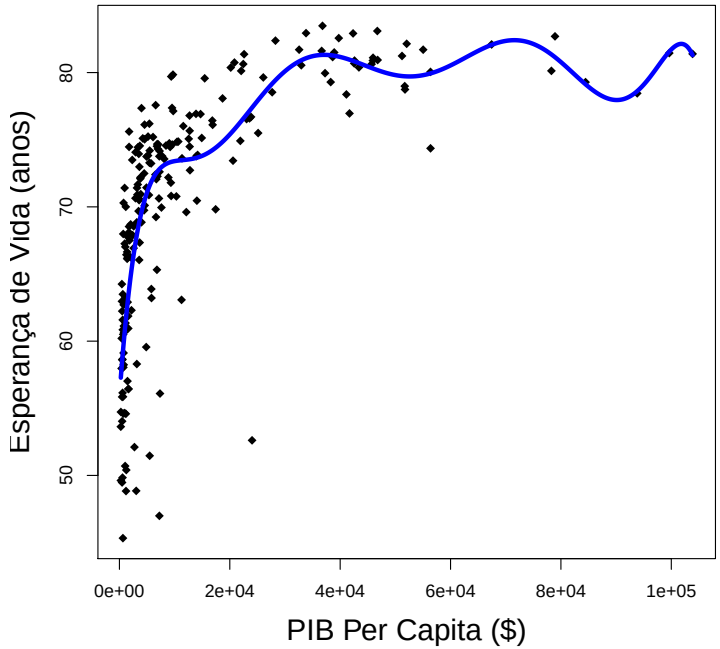
k=2

Como escolher k? Validação cruzada!

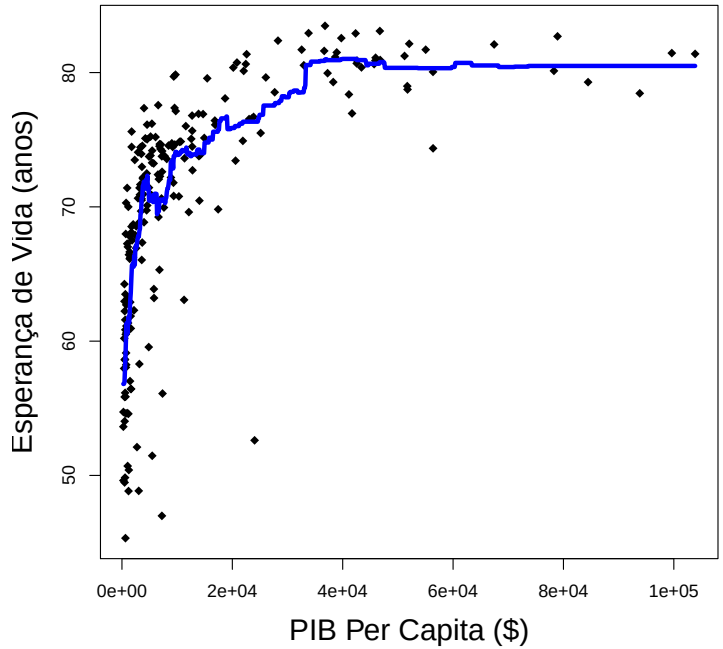
Qual se papel no balanço viés-variância?



k=2



(a) Séries

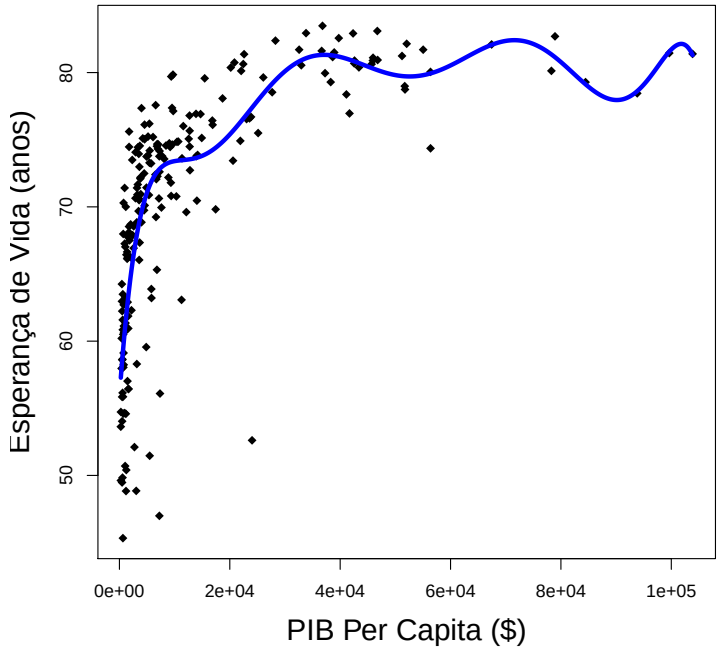


(b) KNN

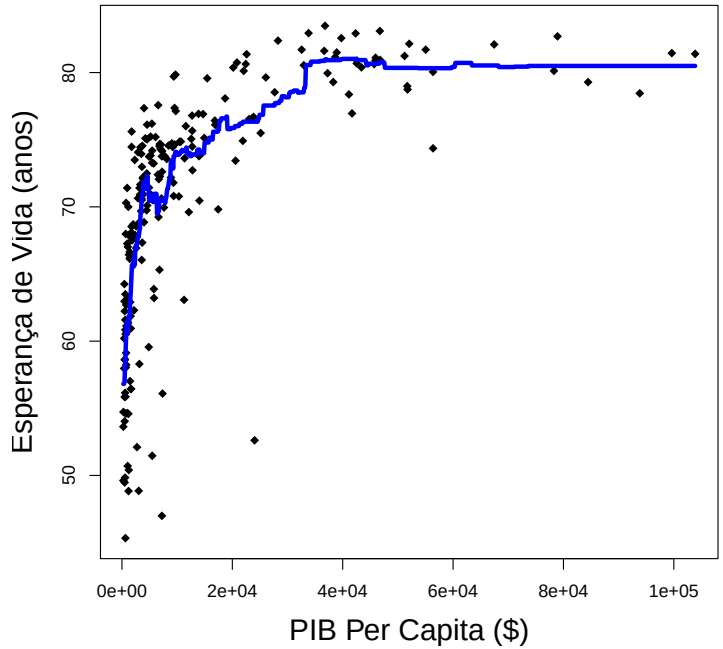
Riscos estimados: 31.15 e 31.45

KNN é simples e frequentemente funciona muito bem.

Contudo, é necessário guardar todas as observações para se fazer predições



(c) Séries

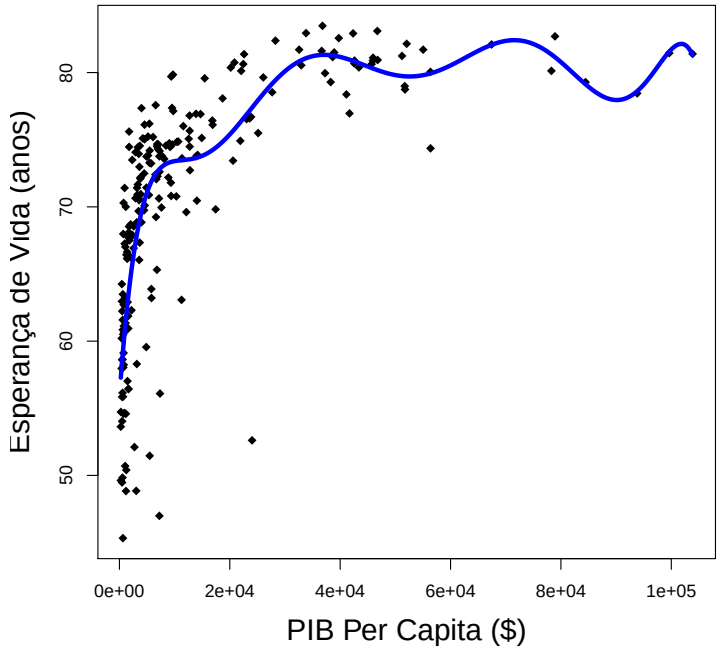


(d) KNN

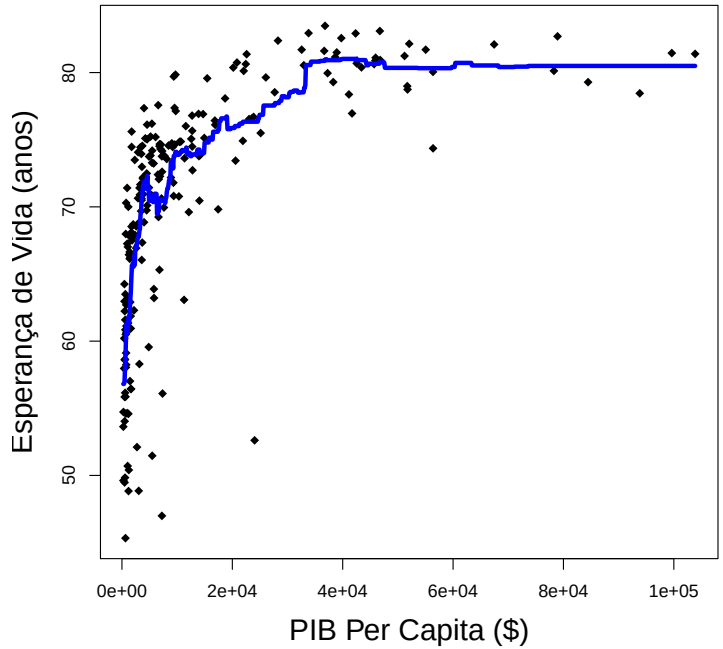
Riscos estimados: 31.15 e 31.45

KNN é simples e frequentemente funciona muito bem.

Contudo, é necessário guardar todas as observações para se fazer predições



(e) Séries

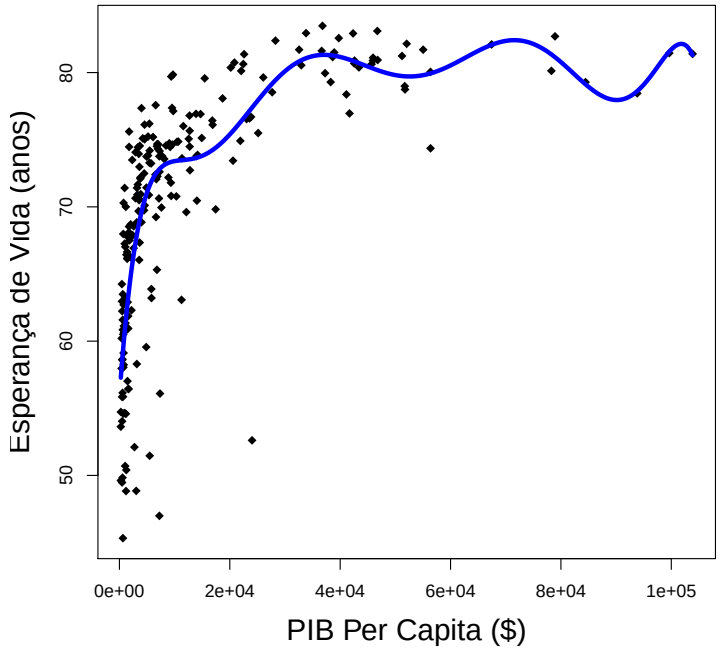


(f) KNN

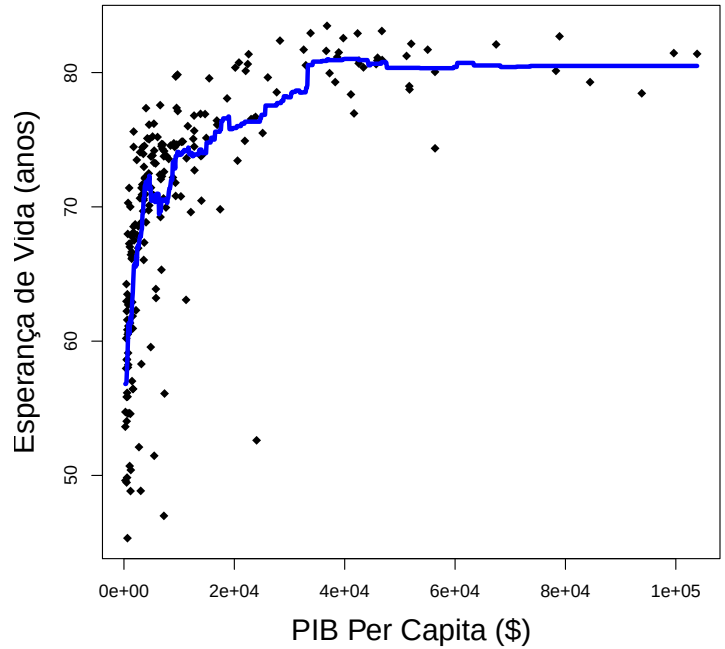
Riscos estimados: 31.15 e 31.45

KNN é simples e frequentemente funciona muito bem.

Contudo, é necessário guardar todas as observações para se fazer predições



(g) Séries



(h) KNN

Riscos estimados: 31.15 e 31.45

KNN é simples e frequentemente funciona muito bem.

Contudo, é necessário guardar todas as observações para se fazer predições

Nadaraya-Watson

Nadaraya-Watson

Nadaraya (1964) e Watson (1964). Variação do KNN.

Ideia: atribuir um peso a cada observação

Nadaraya-Watson

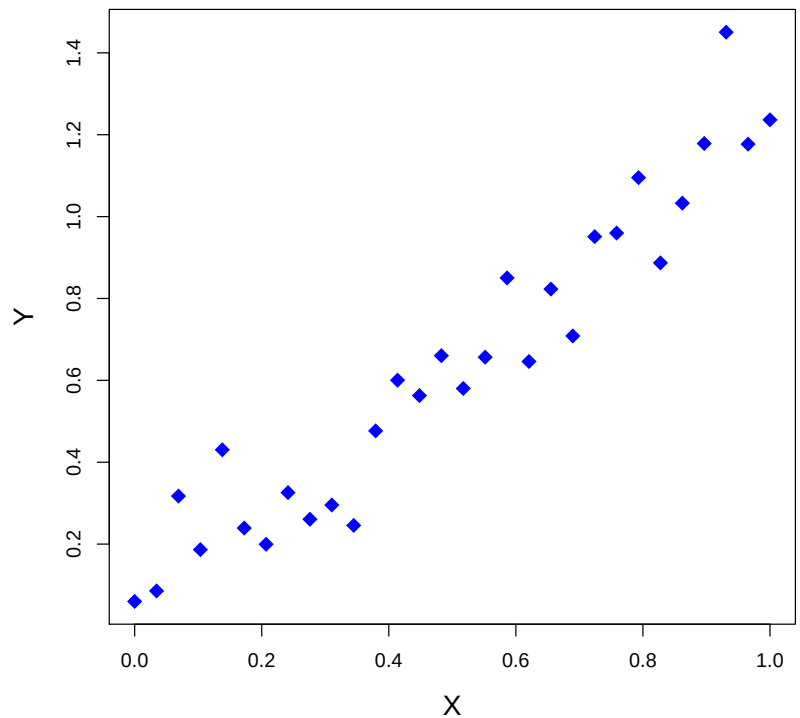
Nadaraya (1964) e Watson (1964). Variação do KNN.

Ideia: atribuir um peso a cada observação

Nadaraya-Watson

Nadaraya (1964) e Watson (1964). Variação do KNN.

Ideia: atribuir um peso a cada observação



Nadaraya-Watson

Formalmente:

$$\widehat{r}(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x}) y_i,$$

$$w_i(\mathbf{x}) = \frac{K(\mathbf{x}, \mathbf{x}_i)}{\sum_{j=1}^{n} K(\mathbf{x}, \mathbf{x}_i)},$$

onde $K(\mathbf{x}, \mathbf{x}_i)$ é um kernel usado para medir a similaridade entre as observações.

- $K(\mathbf{x}, \mathbf{x}_i) = \mathbb{I}(d(\mathbf{x}, \mathbf{x}_i) \leq h)$ (kernel uniforme)
- $K(\mathbf{x}, \mathbf{x}_i) = (\sqrt{2\pi h^2})^{-1} exp\left\{ -\frac{d^2(\mathbf{x}, \mathbf{x}_i)}{2h^2} \right\}$ (kernel Gaussiano)

Nadaraya-Watson

Formalmente:

$$\widehat{r}(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x}) y_i,$$

$$w_i(\mathbf{x}) = \frac{K(\mathbf{x}, \mathbf{x}_i)}{\sum_{j=1}^{n} K(\mathbf{x}, \mathbf{x}_i)},$$

onde $K(\mathbf{x}, \mathbf{x}_i)$ é um kernel usado para medir a similaridade entre as observações.

- $K(\mathbf{x}, \mathbf{x}_i) = \mathbb{I}(d(\mathbf{x}, \mathbf{x}_i) \leq h)$ (kernel uniforme)
- $K(\mathbf{x}, \mathbf{x}_i) = (\sqrt{2\pi h^2})^{-1} exp\left\{ -\frac{d^2(\mathbf{x}, \mathbf{x}_i)}{2h^2} \right\}$ (kernel Gaussiano)

Nadaraya-Watson

Formalmente:

$$\widehat{r}(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x}) y_i,$$

$$w_i(\mathbf{x}) = \frac{K(\mathbf{x}, \mathbf{x}_i)}{\sum_{j=1}^{n} K(\mathbf{x}, \mathbf{x}_i)},$$

onde $K(\mathbf{x}, \mathbf{x}_i)$ é um kernel usado para medir a similaridade entre as observações.

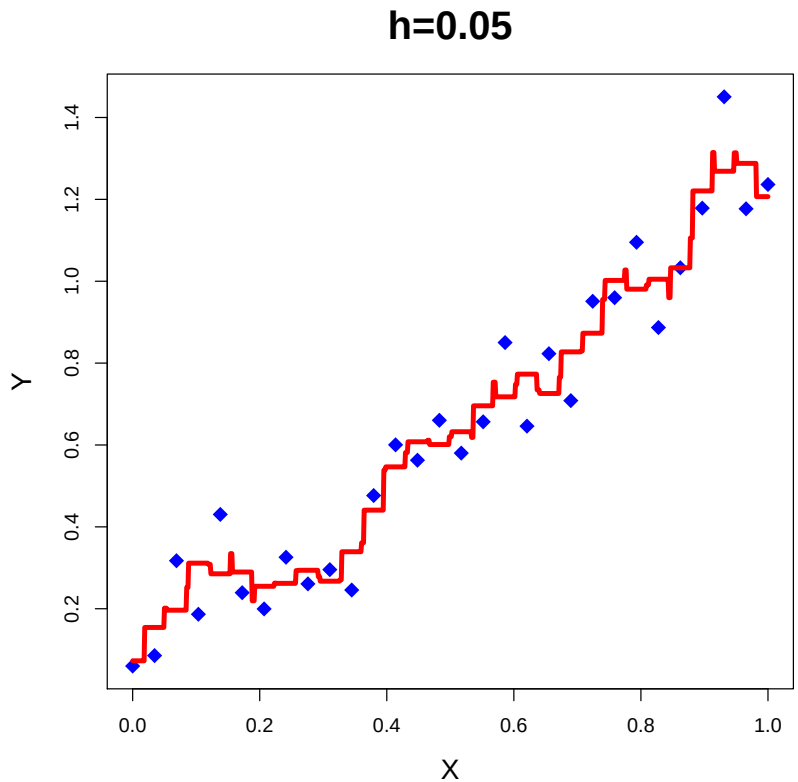
- $K(\mathbf{x}, \mathbf{x}_i) = \mathbb{I}(d(\mathbf{x}, \mathbf{x}_i) \leq h)$ (kernel uniforme)
- $K(\mathbf{x}, \mathbf{x}_i) = (\sqrt{2\pi h^2})^{-1} exp\left\{ -\frac{d^2(\mathbf{x}, \mathbf{x}_i)}{2h^2} \right\}$ (kernel Gaussiano)

Nadaraya-Watson

Kernel Uniforme: $K(\mathbf{x}, \mathbf{x}_i) = \mathbb{I}(d(\mathbf{x}, \mathbf{x}_i) \leq h)$

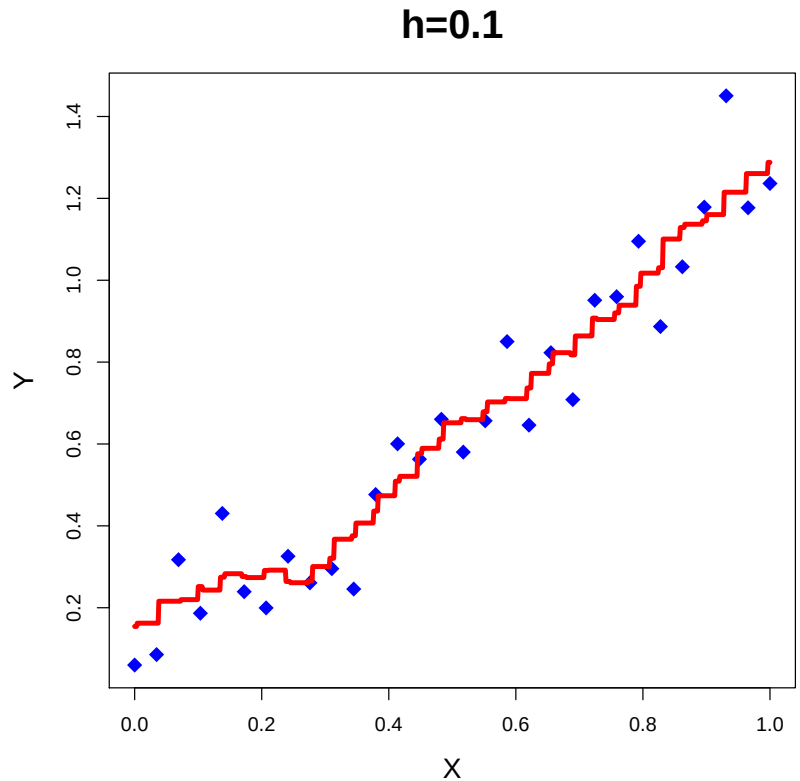
Nadaraya-Watson

Kernel Uniforme: $K(\mathbf{x}, \mathbf{x}_i) = \mathbb{I}(d(\mathbf{x}, \mathbf{x}_i) \leq h)$



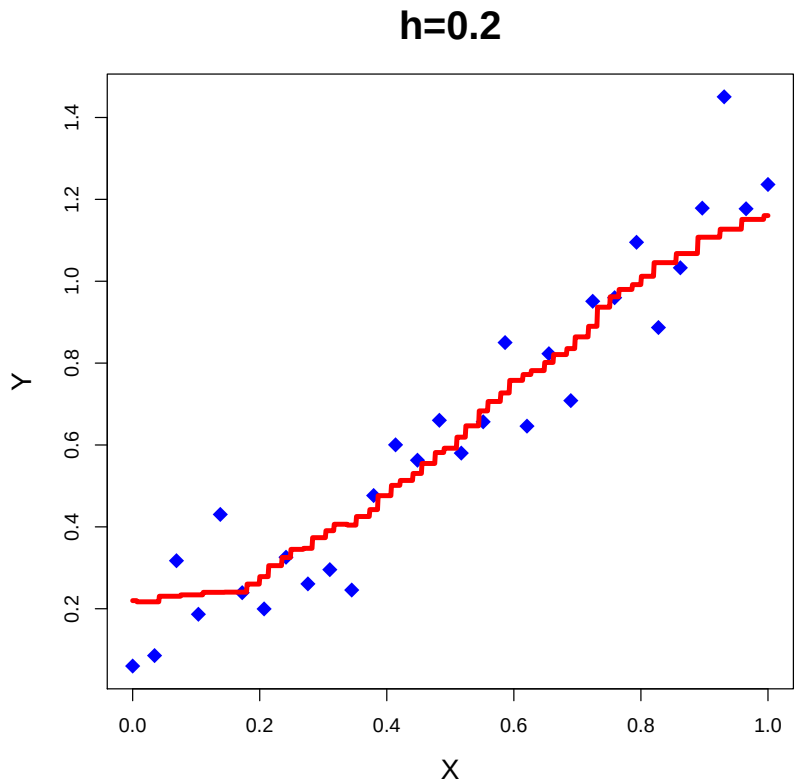
Nadaraya-Watson

Kernel Uniforme: $K(\mathbf{x}, \mathbf{x}_i) = \mathbb{I}(d(\mathbf{x}, \mathbf{x}_i) \leq h)$



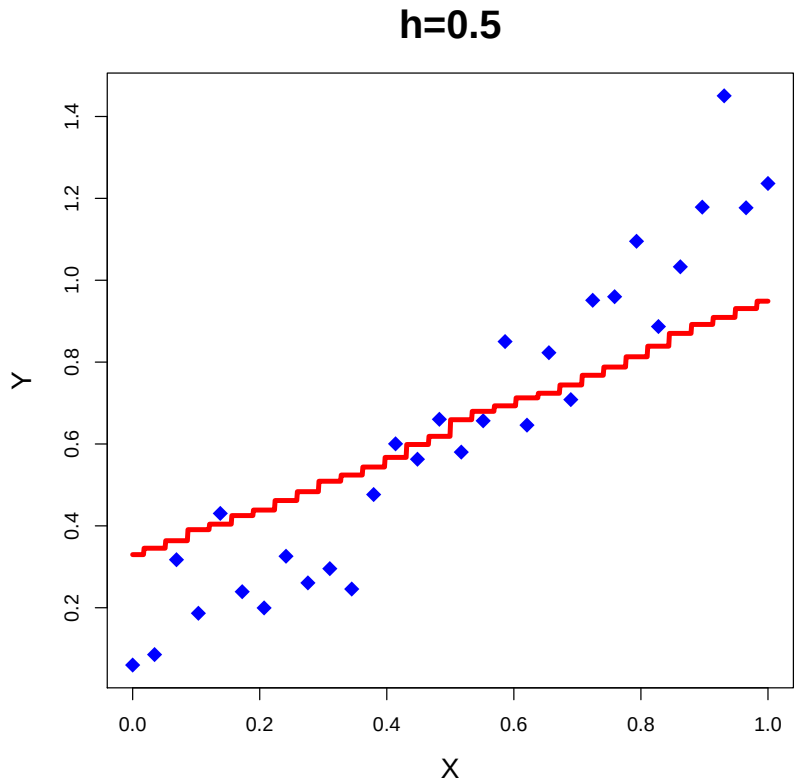
Nadaraya-Watson

Kernel Uniforme: $K(\mathbf{x}, \mathbf{x}_i) = \mathbb{I}(d(\mathbf{x}, \mathbf{x}_i) \leq h)$



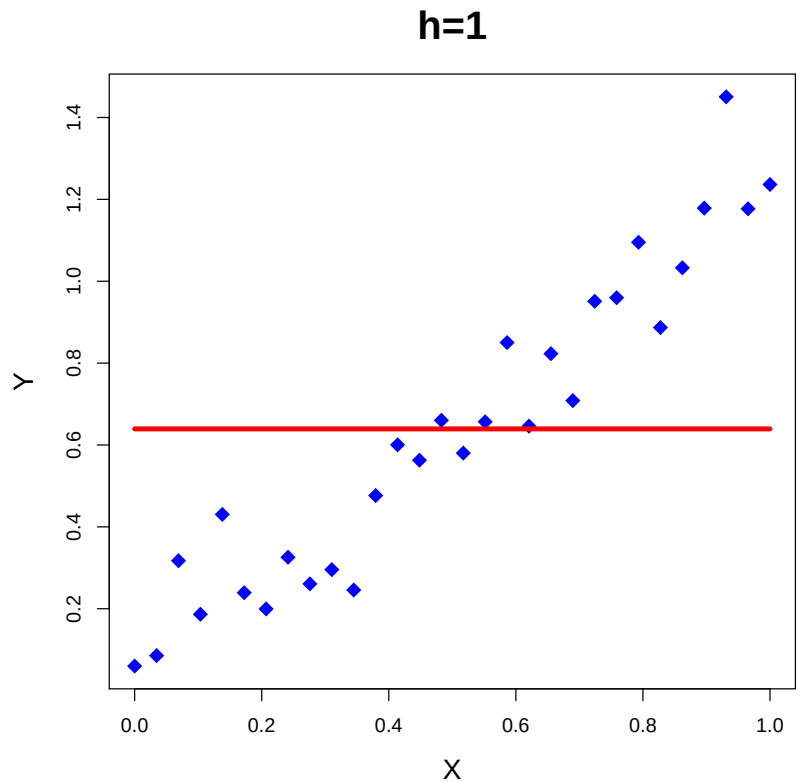
Nadaraya-Watson

Kernel Uniforme: $K(\mathbf{x}, \mathbf{x}_i) = \mathbb{I}(d(\mathbf{x}, \mathbf{x}_i) \leq h)$



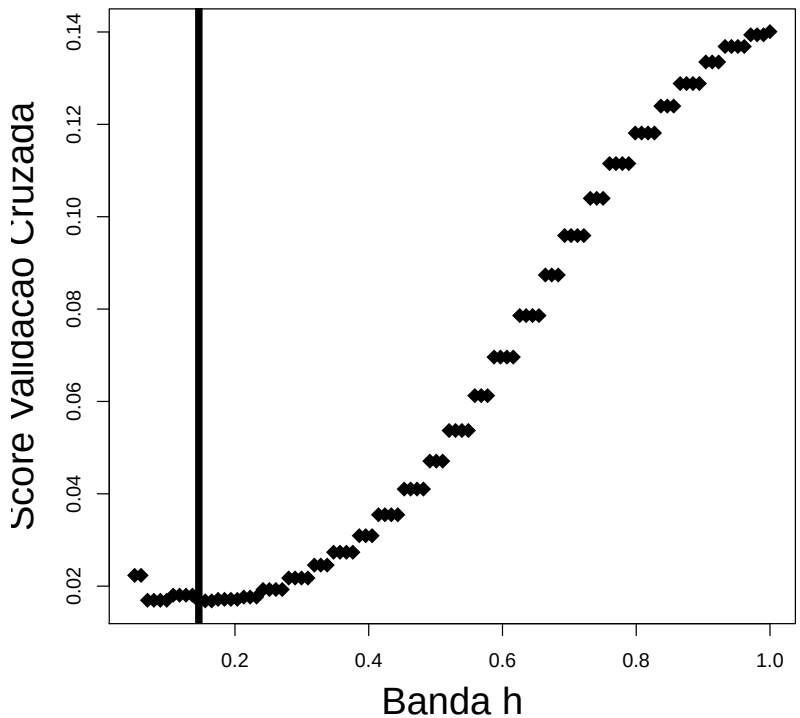
Nadaraya-Watson

Kernel Uniforme: $K(\mathbf{x}, \mathbf{x}_i) = \mathbb{I}(d(\mathbf{x}, \mathbf{x}_i) \leq h)$

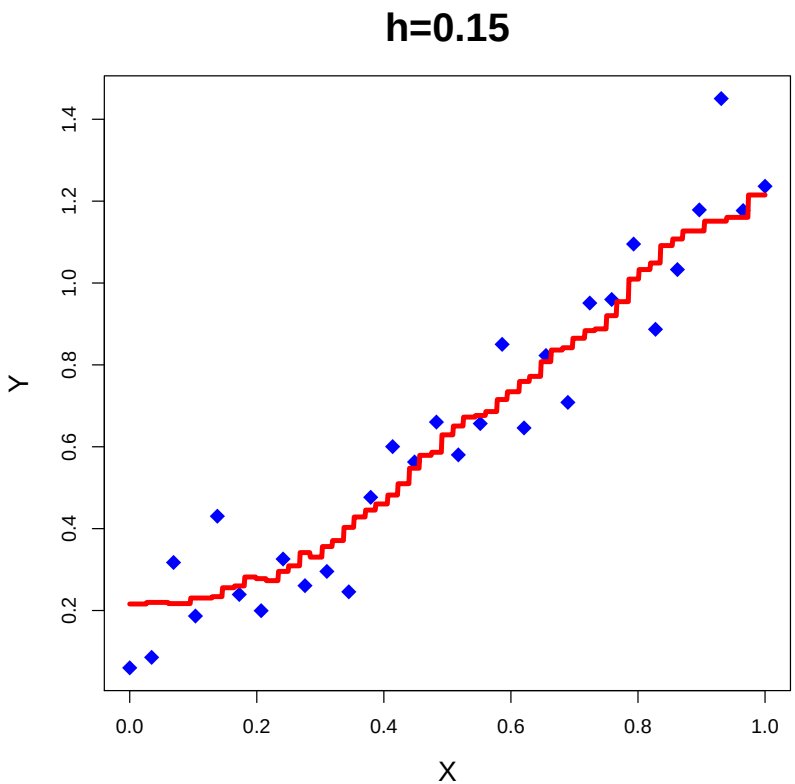


Nadaraya-Watson

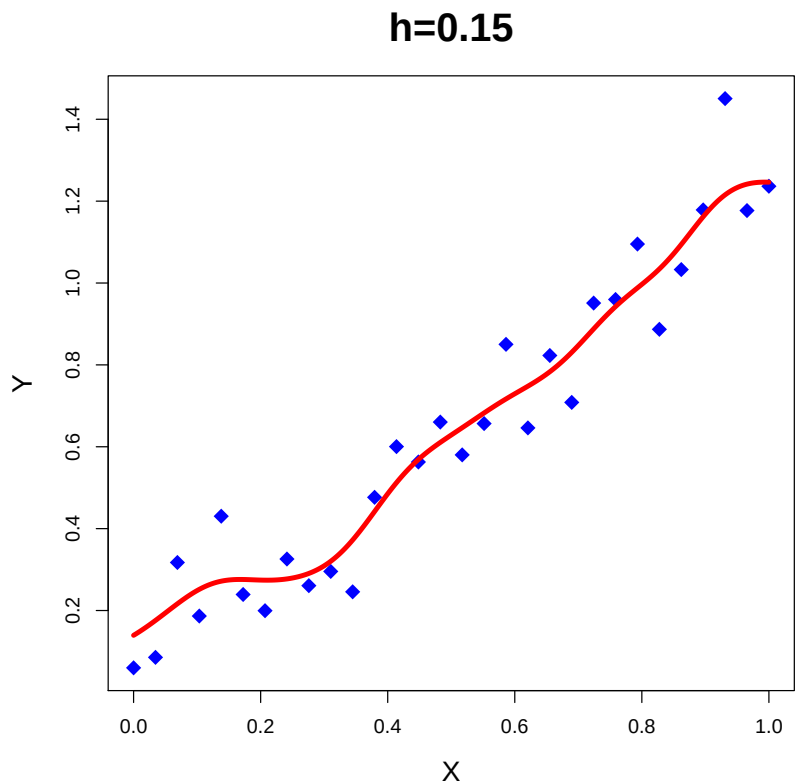
h: banda; balanço viés-variância



Nadaraya-Watson



(a) Kernel uniforme



(b) Kernel Gaussiano

Tipicamente o kernel não influencia muito

Regressão polinomial

Conhecido de vocês.

$$r(x) = \sum_{i=0}^{\infty} \beta_i x^i$$

Número infinito de parâmetros $\Rightarrow$ é um método não paramétrico.

Na prática temos que usar $r(x) = \sum_{i=0}^{I} \beta_i x^i$

Escolhemos I por validação cruzada.

Conhecido de vocês.

$$r(x) = \sum_{i=0}^{\infty} \beta_i x^i$$

Número infinito de parâmetros $\Rightarrow$ é um método não paramétrico.

Na prática temos que usar $r(x) = \sum_{i=0}^{I} \beta_i x^i$

Escolhemos I por validação cruzada.

Conhecido de vocês.

$$r(x) = \sum_{i=0}^{\infty} \beta_i x^i$$

Número infinito de parâmetros $\Rightarrow$ é um método não paramétrico.

Na prática temos que usar $r(x) = \sum_{i=0}^{I} \beta_i x^i$

Escolhemos I por validação cruzada.

Conhecido de vocês.

$$r(x) = \sum_{i=0}^{\infty} \beta_i x^i$$

Número infinito de parâmetros $\Rightarrow$ é um método não paramétrico.

Na prática temos que usar $r(x) = \sum_{i=0}^{I} \beta_i x^i$

Escolhemos I por validação cruzada.

Conhecido de vocês.

$$r(x) = \sum_{i=0}^{\infty} \beta_i x^i$$

Número infinito de parâmetros $\Rightarrow$ é um método não paramétrico.

Na prática temos que usar $r(x) = \sum_{i=0}^{I} \beta_i x^i$

Escolhemos I por validação cruzada.

Importante

Notação: quantidades como l (regressão polinomial), k (nearest neighbors), h (Nadaraya-Watson), λ (lasso) são chamados de tuning parameters.

Em geral, eles controlam o tradeoff entre viés e variância.

Ex:

l grande: viés baixo, variância alta; l pequeno: viés grande, variância pequena

λ pequeno: viés baixo, variância alta; λ grande: viés grande, variância pequena

Importante

Notação: quantidades como l (regressão polinomial), k (nearest neighbors), h (Nadaraya-Watson), λ (lasso) são chamados de tuning parameters.

Em geral, eles controlam o tradeoff entre viés e variância.

Ex:

l grande: viés baixo, variância alta; l pequeno: viés grande, variância pequena

λ pequeno: viés baixo, variância alta; λ grande: viés grande, variância pequena

Importante

Notação: quantidades como l (regressão polinomial), k (nearest neighbors), h (Nadaraya-Watson), λ (lasso) são chamados de tuning parameters.

Em geral, eles controlam o tradeoff entre viés e variância.

Ex:

l grande: viés baixo, variância alta; l pequeno: viés grande, variância pequena

λ pequeno: viés baixo, variância alta; λ grande: viés grande, variância pequena

Importante

Notação: quantidades como l (regressão polinomial), k (nearest neighbors), h (Nadaraya-Watson), λ (lasso) são chamados de tuning parameters.

Em geral, eles controlam o tradeoff entre viés e variância.

Ex:

l grande: viés baixo, variância alta; l pequeno: viés grande, variância pequena

λ pequeno: viés baixo, variância alta; λ grande: viés grande, variância pequena

Importante

Notação: quantidades como l (regressão polinomial), k (nearest neighbors), h (Nadaraya-Watson), λ (lasso) são chamados de tuning parameters.

Em geral, eles controlam o tradeoff entre viés e variância.

Ex:

l grande: viés baixo, variância alta; l pequeno: viés grande, variância pequena

λ pequeno: viés baixo, variância alta; λ grande: viés grande, variância pequena

No R

K-NN: função `knn.reg` no pacote `FNN`

Alternativa: implemente seu método! Dicas:

Use `rdist` do pacote `fields` para calcular distâncias (muito mais rápido);

Use `sort(...,index.return=TRUE)$ix[1:k]` para buscar os k vizinhos mais próximo

Se seu banco de dados for muito grande, considere usar *k-d trees* (uma forma mais rápida de se calcular os vizinhos mais próximos)

Regressão polinomial: Vocês já são experts :)

No R

K-NN: função `knn.reg` no pacote `FNN`

Alternativa: implemente seu método! Dicas:

Use `rdist` do pacote `fields` para calcular distâncias (muito mais rápido);

Use `sort(...,index.return=TRUE)$ix[1:k]` para buscar os k vizinhos mais próximo

Se seu banco de dados for muito grande, considere usar *k-d trees* (uma forma mais rápida de se calcular os vizinhos mais próximos)

Regressão polinomial: Vocês já são experts :)

No R

K-NN: função `knn.reg` no pacote `FNN`

Alternativa: implemente seu método! Dicas:

Use `rdist` do pacote `fields` para calcular distâncias (muito mais rápido);

Use `sort(...,index.return=TRUE)$ix[1:k]` para buscar os k vizinhos mais próximo

Se seu banco de dados for muito grande, considere usar *k-d trees* (uma forma mais rápida de se calcular os vizinhos mais próximos)

Regressão polinomial: Vocês já são experts :)

No R

K-NN: função `knn.reg` no pacote `FNN`

Alternativa: implemente seu método! Dicas:

Use `rdist` do pacote `fields` para calcular distâncias (muito mais rápido);

Use `sort(...,index.return=TRUE)$ix[1:k]` para buscar os k vizinhos mais próximo

Se seu banco de dados for muito grande, considere usar *k-d trees* (uma forma mais rápida de se calcular os vizinhos mais próximos)

Regressão polinomial: Vocês já são experts :)

No R

K-NN: função `knn.reg` no pacote `FNN`

Alternativa: implemente seu método! Dicas:

Use `rdist` do pacote `fields` para calcular distâncias (muito mais rápido);

Use `sort(...,index.return=TRUE)$ix[1:k]` para buscar os k vizinhos mais próximo

Se seu banco de dados for muito grande, considere usar *k-d trees* (uma forma mais rápida de se calcular os vizinhos mais próximos)

Regressão polinomial: Vocês já são experts :)

No R

K-NN: função `knn.reg` no pacote `FNN`

Alternativa: implemente seu método! Dicas:

Use `rdist` do pacote `fields` para calcular distâncias (muito mais rápido);

Use `sort(...,index.return=TRUE)$ix[1:k]` para buscar os k vizinhos mais próximo

Se seu banco de dados for muito grande, considere usar *k-d trees* (uma forma mais rápida de se calcular os vizinhos mais próximos)

Regressão polinomial: Vocês já são experts :)

Resumo da Aula:

Próxima Aula: Manipulando Documentos de Imagem e Texto

Resumo da Aula:

Próxima Aula: Manipulando Documentos de Imagem e Texto