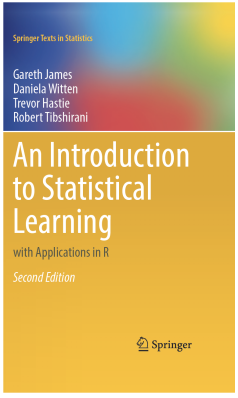
Aula 5: GLMs e Métodos de Reamostragem

Machine Learning

Paulo Orenstein

Verão, 2025
IMPA



Capítulo 4: Métodos lineares para classificação

Créditos de figuras e slides: James, Witten, Hastie e Tibshirani

Capítulo 4

- Classificação: setup geral
- Métodos discriminativos: regressão logística e regressão multinomial
- Métodos generativos: LDA, QDA e naive Bayes
- Avaliação de classificadores
- Modelos lineares generalizados

Modelos lineares generalizados

- Até agora, separamos os nossos métodos de acordo com a natureza do output y
  - Se $y \in \mathbb{R}$, usamos métodos de regressão (capítulo 3)
  - Se $y \in \{0, 1\}$ ou conjuntos finitos, usamos métodos de classificação (capítulo 4)
- O que fazer quando y não é nem um nem outro? Por exemplo, $y \in \{0, 1, 2, 3, \ldots\}$, como é o caso com dados de contagem?
  - Regressão linear poderia retornar valores negativos (ou racionais)
  - Regressão logística funciona quando temos valores categóricos, sem ordenação
- Podemos generalizar a nossa motivação para regressão logística
  - Vamos definir a distribuição de y (por exemplo, $Y|X \sim \text{Pois}(\lambda(X))$)
  - Escolhemos $\lambda(X)$ de tal maneira que o impacto sobre y na log-verossimilhança é linear

Regressão de Poisson

- Um bom modelo para contagens é a distribuição de Poisson: se $Y \sim \text{Pois}(\lambda(X))$, então

$$\mathbb{P}[Y = k|X = x] = \frac{e^{-\lambda(x)}\lambda(x)^k}{k!}, \qquad k = 0, 1, 2, \ldots$$

  Em particular, $\lambda(x) = \mathbb{E}[Y|X = x] = \mathbb{V}[Y|X = x]$

- A log-verossimilhança é:

$$L(y, \lambda(x)) = -\lambda(x) + y \log(\lambda(x)) - \log(y!)$$

- Como o impacto de X sobre y se dá por meio de $\log(\lambda(x))$, gostaríamos que ele fosse linear:

$$\log(\lambda(x)) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p,$$

  ou seja,

$$\lambda(x) = e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p}$$

- Precisamos estimar $\lambda(x)$ ou, equivalentemente, $\beta_0, \ldots, \beta_p$

Regressão de Poisson: estimação

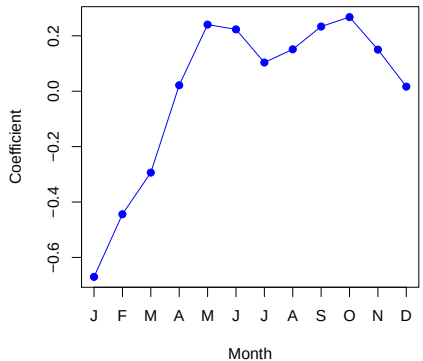
- Como antes, estimamos os coeficientes $\beta_0, \beta_1, \ldots, \beta_p$ via máxima verossimilhança:

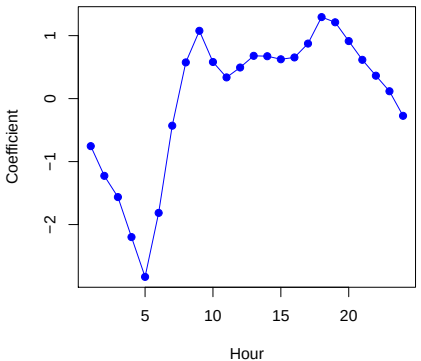
$$
\begin{aligned}
(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p) &= \operatorname*{argmin}_{\tilde{\beta}_0, \ldots, \tilde{\beta}_p} - \left(\sum_{i=1}^{n} -\tilde{\lambda}(x_i) + y_i \log(\tilde{\lambda}(x_i)) \right) \\
&= \operatorname*{argmin}_{\tilde{\beta}_0, \ldots, \tilde{\beta}_p} \sum_{i=1}^{n} e^{\tilde{\beta}_0 + \tilde{\beta}_1 x_{i1} + \cdots + \tilde{\beta}_p x_{ip}} - y_i(\tilde{\beta}_0 + \tilde{\beta}_1 x_{i1} + \cdots + \tilde{\beta}_p x_{ip})
\end{aligned}
$$

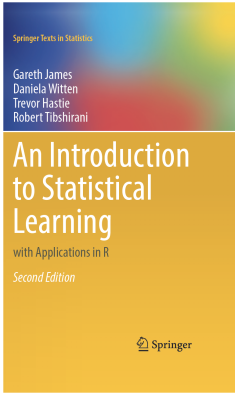
- Essa função é convexa nos coeficientes; podemos usar métodos de otimização convexa
- Depois de encontrar os coeficientes, podemos achar, *e.g.*, $\hat{\mathbb{P}}[Y = 0|X = x]$ ou $\hat{\mathbb{P}}[Y > 5|X = x]$
- Diferenças da regressão de Poisson em relação a regressão linear:
  - Variância: regressão de Poisson supõe certa rigidez, pois $\lambda = \mathbb{E}[Y|X] = \mathbb{V}[Y|X]$
  - Valores não-negativos: valores negativos nunca são previstos (não era o caso antes)
  - Interpretação: um aumento de X_j em 1 tem impacto de e^{β_j} em $\mathbb{E}[Y|X]$ (versus β_j antes)

Regressão de Poisson: exemplo

- Queremos estimar o número de usuários por hora num sistema de compartilhamento de bicicletas públicas via regressão de Poisson no dataset **`bikeshare`**







Capítulo 5: Métodos de reamostragem

Créditos de figuras e slides: James, Witten, Hastie e Tibshirani

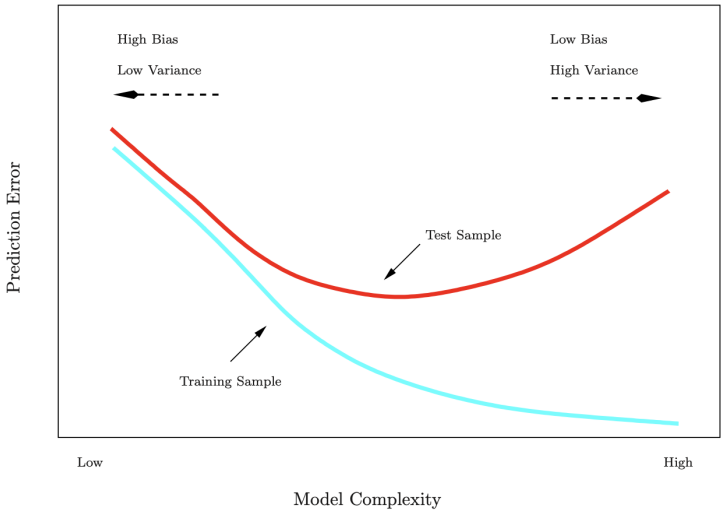
Capítulo 5

- Validação: estimando erros de teste
  - Conjunto de validação
  - Validação cruzada leave-one-out
  - Validação cruzada em k-folds
  - O caso de classificação
- Bootstrap: obtendo medidas de incerteza sobre estimadores

Erros de treino e teste

- O erro de treino é o erro que obtemos ao usar um modelo treinado em dados de treino quando ele tenta prever os próprios dados de treino
- O erro de teste é o erro que um modelo treinado comete em dados que não foram utilizados em seu treinamento
- Quase sempre estamos interessados no erro de teste e não no de treino, pois é uma medida justa do erro que o método terá na prática
- O erro de treino costuma subestimar dramaticamente o erro de teste (*e.g.*, overfitting)
- A relação entre erro de treino e teste indica se há underfitting ou overfitting
- O erro de teste é útil não só em si mesmo, mas também para escolher hiperparâmetros do modelo (*e.g.*, k em kNN); nesse caso, costumamos usar um conjunto extra de validação

Erros de treino e teste

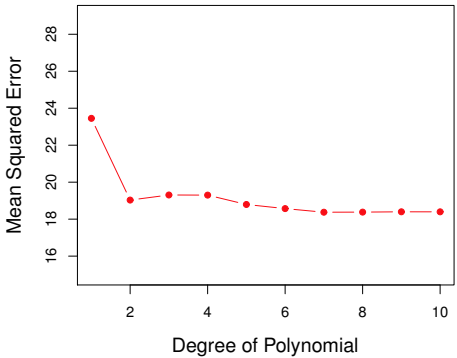


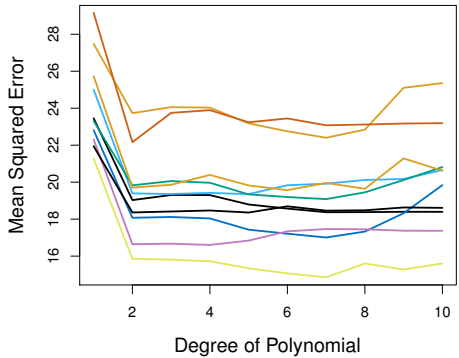
Conjunto de validação: estratégia

- Objetivo: estimar o erro de validação de um método supervisionado (*e.g.*, escolher dentre modelos)
- Estratégia:
  1. Dividir os dados em duas partes
  2. Treinar o método de ML na primeira parte
  3. Computar o erro (*e.g.*, MSE) na segunda parte

Conjunto de validação: exemplo

- Regressão polinomial de **`mpg`** em **`horsepower`** (nos dados **`Auto`**; splits diferentes à direita)



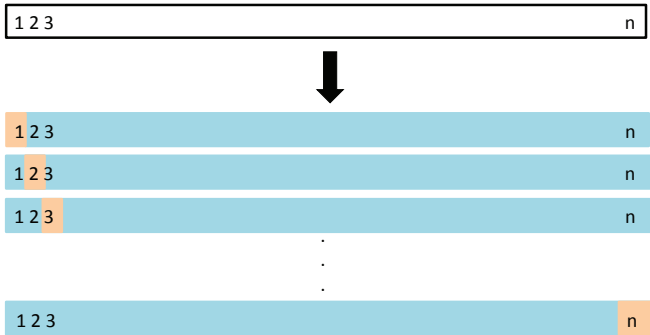


Conjunto de validação: problemas

- Estimativas variam muito com a divisão dos dados, como vimos
- Apenas um subconjunto dos dados é usado para treinar o modelo
- O erro de validação costuma sobrestimar o erro de teste
  - Ao invés de treinar o modelo com n pontos, treinamos só com uma fração dos dados
  - Para fazer uma previsão sobre o futuro, retreinamos o modelo com os n pontos
  - Os dados a mais vão ajudar o modelo, e ele provavelmente vai ter um erro de teste menor do que o estimado

Validação cruzada leave-one-out: estratégia

- Estratégia: para cada $i = 1, \ldots, n$,
  - Treine o modelo em todos os pontos, exceto o i-ésimo
  - Calcule o erro de validação usando apenas o ponto i
- Tire a média de todos os erros de validação

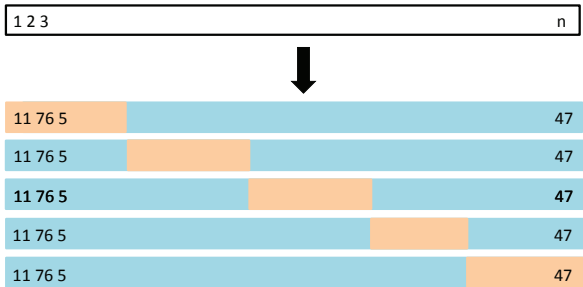


Validação cruzada leave-one-out: vantagens e desvantagens

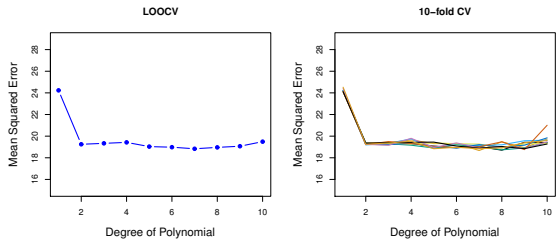
- Seja $\hat{y}_i^{(-i)}$ a previsão do i-ésimo ponto com modelo treinado em todos os pontos menos o i-ésimo
  - Regressão: $\text{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{(-i)})^2$
  - Classificação: $\text{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}_{[y_i \neq \hat{y}_i^{(-i)}]}$
- Vantagens de LOOCV sobre conjunto de validação:
  - LOOCV não é aleatório: sempre retorna o mesmo estimador
  - LOOCV usa quase todos os dados de treino: não sobrestima o erro de teste
- Desvantagens:
  - Modelos treinados em dados correlacionados, aumentando a variância da estimativa de teste
  - É computacionalmente intensivo (exceto no caso de regressão linear, onde há um atalho)

Validação cruzada em k-folds: estratégia

- Divida os dados em k subconjuntos (tipicamente, $k = 5$ ou 10)
- Para cada $i = 1, \ldots, k$:
  - Treine o modelo em todos os subconjuntos exceto o i-ésimo
  - Calcule o erro de validação no i-ésimo subconjunto
- Tire a média de todos os erros de validação

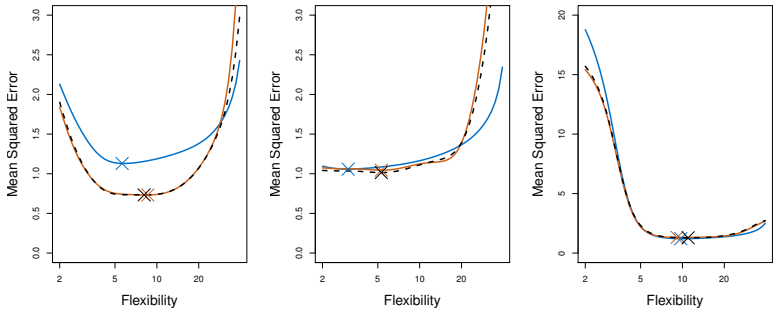


LOOCV vs k-fold CV



- k-fold CV depende da divisão dos dados, mas é menos computacionalmente intensivo
- Em k-fold CV, como no conjunto de validação, treinamos o modelo com menos dados do que o disponível; isso introduz viés, sobrestimando a estimativa de erro
- Em LOOCV, as amostras de treino são muito parecidas umas com as outras; tirar a média de n folds não traz redução tão drástica na variância do estimador de erro de validação

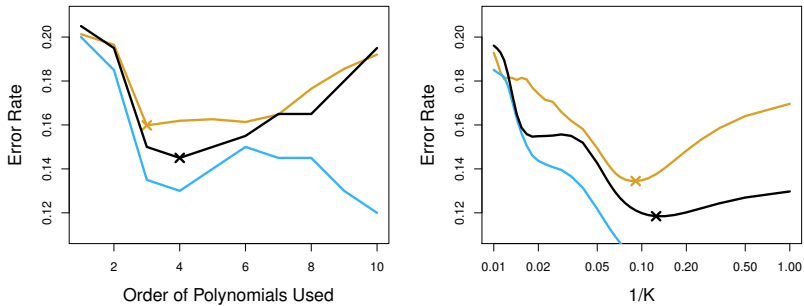
Escolhendo hiperparâmetros: regressão



- Mesmo que as estimativas de LOOCV não sejam sempre iguais a 10-fold CV (versus MSE de teste), os mesmos valores de hiperparâmetros costumam ser escolhidos

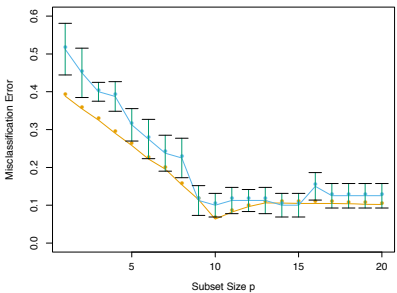
Escolhendo hiperparâmetros: classificação

- A situação é parecida em problemas de classificação (10CV, treino, teste), mas usamos perda 0-1



Escolhendo hiperparâmetros: regra do desvio padrão

- Exemplo de forward selection com p previsores (10-CV, erro verdadeiro)



- Modelos $p = 9, 10, 11, \ldots, 15$ têm quase o mesmo erro de CV
- Regra: escolha o modelo mais simples a um desvio padrão do melhor modelo ($p = 9$)

Jeito errado de fazer CV

- Queremos classificar se 200 indivíduos têm ou não câncer. Usamos regressão logística com 1000 medidas de expressão genética
- Proposta:
  - Usando todos os dados, usar testes z para encontrar os 20 genes mais significativos
  - Estimar o erro de teste da regressão logística nesses 20 previsores via 10-CV
- Isso é razoável?

Jeito errado de fazer CV

- Suponha o seguinte setup:
  - A expressão de um gene é Normal e independente dos outros
  - A resposta (câncer ou não) vem do jogar de uma moeda, sem correlação com a genética
- Qual é a taxa de erro para um classificador usando esses previsores? Em torno de 50%
- Mas usando o método do slide anterior, o erro de 10CV é de 3%!
  - Com $n = 200$ indivíduos e $p = 1000$ variáveis, alguns indivíduos vão ser acidentalmente correlacionados com a resposta
  - Fizemos seleção de previsores usando todos os dados, então em todos os subconjuntos deve haver alguma correlação com a resposta

Jeito certo de fazer CV

- Dividir os dados em 10 subconjuntos
- Para $i = 1, \ldots, 10$:
  - Usando todos os subconjuntos exceto o i-ésimo, faça a seleção de variáveis e treine o modelo com as variáveis selecionadas
  - Calcule o erro de teste no i-ésimo subconjunto
- Tire a média dos 10 erros de testes encontrados
- Dessa maneira, o erro estimado volta a ser perto de 50%
- Nota: em folds diferentes escolhemos variáveis diferentes!
- Moral da história: qualquer aspecto metodológico a ser aprendido através dos dados (*e.g.*, seleção de variáveis) precisa estar dentro da validação cruzada

Capítulo 5

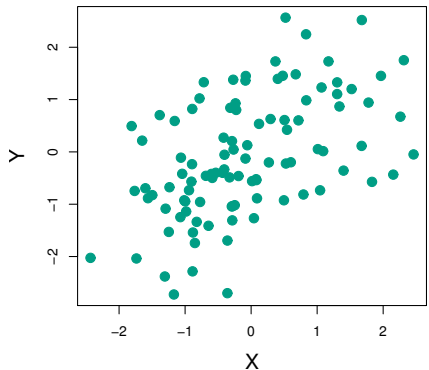
- Validação: estimando erros de teste
  - Conjunto de validação
  - Validação cruzada leave-one-out
  - Validação cruzada em k-folds
  - O caso de classificação
- Bootstrap: obtendo medidas de incerteza sobre estimadores

Bootstrap

- Objetivo: determinar a incerteza de estimadores de maneira não-paramétrica
- Até agora, encontramos o erro-padrão de estimadores de maneira específica
- Por exemplo: como estimar a variância da amostra $x_1, \ldots, x_n$ e o erro padrão desse estimador?
  - Usamos $\hat{\sigma}^2 = (n-1)^{-1} \sum_{i=1n}^{n} (x_i - \overline{x})^2$
  - Para estimar o erro padrão, assumimos que $X_i \sim N(\mu, \sigma^2)$
  - Daí $\hat{\sigma}^2(n-1) \sim \chi^2_{n-1}$, ou seja sabemos qual é a distribuição do estimador
  - Isso nos diz tudo que gostaríamos de saber sobre $\hat{\sigma}^2$; por exemplo, $\mathbb{V}[\hat{\sigma}^2]$ ou $\mathbb{P}[\hat{\sigma}^2 > t]$
- O que fazer quando não é razoável assumir dados Normais? E se o estimador não tiver uma distribuição conhecida?

Exemplo: investimento em dois ativos

- Suponha que X e Y denotem os retornos de dois ativos



Exemplo: investimento em dois ativos

- Queremos investir α do nosso dinheiro em X e $1-\alpha$ em Y. Quanto investir em cada?
- O retorno será $\alpha X + (1-\alpha)Y$. O α que minimiza a variância é

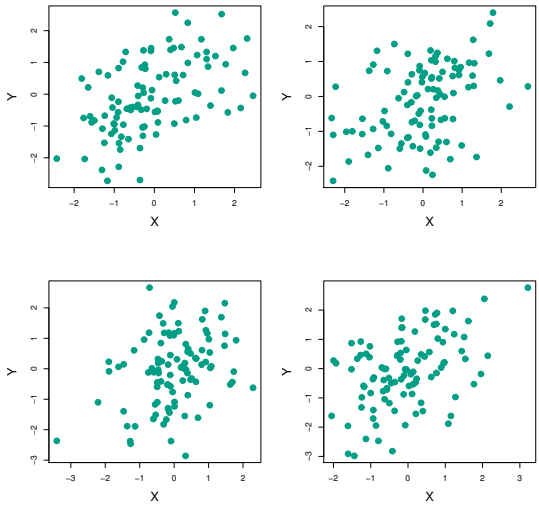
$$\alpha = \frac{\sigma_Y^2 - \mathrm{Cov}(X, Y)}{\sigma_X^2 + \sigma_Y^2 - 2\,\mathrm{Cov}(X, Y)}$$

- Podemos estimar essa quantidade via

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \widehat{\mathrm{Cov}}(X, Y)}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\widehat{\mathrm{Cov}}(X, Y)}$$

- Suponha que com uma amostra obtivéssemos $\hat{\alpha} = 0.6$. Podemos confiar nesse valor?
- Se soubéssemos a distribuição conjunta $\mathbb{P}[X, Y]$, bastaria simular e ver como os dados variam

Exemplo: investimento em dois ativos



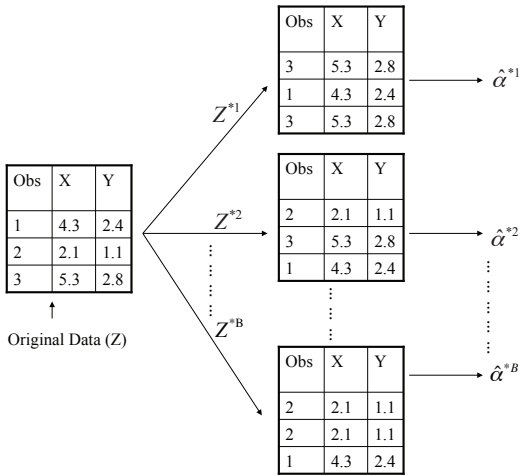
Bootstrap: ideia

- Ideia: para cada reamostragem dos dados, calculamos um valor de $\hat{\alpha}$:

$$
\begin{aligned}
(x_1^{(1)}, y_1^{(1)}), \ldots, (x_n^{(1)}, y_n^{(1)}) &\mapsto \hat{\alpha}^{(1)} \\
(x_1^{(2)}, y_1^{(2)}), \ldots, (x_n^{(2)}, y_n^{(2)}) &\mapsto \hat{\alpha}^{(2)} \\
&\vdots \\
(x_1^{(m)}, y_1^{(m)}), \ldots, (x_n^{(m)}, y_n^{(m)}) &\mapsto \hat{\alpha}^{(m)}
\end{aligned}
$$

- Vamos estimar o erro padrão de $\hat{\alpha}$ pelo desvio padrão de $\hat{\alpha}^{(1)}, \ldots, \hat{\alpha}^{(m)}$
- Note que só temos n pontos originais, então vamos reamostrar os n dados com reposição
- Isso equivale a "simular" a incerteza

Bootstrap: mecanismo



Bootstrap: comparação entre distribuição verdadeira e bootstrap

- Suponha que façamos $m = 1000$ reamostragens
- No nosso exemplo, obtemos

$$\overline{\alpha} = \frac{1}{1000} \sum_{r=1}^{1000} \hat{\alpha}_r = 0.5996,$$

  bem próximo do valor real $\alpha = 0.6$

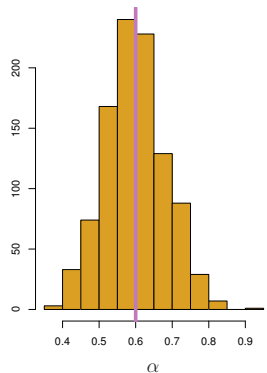
- O desvio padrão das estimativas é

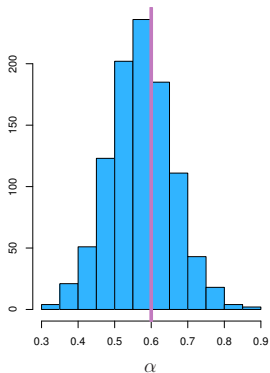
$$\sqrt{\frac{1}{1000-1} \sum_{r=1}^{1000} (\hat{\alpha}_r - \overline{\alpha})^2} = 0.083,$$

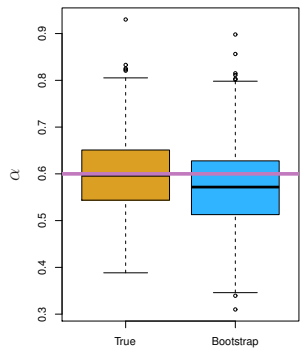
  então estimamos o erro padrão como $\widehat{\mathrm{SE}}(\hat{\alpha}) = 0.083$. Isso é próximo do verdadeiro valor.

- Mais útil ainda: a distribuição de Bootstrap do estimador aproxima a distribuição verdadeira

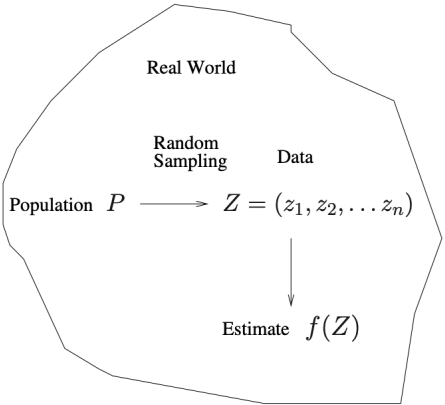
Bootstrap: comparação entre distribuição verdadeira e bootstrap

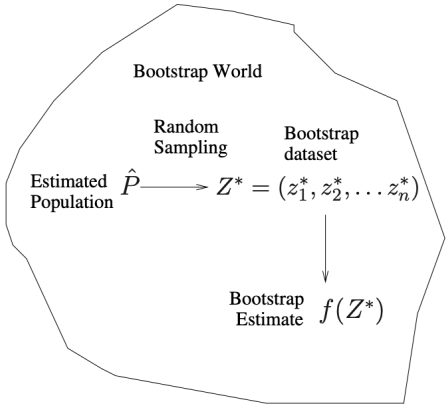






Bootstrap: filosofia





Bootstrap: usos

- Principal uso: obter desvios padrões de estimadores (e outras quantificações de incerteza)
- Também pode ser usado para encontrar intervalos de confiança (usando os quantis desejados da distribuição de bootstrap)
- Situações mais complexas exigem cuidado; *e.g.*, séries temporais e o bootstrap em blocos
- Seria possível usar bootstrap para estimar erro de teste?
  - Usaríamos cada amostra de bootstrap como amostra de treino, os dados de treino como a amostra completa
  - Problema: há interseção entre dados de treino e teste! Isso subestima o erro de teste
  - Alternativa: usar dados de teste como os pontos não sorteados na amostra de bootstrap; mas começa a ficar complicado. Melhor usar k-fold CV

Perguntas para revisão

- O que são GLMs? Como encontrar $\hat{\beta}$? Como interpretar os coeficientes?
- O que é possível aprender sobre um modelo comparando erros de treino e de teste?
- Qual é a diferença entre conjunto de validação, LOOCV e k-fold CV? Quando preferir um ao outro?
- Qual é o jeito errado de fazer CV?
- O que é o bootstrap e como encontrar estimativas de incerteza de método arbitrários?
- Quais são as limitações de bootstrapping?