Introdução à Econometria Semiparamétrica

Aula 2 - Estimação Não Paramétrica Moderna

Luis A. F. Alvarez

10 de outubro de 2024

Recapitulando a estimação por séries

- Recorde-se do análogo populacional

$$\min_{s \in \mathcal{H}} \mathbb{E}[(Y_i - s(\boldsymbol{X}_i))^2],$$

onde $\mathcal{H}$ é um sub-espaço de $L_2(\mathbb{P}_{\boldsymbol{X}})$ "simples" (dimensão finita).

- Na análise de séries, $\mathcal{H} = \Theta_{J_n}$, onde $(\Theta_j)_{j \in \mathbb{N}}$ é uma sequência crescente de espaços com propriedades de aproximação global.
  - A escolha de J_n na prática visava a operar o *trade-off* viés-variância de modo a produzir um estimador com boas propriedades.
    - Por exemplo, para o *spline* cúbico, a escolha ótima em termos de velocidade de convergência do estimador é: $J_n \propto (\log(n)/n)^{s/(s+d)}$

Estimação não paramétrica moderna

- Os métodos da literatura que se convencionou chamar aprendizagem estatística (ou aprendizagem de máquina, em seu braço mais computacional) também partem do problema populacional.

$$\min_{s \in \mathcal{H}} \mathbb{E}[(Y_i - s(\boldsymbol{X}_i))^2],$$

- O que esses problemas adicionam, em relação à estimação clássica por séries?
  - Classes $\mathcal{H}$ de funções que incorporam não linearidade ("expressividade") de um jeito "inteligente", com "menor" complexidade (estimadores de $\downarrow$ variância) que métodos de séries.
  - De modo relacionado, métodos de seleção da complexidade da aproximação utilizada que, implícita ou explicitamente, operam no trade-off viés-variância de modo eficiente.

Estimação não paramétrica em altas dimensões

- Alguns dos métodos de aprendizagem estatística também são bastante úteis em ambiente de alta dimensionalidade.
  - Conceitualmente, ambientes de alta dimensionalidade são aqueles em que a aproximação assintótica mais adequada para representar o processo gerador é uma em que a dimensão de $\boldsymbol{X}$, d_n, diverge com $n \to \infty$, com possivelmente $d_n >> n$.
- Veremos que são as restrições, implícitas ou explícitas, na expressividade das classes $\mathcal{H}$ usadas por métodos de aprendizagem estatística, que garantem seu bom comportamento em ambientes de alta dimensionalidade.
  - O bom funcionamento prático desses métodos decorre, pois, de essas restrições servirem de boa aproximação para o processo gerador verdadeiro.
- Essas restrições levam a um bom funcionamento dos métodos mesmo quando $d_n > n$, evitando o problema do sobreajuste de métodos clássicos.

Problema do sobreajuste

- Considere um contexto em que temos n observações independentes do par $(Y, \boldsymbol{X})$, onde: Y é uma resposta escalar de interesse e $\boldsymbol{X}$ é um vetor de $k < n$ controles
  - Suponha que a matriz de desenho $\mathbb{X}_{n\times k} = [\boldsymbol{X}_1, \ldots \boldsymbol{X}_n]'$ apresenta posto k, e que $\mathbb{E}[Y_i|\boldsymbol{X}_i] = \gamma'\boldsymbol{X}_i$.
- Considere gerar $n - k$ vetores de controles adicionais Z_j, $j = k + 1, \ldots n$, sorteando-os independentemente dos dados e entre si, de uma, $\mathcal{N}(0, 1)$.
- Seja $\hat{\beta}$ o estimador de MQO de Y em X; e $(\tilde{\beta}, \tilde{\psi})$ o estimador de MQO de Y em X e os Z_j.
  - Qual estimador tem o melhor ajuste na amostra?
- Considere realizar uma previsão de Y, com base nos estimadores da amostra, e num novo ponto $\boldsymbol{X}^*$, independente das demais observações?
  - Qual estimador esperamos que funcionará melhor, em termos de erro quadrático médio?

Problema de sobreajuste (cont.)

- O exemplo anterior mostra, num cenário extremo, que estimadores baseados na minimização do risco empírico podem apresentar comportamento bastante indesejável quando a dimensão dos controles é alta.
  - Quando o número de controles k é moderadamente grande em comparação a n, estimador pode exibir um excelente ajuste dentro da amostra, mas funcionar bastante mal em termos de aproximar $\mathbb{E}[Y|\boldsymbol{X}]$, o objeto de interesse.
    - Estimador se ajusta inclusive ao erro idiossincrático ϵ_i dos $Y_i = \mathbb{E}[Y|\boldsymbol{X}_i] + \epsilon_i$ usados na estimação, produzindo alta variância.
- Esse problema é especialmente acentuado na estimação por séries, em que a dimensão do vetor utilizado na estimação, (número de elementos da base de Θ_{J_n}) , cresce exponencialmente no número de entradas de $\boldsymbol{X}$.

Regularização em estimação por séries

- Uma solução, na estimação por séries, é considerar o seguinte problema regularizado.

$$\min_{s \in \Theta_{\bar{J}}} \sum_{i=1}^{n} (y_i - s(\boldsymbol{X}_i))^2 + \lambda \Phi(s)\,,$$

onde $\bar{J}$ pode ser relativamente "grande", e Φ é uma função que denota a "complexidade"de um candidato s.

  - $\lambda > 0$ dá o peso relativo da penalização, vis-à-vis ajuste na amostra (*trade-off* viés-variância).

- **Exemplo:** para *splines* cúbicos, pode-se tomar $\Phi(s) = \int (s''(x))^2 dx$.
  - Nesse caso, se λ é escolhido apropriadamente, número de nós $\bar{J} - 4$ pode ser bastante grande (Claeskens, Krivobokova e Opsomer, 2009; Xiao, 2019).

- Métodos modernos de aprendizagem estatística valem-se de penalizações que induzem estruturas "desejáveis" na solução da otimização.

Penalização L_1

- Seja $\boldsymbol{Z}$ um vetor de k controles (que pode inclusive conter transformações de um vetor original $\boldsymbol{X}$)
- O estimador de mínimos quadrados com penalização L_1 (conhecido como regressão *Lasso*) é dado por:

$$\min_{b \in \mathbb{R}^k} \sum_{i=1}^{n} (Y_i - b' \boldsymbol{Z}_i)^2 + \lambda \sum_{j=1}^{k} \hat{\omega}_j |b_j|$$

- Problema não diferenciável, embora possa ser reescrito como a otimização de um objetivo convexo diferenciável com restrições de desigualdade (escrevendo $b_j = b_j^+ + b_j^-$).
- Penalizamos os coeficientes do modelo quão maior seja seu valor absoluto, com λ o grau de penalização.
  - $\hat{w}_j$ são fatores que podem refletir a escala distinta das entradas de $\boldsymbol{Z}_i$.
  - Escolha comum é $\hat{w}_j = \sqrt{\sum_{i=1}^{n} Z_{i,j}^2}$, o que torna o problema invariante à escala das variáveis/equivalente a estandardizar variáveis e fazer $\hat{w}_j = 1$

Esparsidade da solução e seleção de variáveis

- A estimação por Lasso pode ser vista como realizando seleção automática de variáveis.
- Isso se deve ao fato de que, pela natureza da penalidade, temos que se:

$$\left| -2 \sum_{i=1}^{n} (Y_i - \hat{b}' \boldsymbol{Z}_i) Z_{ij} \right| < \lambda \hat{w}_j \implies \hat{b}_j = 0$$

- Dessa forma, a solução da otimização contará com zeros **exatos**, e, se λ for relativamente grande, haverá muitos desses zeros no vetor $\hat{b}$ (esparsidade).
  - Em contraste, o estimador de MQO que inclui uma variável contínua apresentará um zero na entrada correspondente com probabilidade zero.
- Dado a natureza esparsa das soluções do Lasso, parece razoável utilizá-lo como método de estimação, inclusive quando $k > n$.
  - Qual a condição sobre o processo gerador e as penas para que isso funcione?

Esparsidade aproximada

- A condição crucial para que o Lasso ofereça boas aproximações à esperança condicional é conhecida como esparsidade aproximada (Bickel, Ritov e Tsybakov, 2009), qual seja:

$$\mathbb{E}[Y|\boldsymbol{Z}] = \boldsymbol{Z}'\gamma^* + e(\boldsymbol{Z}),$$

onde $s := \#\{j : \gamma_j^* \neq 0\} = o(k)$ e o erro de aproximação satisfaz:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n} e(\boldsymbol{Z}_i)^2} = O_{\mathbb{P}}\left(\sqrt{\frac{s}{n}}\right)$$

- Isto é, existe uma boa aproximação à esperança condicional utilizando-se somente uma fração dos controles inclusos na especificação.
  - Isso nos permitirá que k seja muito grande relativamente n e ainda assim tenhamos uma boa aproximação
  - Em contraste, estimação clássica por séries permite que os J_n coeficientes associados à aproximação sejam todos zero. Nesse caso J_n não pode ser muito grande.

Escolhendo a penalidade

- O bom comportamento do estimador de Lasso requer uma boa escolha da penalidade $\lambda > 0$.
- A penalidade deve ser alta o suficiente para que, com alta probabilidade, atribuamos um zero às que, de fato, não contribuem à aproximação esparsa ($\downarrow$ variância).
  - Por outro lado, valores λ muito altos podem levar a que variáveis importantes sejam zeradas ($\uparrow$ vies).
- A escolha ideal de penalidade deve dominar o ruído na estimação dos gradientes, quais sejam: $S_j = -2\sum_{i=1}^{n}(Y_i - \mathbb{E}[Y_i|\boldsymbol{Z}_i])\boldsymbol{Z}_{i,j}$, $j = 1, \dots k$.
- Ideia de Belloni, Chen et al. (2012): usar aproximação normal para calcular distribuição de $\max_{j=1,\dots,k}|S_j|$ e escolher pena λ que domine esta quantidade, com alta probabilidade.
  - Veja Chetverikov e Sørensen (2021) para uma alternativa baseada nos valores críticos calculados por reamostragem.

Taxa de convergência

- Sob a condição de esparsidade, a escolha de penalidade ideal, e condições técnicas adicionais, Belloni, Chen et al. (2012) chegam à seguinte taxa de aproximação:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{b}'_{\text{LASSO}}\boldsymbol{Z}_i - \mathbb{E}[Y_i|\boldsymbol{Z}_i])^2} = O_{\mathbb{P}}\left(\sqrt{\frac{s\log(k \vee n)}{n}}\right)$$

- Note que a taxa acima permite que $k >> n$ e ainda se obtenha consistência na norma L_2 empírica
- Autores também consideram a possibilidade de se utilizar MQO após o Lasso, somente incluindo as variáveis que foram selecionadas pelo Lasso (*post-Lasso* de Belloni e Chernozhukov, 2013) e chegam às mesmas taxas.

Regularização L_2

- Vimos que a regularização L_1 funciona bem quando a esperança condicional é bem aproximada por uma especificação esparsa.
- Quando, alternativamente, a aproximação ideal é uma em que muitos coeficientes podem ser diferentes de zero ("densa"), mas a magnitude dos coeficientes é pequena e parecida, costuma-se optar pela regularização L_2 (regressão de *ridge* ou estimador de *shrinkage*):

$$\min_{b \in \mathbb{R}^k} \sum_{i=1}^{n} (Y_i - b' \boldsymbol{Z}_i)^2 + \lambda \sum_{j=1}^{k} \hat{\omega}_j |b_j|^2$$

Propriedades "clássicas" do *Ridge*

- Suponha, por simplicidade, que as variáveis já estejam estandardizadas, de modo que podemos tomar $\hat{\omega}_j = 1$.
- Nesse caso, estimador de *ridge* pode ser escrito como:

$$\hat{b} = (\mathbb{Z}'\mathbb{Z} + \lambda I_k)^{-1}\mathbb{Z}'\mathbb{Y}$$

  onde $\mathbb{Z} = [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n]'$ e $\mathbb{Y} = [Y_1, Y_2, \ldots, Y_n]'$
- Da expressão acima, fica claro que o estimador de *Ridge* opera na fronteira de viés-variância, introduzindo viés no estimador do MQO como forma de reduzir a variância.
  - A ideia de distorcer um estimador não viesado como forma de reduzir o erro quadrático médio não é novo, datando de pelo menos 1956 (estimador de Stein).
  - Distorção é especialmente útil quando $\mathbb{Z}'\mathbb{Z}$ é quase não invertível (muita colinearidade nas variáveis, de modo que variância é alta).
- Procedimento também tem interpretação Bayesiana. Para visualizar isso mais claramente, tome $\boldsymbol{Z} = 1$, de modo que $\hat{b} = \frac{n}{n+\lambda}\bar{Y}$.

Propriedades do *Ridge* em alta dimensão

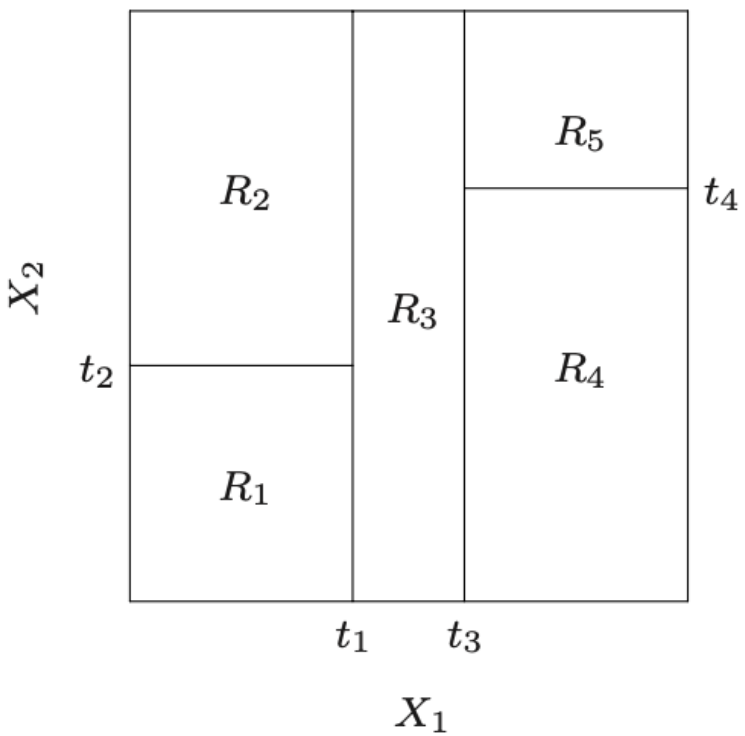
- A discussão anterior motivou o *ridge* do ponto de vista "clássico".
  - Por exemplo, a ideia de introduzir viés no MQO para reduzir variância só faz sentido se o estimador de MQO está bem definido, i.e. se $k \leq n$.
- No entanto, o *ridge* também é utilizado em ambientes de alta dimensão, como forma de garantir invertibilidade da matriz de desenho.
- Nesse caso, suas propriedades não são tão bem entendidas, embora haja desenvolvimentos recentes nessa direção (Spiess, Imbens e Venugopal, 2023).

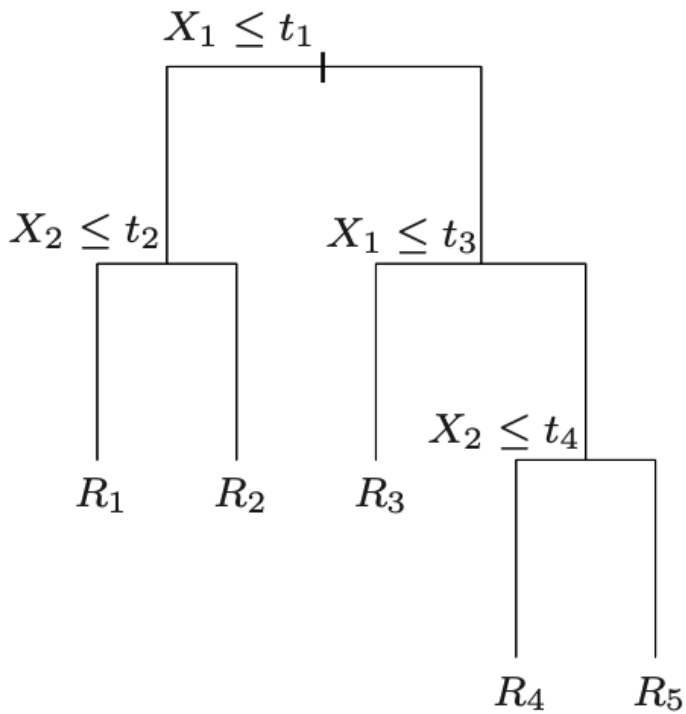
Árvores de regressão

- Uma árvore de regressão é uma função $h(\boldsymbol{Z})$ definida por uma partição retangular do suporte de $\mathbf{Z}$, $\{R_j\}_{j=1}^{J}$, tal que:

$$h(\boldsymbol{Z}) = \sum_{j=1}^{J} a_j \mathbf{1}\{\boldsymbol{Z} \in R_j\}$$

- Essas funções podem ser representadas por árvores de decisão.





Estimação da árvore de regressão

- A estimação de uma árvore de regressão a partir da minimização da soma dos quadrados dos resíduos (risco empírico) é impraticável.
- Nesse caso, é costumeiro se adotar algoritmos gulosos (*greedy optimization*) para a estimação recursiva da árvore.
- Seja v um nó atualmente colocado na árvore, e $N(v)$ os índices das observações que caem sob esse nó.
- Seja um conjunto $P(v) \subseteq \{1, \ldots, k\}$ das direções de partição permitidas no nó v (*split directions*).
  - Pode ser o conjunto de todas as variáveis, ou, em implementações computacionalmente mais tratáveis, um subconjunto aleatório das variáveis.
- A regra de quebra (colocação das folhas) do nó v é dada por $\boldsymbol{Z}_{j*} \geq c^*$, onde

$$\min_{j \in P(v)} \min_{c \in \mathbb{R}} \min_{\overline{y}, \underline{y}} \sum_{i \in N(v)} (Y_i - \overline{y}\mathbf{1}\{\boldsymbol{Z}_j \geq c\} - \underline{y}\mathbf{1}\{\boldsymbol{Z}_j < c\})^2 =$$

$$\min_{j \in P(v)} \min_{c \in \mathbb{R}} \sum_{i \in N(v)} (Y_i - \bar{Y}_{\boldsymbol{Z}_j \geq c}\mathbf{1}\{\boldsymbol{Z}_j \geq c\} - \bar{Y}_{\boldsymbol{Z}_j < c}\mathbf{1}\{\boldsymbol{Z}_j < c\})^2 \quad (1)$$

Regra de parada e poda

- A regra de parada do algoritmo pode se dar com base num número máximo de nós terminais T_0.
  - Além disso, paramos a quebra em um nó quando o número de observações $N(v)$ dentro dele torna-se pequeno.
- As predições finais da árvore são dadas pela média de Y em cada nó terminal.
- Note que a complexidade da árvore estimada depende, crucialmente, do número de nós terminais.
  - Quanto maior o número de nós terminais, menor o viés, embora maior a variância.
- Uma possibilidade é estimar uma árvore com T_0 grande, e depois considerar o efeito de se desfazer alguma das quebras sobre a qualidade preditiva, penalizada pela complexidade do modelo.
  - Isto é, denotando por $\mathcal{T}$ e $\mathcal{T}' \preceq \mathcal{T}$ uma sub-árvore obtida colapsando-se alguns dos *splits*, fazemos a poda ou *pruning*:

$$\min_{\mathcal{T}' \preceq \mathcal{T}} \frac{1}{n} \sum_{i=1}^{n} (Y_i - \mathcal{T}'(\boldsymbol{Z}_i))^2 + \alpha |\mathcal{T}'|, \quad \alpha > 0$$

Extensões

- Uma árvore de regressão, tal qual como a construímos, possui baixa expressividade.
- No entanto, o espaço gerado por combinações lineares de árvores é bem mais flexível.
- Discutiremos dois métodos de gerar essas combinações lineares, que diferem na maneira como lidam com o *trade-off* viés-variância.
  - *Boosting*.
  - Bagging (*random forests*).

Boosting

- Na metodologia de *boosting*, partimos de um estimador inicial $\hat{h}_0$ de baixa complexidade (viés alto, mas variância pequena) e combinamo-lo sequencialmente a J estimadores de baixa complexidade, como forma de reduzir o viés adaptivamente.
  - No caso de árvores de regressão, trabalharemos com árvores com T_0 pequeno.
- Ideia do boosting é, seja $\hat{h}_j$ o estimador obtido até a j-ésima iteração do algoritmo. Se o objetivo é reduzir o EQM do estimador, gostaríamos de perturbá-lo de modo a reduzir:

$$\frac{1}{2}\mathbb{E}[(Y - \hat{h}_j(\boldsymbol{Z}))^2|\boldsymbol{Z}]\,,$$

dada a convexidade da função objetivo, sabemos que isso poderia ser obtido fazendo $\tilde{h}_{j+1}(\boldsymbol{Z}) = \hat{h}_{j+1}(\boldsymbol{Z}) + \delta\mathbb{E}[(Y - \hat{h}_j(\boldsymbol{Z}))|\boldsymbol{Z}]$ para $\delta \in (0, 1)$. ▸ Detalhes

Boosting (cont.)

- Na prática, não observamos $\mathbb{E}[(Y - \hat{h}_j(\boldsymbol{Z}))|\boldsymbol{Z}]$, mas podemos estimá-la aplicando um estimador de baixa-complexidade aos resíduos $\hat{U}_i = (Y_i - \hat{h}_j(\boldsymbol{Z}_i))$, $i = 1 \ldots, n$, de modo a produzir uma função $\hat{g}_j$. que aproxime $\mathbb{E}[\hat{U}_i|\boldsymbol{Z}]$.
- Estimador da etapa $j + 1$ será, então:

$$\hat{h}_{j+1}(\boldsymbol{Z}) = \hat{h}_{j+1}(\boldsymbol{Z}) + \delta \hat{g}_j(\boldsymbol{Z})$$

- Escolha de J opera na fronteira viés-variância. Quanto maior J, menor viés, mas maior a variância.
  - Embora sobreajuste pareça crescer bastante lentamente com J, é importante que se pare J antes da convergência numérica das estimativas (Bühlmann e Yu, 2003; Bühlmann e Hothorn, 2007).
  - Valor de δ geralmente é fixo em uma quantidade pequena.
- Existe ampla literatura com as propriedades estatísticas do *boosting* em regimes de baixa dimensão.
  - Para resultados das propriedades de *boosting* em altas dimensões sob esparsidade (com regressões lineares simples sendo o estimador de baixa complexidade), ver Kueck et al. (2023).

Bagging

- A estimação por *bootstrap aggregation* (*bagging*) toma caminho oposto ao do *boosting*
- Ideia é trabalhar com a média de muitas árvores profundas (baixo viés, mas alta variância), como forma de reduzir sua variância.
- Para que o efeito da agregação sobre a variância seja acentuado, é interessante estimar cada uma das $\hat{h}_j$, $j = 1, \ldots J$ árvores a serem agregadas em amostras menos correlacionadas.
  - No *bagging*, isso é feito ajustando-se a j-ésima árvore numa amostra de n observações sorteada **com reposição** dos dados.
- Estimador resultante é dado por $\frac{1}{S}\sum_{s=1}^{S}\hat{h}_s$ e é conhecido como *random forest*.

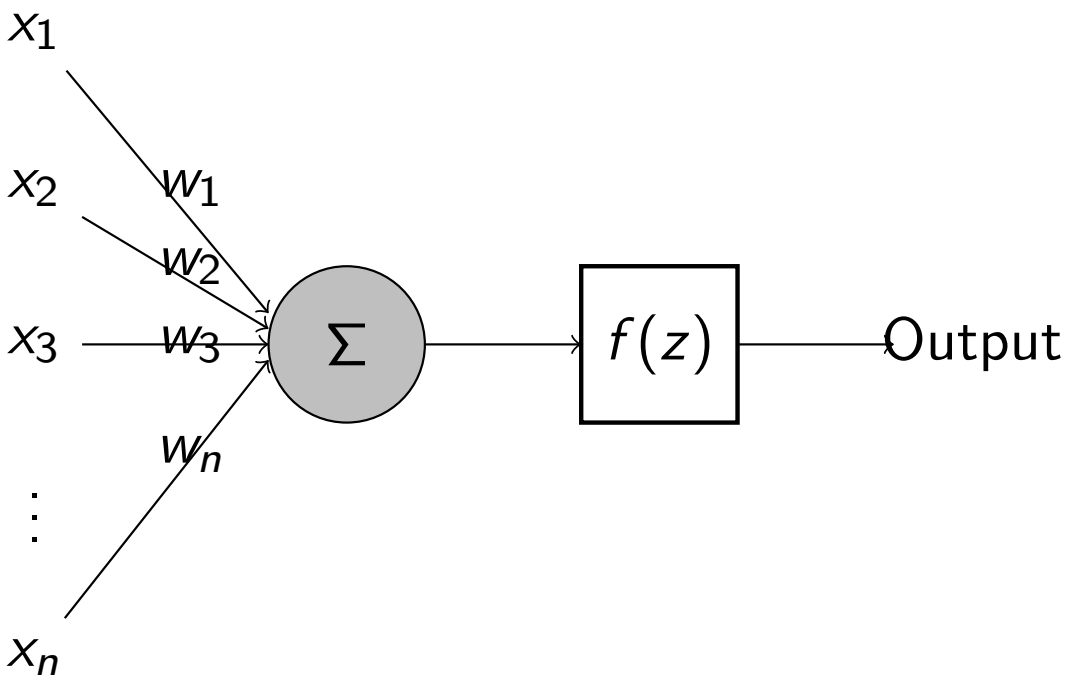
Random forests (cont)

- Propriedades da *random forest* são bem conhecidas em regimes de baixa dimensão, pois nesse caso, o estimador pode ser visto como um estimador de Nadaraya-Watson com *kernel* dado pela estrutura da árvore e em que a profundidade faz as vezes da banda.
  - Taxa de convergência apresentada nesses resultados é da ordem dos estimadores não paramétricos locais que estudamos (portanto sujeita a maldição da dimensionalidade).
  - Para a validade de métodos de inferência, costuma-se trabalhar como uma versão do estimador em que, para cada reamostragem $\mathcal{S}_j$ usada na estimação da j-ésima árvore, somente uma fração das observações é usada na construção da árvore, enquanto a outra fração é usada no cômputo dos valores preditos dos nós terminais (*honestidade*).
- Observação da relação entre *random forest* e Nadarya-Watson levou à consideração de regressões lineares dentro de cada *split* (Rina Friedberg e Wager, 2021).

Random forests (cont)

- No entanto, *random forests* são frequentemente utilizadas em ambientes de alta dimensão.
  - Literatura vem avançando na compreensão desses casos, notando que a taxa parece se adaptar bem a certas noções de esparsidade (Syrgkanis e Zampetakis, 2020).

Percéptron



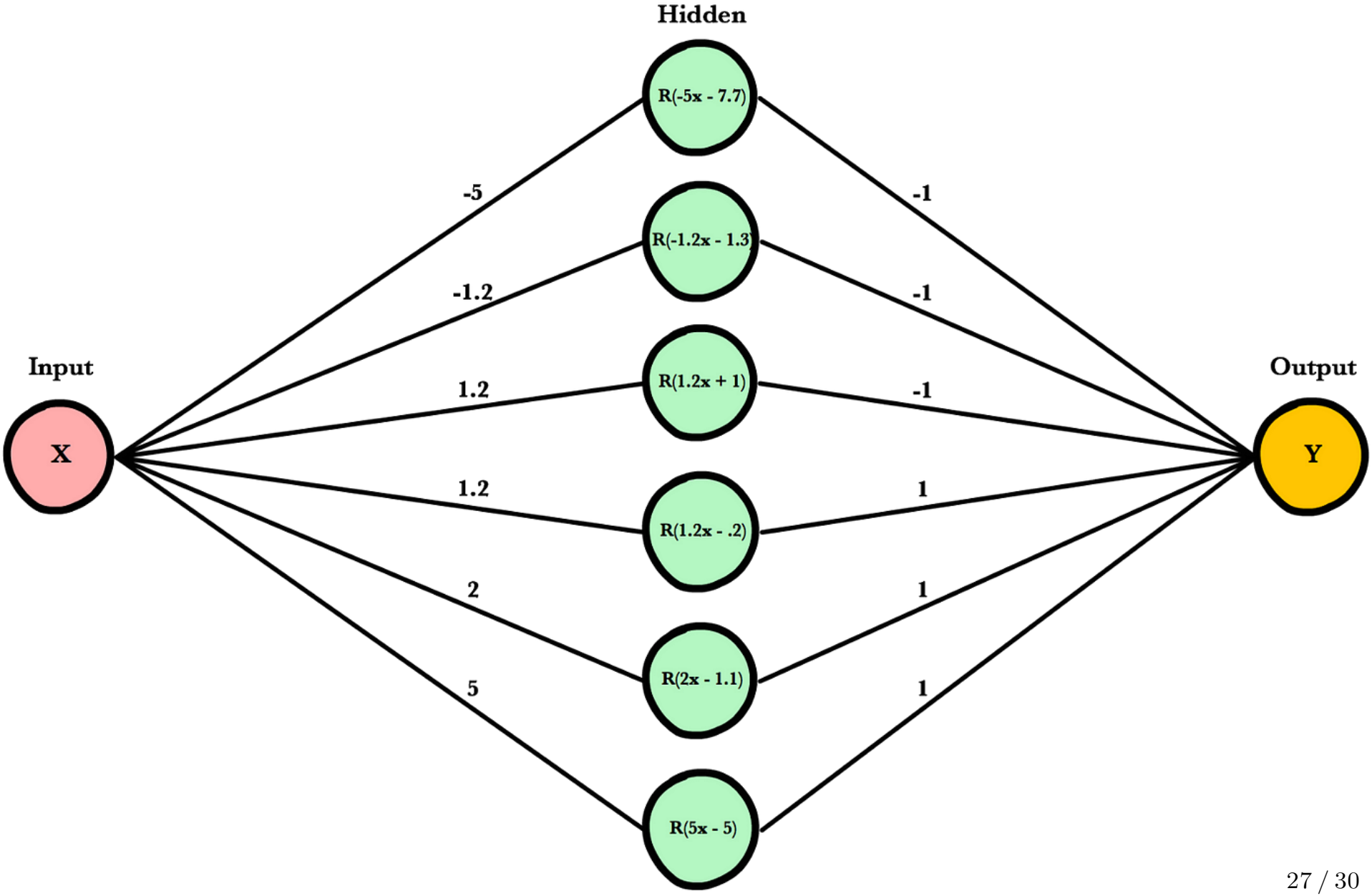
$$\text{Output} = f\left(\sum_{j=1}^{n} w_j x_j\right)$$

Funções de ativação

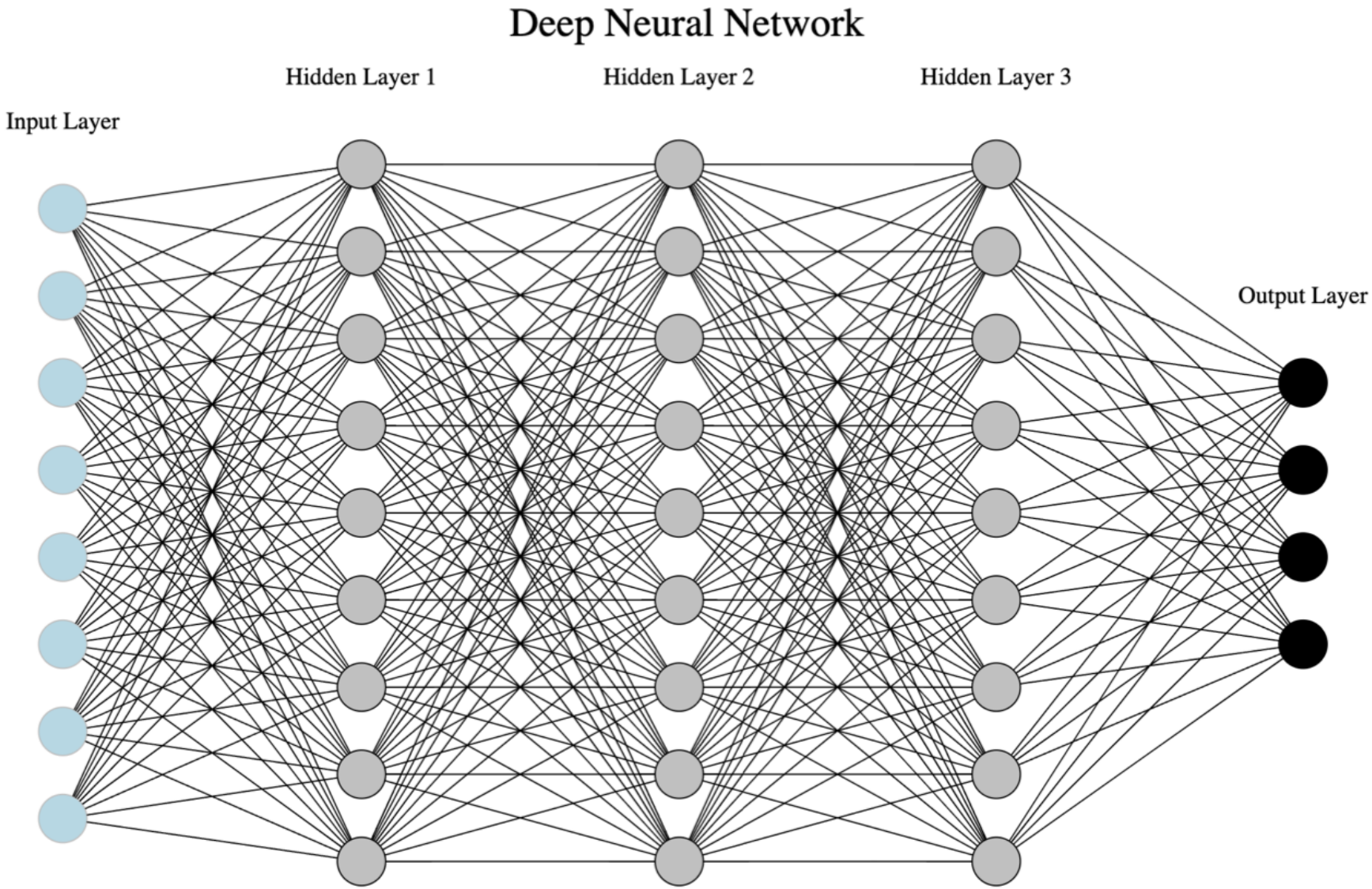
- A função f é conhecida como função de ativação. Algumas escolhas populares são.

Activation Function	**Formula**	**Range**
Step Function	$f(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$	$\{0, 1\}$
Sigmoid (Logistic)	$f(z) = \frac{1}{1+e^{-z}}$	$(0, 1)$
ReLU (Rectified Linear Unit)	$f(z) = \max(0, z)$	$[0, \infty)$
Leaky ReLU	$f(z) = \begin{cases} z & \text{if } z \geq 0 \\ \alpha z & \text{if } z < 0 \end{cases}$	$\mathbb{R}$
Identity	$f(z) = z$	$\mathbb{R}$

Rede neural *feed-forward* com uma camada oculta



Rede neural *feed-forward* profunda



Resultados teóricos

- Os resultados clássicos sobre a qualidade de aproximação das redes neurais focam no caso de baixa dimensão (dimensão dos controles fixo) e em redes com uma única camada oculta.
  - Nesse caso, as redes neurais podem ser vistas como uma versão do estimador de séries, e se a largura (número de percéptrons na camada oculta) J_n cresce otimamente como função do tamanho amostral, o estimador com função de ativação suave apresenta taxas de convergência de no mínimo $n^{-1/4}$ para $\mathbb{E}[Y|\boldsymbol{X}]$ suficientemente suave (Chen e White, 1999; Chen, 2007).
- No entanto, na prática, redes profundas são conhecidamente melhores.
  - Taxas de convergência para redes profundas, em que profundidade e crescimento são funções do tamanho amostral, no caso de dimensão finita, por (Farrell, Liang e Misra, 2021).
- Para ambientes de alta dimensão, resultados recentes sobre o comportamento de redes neurais profundas sob versões de esparsidade são discutidos em Chernozhukov et al., 2024.

Kernel ridge regression

- Um método de introduzir não linearidade que, por limitação de espaço, não vamos discutir aqui conhecido como *kernel ridge regression*.
  - Ideia é introduzir não linearidade no problema trabalhando-se num espaço de funções "complexo" indiretamente, sem a necessidade de calcular as funções, através da definição de um produto interno nele (*kernel trick*).
- Veja a seção 5.8 de Hastie et al. (2009) para uma introdução e Singh e Vijaykumar (2023) para propriedades estatísticas.

Derivação da descida do gradiente

- Uma função convexa $\psi : \mathbb{R} \mapsto \mathbb{R}$ satisfaz:

$$\psi(a) - \psi(b) \geq \psi'(b)(a - b)\,.$$

- Usando convexidade de $x \mapsto \frac{1}{2}(Y - x)^2$, temos, para estimadores $\hat{h}_j(\boldsymbol{Z})$ e $\tilde{h}_{j+1}(\boldsymbol{Z})$.

$$\frac{1}{2}(Y - \tilde{h}_{j+1}(\boldsymbol{Z}))^2 - \frac{1}{2}(Y - \hat{h}_j(\boldsymbol{Z}))^2 \leq -(Y - \tilde{h}_{j+1}(\boldsymbol{Z}))(\tilde{h}_{j+1}(\boldsymbol{Z}) - \hat{h}_j(\boldsymbol{Z}))\,,$$

tomando a esperança condicional a $\boldsymbol{Z}$, temos:

$$\frac{1}{2}\mathbb{E}[(Y - \tilde{h}_{j+1}(\boldsymbol{Z}))^2|\boldsymbol{Z}] - \frac{1}{2}\mathbb{E}[(Y - \hat{h}_j(\boldsymbol{Z}))^2|\boldsymbol{Z}] \leq$$
$$-\mathbb{E}[(Y - \tilde{h}_{j+1}(\boldsymbol{Z}))|\boldsymbol{Z}](\tilde{h}_{j+1}(\boldsymbol{Z}) - \hat{h}_j(\boldsymbol{Z}))\,,$$

- Tomando $(\tilde{h}_{j+1}(\boldsymbol{Z}) - \hat{h}_j(\boldsymbol{Z})) = \delta\mathbb{E}[(Y - \hat{h}_j(\boldsymbol{Z}))|\boldsymbol{Z}]$, o limite superior fica:

$$(\delta^2 - \delta)\mathbb{E}[(Y - \hat{h}_j(\boldsymbol{Z}))|\boldsymbol{Z}]^2\,,$$

que é negativo para $0 < \delta < 1$. ◂ Voltar

Bibliografia I

- Belloni, Alexandre, D. Chen et al. (2012). "Sparse Models and Methods for Optimal Instruments With an Application to Eminent Domain". Em: *Econometrica* 80.6, pp. 2369–2429. DOI: https://doi.org/10.3982/ECTA9626. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.3982/ECTA9626. URL: https://onlinelibrary.wiley.com/doi/abs/10.3982/ECTA9626.
- Belloni, Alexandre e Victor Chernozhukov (2013). "Least squares after model selection in high-dimensional sparse models". Em: *Bernoulli* 19.2, pp. 521–547. DOI: 10.3150/11-BEJ410. URL: https://doi.org/10.3150/11-BEJ410.
- Bickel, Peter J., Ya'acov Ritov e Alexandre B. Tsybakov (2009). "Simultaneous analysis of Lasso and Dantzig selector". Em: *The Annals of Statistics* 37.4, pp. 1705–1732. DOI: 10.1214/08-AOS620. URL: https://doi.org/10.1214/08-AOS620.

Bibliografia II

- Bühlmann, Peter e Torsten Hothorn (2007). "Boosting Algorithms: Regularization, Prediction and Model Fitting". Em: *Statistical Science* 22.4, pp. 477–505. DOI: 10.1214/07-STS242. URL: https://doi.org/10.1214/07-STS242.
- Bühlmann, Peter e Bin Yu (2003). "Boosting With the L2 Loss". Em: *Journal of the American Statistical Association* 98.462, pp. 324–339. DOI: 10.1198/016214503000125. eprint: https://doi.org/10.1198/016214503000125. URL: https://doi.org/10.1198/016214503000125.
- Chen, Xiaohong (2007). "Chapter 76 Large Sample Sieve Estimation of Semi-Nonparametric Models". Em: ed. por James J. Heckman e Edward E. Leamer. Vol. 6. Handbook of Econometrics. Elsevier, pp. 5549–5632. DOI: https://doi.org/10.1016/S1573-4412(07)06076-X. URL: https://www.sciencedirect.com/science/article/pii/S157344120706076X.

Bibliografia III

- Chen, Xiaohong e H. White (1999). "Improved rates and asymptotic normality for nonparametric neural network estimators". Em: *IEEE Transactions on Information Theory* 45.2, pp. 682–691. DOI: 10.1109/18.749011.
- Chernozhukov, Victor et al. (2024). *Applied Causal Inference Powered by ML and AI*. arXiv: 2403.02467 [econ.EM]. URL: https://arxiv.org/abs/2403.02467.
- Chetverikov, Denis e Jesper Riis-Vestergaard Sørensen (2021). "Selecting Penalty Parameters of High-Dimensional M-Estimators using Bootstrapping after Cross-Validation". Em: *arXiv preprint arXiv:2104.04716*.

Bibliografia IV

- Claeskens, Gerda, Tatyana Krivobokova e Jean D. Opsomer (set. de 2009). "Asymptotic properties of penalized spline estimators". Em: *Biometrika* 96.3, pp. 529–544. ISSN: 0006-3444. DOI: 10.1093/biomet/asp035. eprint: https://academic.oup.com/biomet/article-pdf/96/3/529/709761/asp035.pdf. URL: https://doi.org/10.1093/biomet/asp035.
- Farrell, Max H., Tengyuan Liang e Sanjog Misra (2021). "Deep Neural Networks for Estimation and Inference". Em: *Econometrica* 89.1, pp. 181–213. DOI: https://doi.org/10.3982/ECTA16901. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.3982/ECTA16901. URL: https://onlinelibrary.wiley.com/doi/abs/10.3982/ECTA16901.
- Hastie, Trevor et al. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer.

Bibliografia V

- Kueck, Jannis et al. (2023). "Estimation and inference of treatment effects with L2-boosting in high-dimensional settings". Em: *Journal of Econometrics* 234.2, pp. 714–731. ISSN: 0304-4076. DOI: https://doi.org/10.1016/j.jeconom.2022.02.005. URL: https://www.sciencedirect.com/science/article/pii/S0304407622000471.
- Rina Friedberg Julie Tibshirani, Susan Athey e Stefan Wager (2021). "Local Linear Forests". Em: *Journal of Computational and Graphical Statistics* 30.2, pp. 503–517. DOI: 10.1080/10618600.2020.1831930. eprint: https://doi.org/10.1080/10618600.2020.1831930. URL: https://doi.org/10.1080/10618600.2020.1831930.
- Singh, Rahul e Suhas Vijaykumar (2023). *Kernel Ridge Regression Inference*. arXiv: 2302.06578 [math.ST]. URL: https://arxiv.org/abs/2302.06578.

Bibliografia VI

- Spiess, Jann, Guido Imbens e Amar Venugopal (2023). "Double and Single Descent in Causal Inference with an Application to High-Dimensional Synthetic Control". Em: *Advances in Neural Information Processing Systems*. Ed. por A. Oh et al. Vol. 36. Curran Associates, Inc., pp. 63642–63659. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/c904c5d43d8a01177063977bd67bf6fc-Paper-Conference.pdf.
- Syrgkanis, Vasilis e Manolis Zampetakis (2020). *Estimation and Inference with Trees and Forests in High Dimensions*. arXiv: 2007.03210 [math.ST].
- Xiao, Luo (2019). "Asymptotics of bivariate penalised splines". Em: *Journal of Nonparametric Statistics* 31.2, pp. 289–314. DOI: 10.1080/10485252.2018.1563295. eprint: https://doi.org/10.1080/10485252.2018.1563295. URL: https://doi.org/10.1080/10485252.2018.1563295.