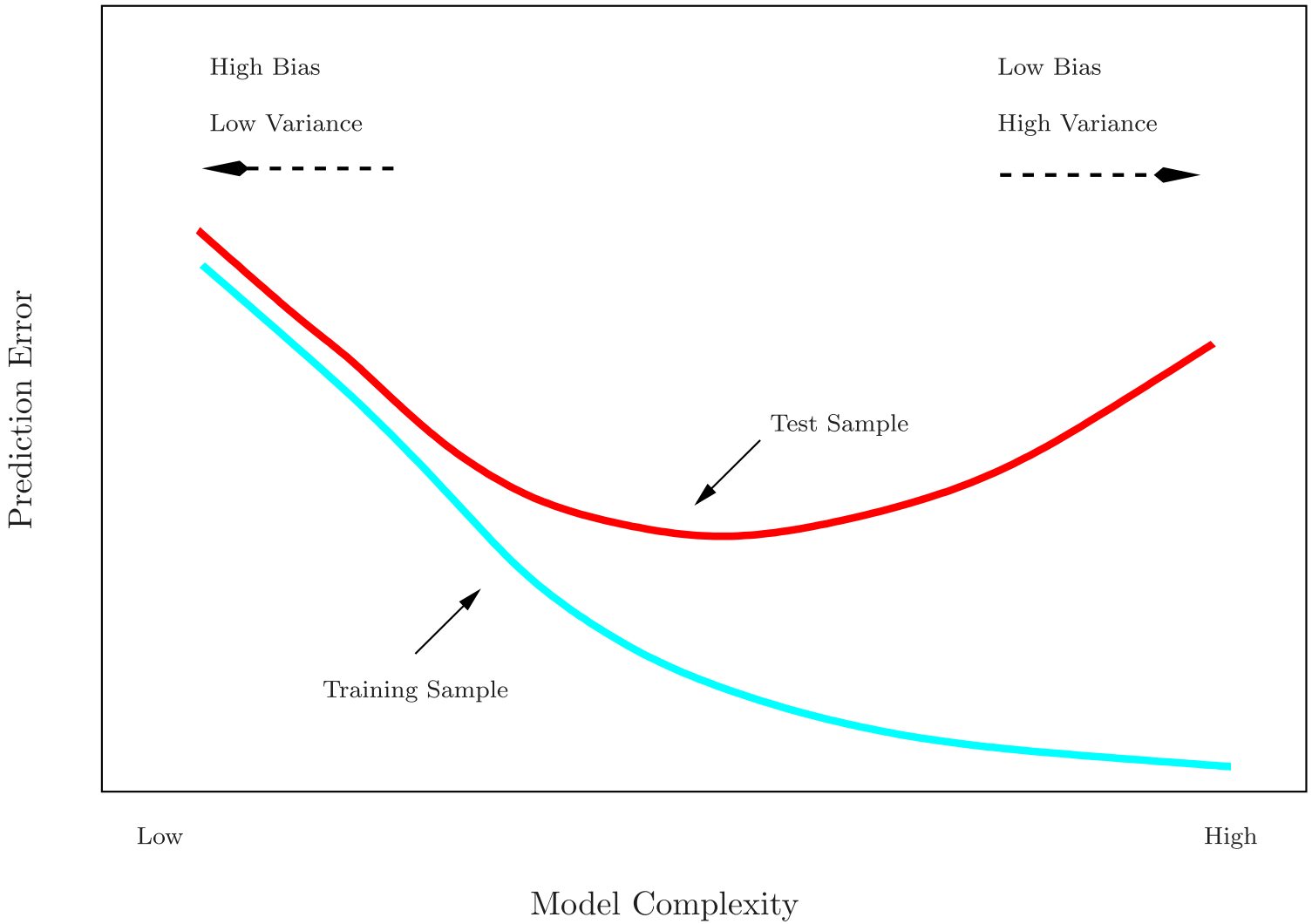
Křížová validace a bootstrap

- V této části probereme dvě metody *převzorkování*: křížovou validaci a bootstrap.
- Tyto metody opětovně prokládají model našeho zájmu vzorky vytvářenými z trénovacího souboru s cílem získat dodatečné informace o proloženém modelu.
- Například dávají odhady chyby předpovědi na testovacím souboru a směrodatnou odchylku a zkreslení našich odhadů parametrů.

Trénovací chyba versus testovací chyba

- Připomeňte si rozdíl mezi *trénovací chybou* a *testovací chybou*.
- *Testovací chyba* je průměrná chyba, která vzniká při použití metody statistického učení k predikci odpovědi na novém pozorování, takovém, které se nepoužilo při tréninku metody.
- Naproti tomu *trénovací chyba* se dá snadno vypočítat tak, že metodu statistického učení aplikujeme na pozorování použitá k jejímu tréninku.
- Ale míra trénovací chyby je často zcela odlišná od míry testovací chyby a především může ta první z nich *dramaticky podhodnocovat* tu druhou.

Efektivita: trénovací soubor versus testovací soubor



Více o odhadech chyby předpovědi

- Nejlepší řešení: velký k tomu určený soubor dat. Často není k dispozici.
- Některé metody provádějí *matematickou úpravu* míry trénovací chyby s cílem odhadnout míru testovací chyby. Patří k nim *Cp statistika*, *AIC* a *BIC*. Mluví se o nich jinde v tomto kurzu.
- Zde se místo toho zabýváme třídou metod, které odhadují testovací chybu tak, že *odloží stranou* z procesu prokládání podmnožinu trénovacích pozorování a pak aplikují metodu statistického učení na tato odložená pozorování.

Přístup používající validační soubor

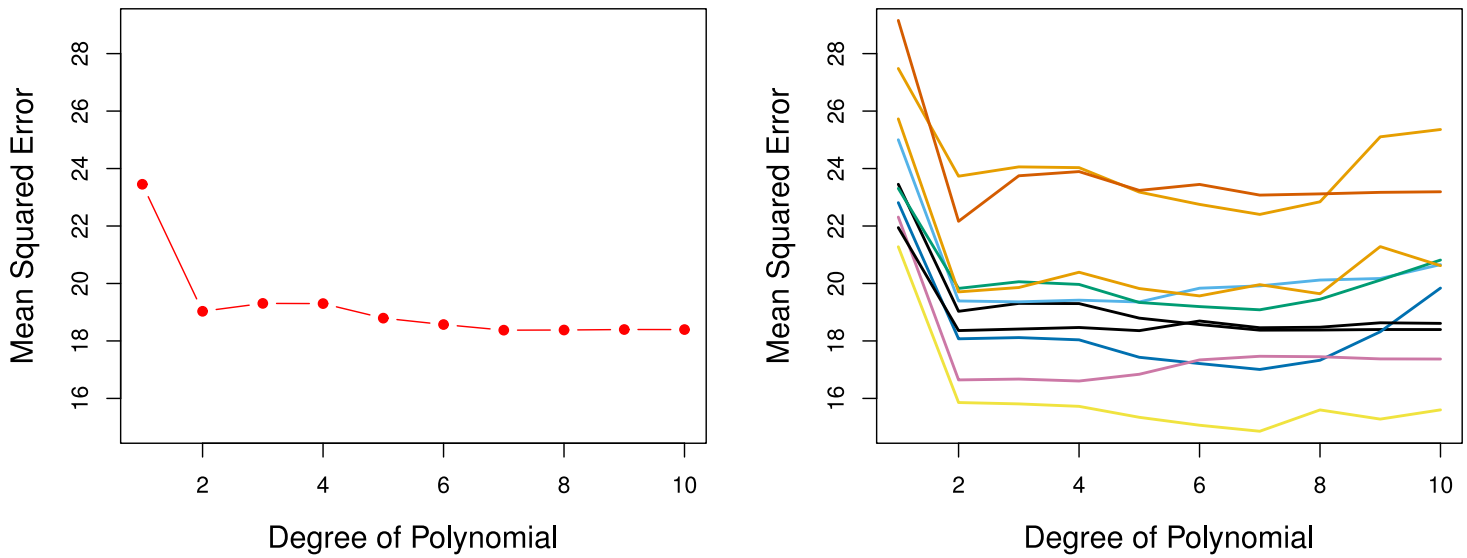
- Zde náhodně rozdělíme dostupný soubor vzorků na dvě části: *tréninkovou sadu* a *validační* neboli *odloženou sadu*.
- Model se proloží na tréninkové sadě a proložený model se použije k předpovědi odpovědí na pozorování ve validační sadě.
- Výsledná chyba na validační sadě nám dává odhad testovací chyby. Ta se obvykle posuzuje pomocí MSE v případě kvantitativní odpovědí a pomocí míry chybné klasifikace v případě kvalitativní (diskrétní) odpovědi.

Proces validace

Náhodné rozdělení na dvě poloviny: levá část je trénovací sada, pravá část je validační sada.

Příklad: údaje o automobilech

- Chceme porovnat lineární členy s členy vyšších řádů v polynomech užitých v lineární regresi.
- Náhodně rozdělíme 392 pozorování na dvě sady, trénovací sadu se 196 datovými body a validační sadu obsahující zbylých 196 pozorování.



Levý panel ukazuje jediné rozdělení; pravý panel ukazuje více různých rozdělení.

Nevýhody přístupu s validačním souborem

- Validační odhad testovací chyby může být vysoce proměnlivý, závisí totiž přesně na tom, která pozorování se zahrnou do tréninkové sady a která jsou zahrnuta do validační sady.
- U validačního přístupu se k proložení modelu používá pouze podmnožina pozorování — ta, která jsou zahrnuta do tréninkové sady a ne ta ve validační sadě.
- To napovídá, že chyba na validační sadě může mít tendenci *nadhodnocovat* testovací chybu pro model proložený celým souborem dat. *Proč?*

K-násobná křížová validace

- *Široce používaný přístup* k odhadování testovací chyby.
- Odhady se dají použít k výběru nejlepšího modelu a k získání představy o testovací chybě finálního zvoleného modelu.
- Myšlenka zde je náhodně rozdělit data do K stejně velkých částí. Vynecháme část k, proložíme model zbylými $K-1$ částmi (kombinovaně) a pak získáme předpovědi pro odloženou k-tou část.
- Toto se provádí po řadě pro každou část $k = 1, 2, \ldots, K$ a pak se výsledky zkombinují.

K-násobná křížová validace podrobně

Rozděl data do K zhruba stejně velkých částí (zde je $K = 5$).

1	2	3	4	5
Validation	Train	Train	Train	Train

Podrobnosti

- Nechť těch K částí jsou $C_1, C_2, \ldots, C_K$, kde C_k označuje indexy pozorování v části k. V části k je n_k pozorování; pokud n je násobkem K, je $n_k = n/K$.
- Vypočítejte

$$\mathrm{CV}_{(K)} = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{MSE}_k,$$

  kde $\mathrm{MSE}_k = \sum_{i \in C_k} (y_i - \hat{y}_i)^2 / n_k$ a $\hat{y}_i$ je aproximace pro pozorování i získaná z dat s odloženou částí k.
- Položíme-li $K = n$, je výsledkem n-násobná validace neboli *křížová validace s vynecháním jednoho* (LOOCV, leave-one out cross-validation).

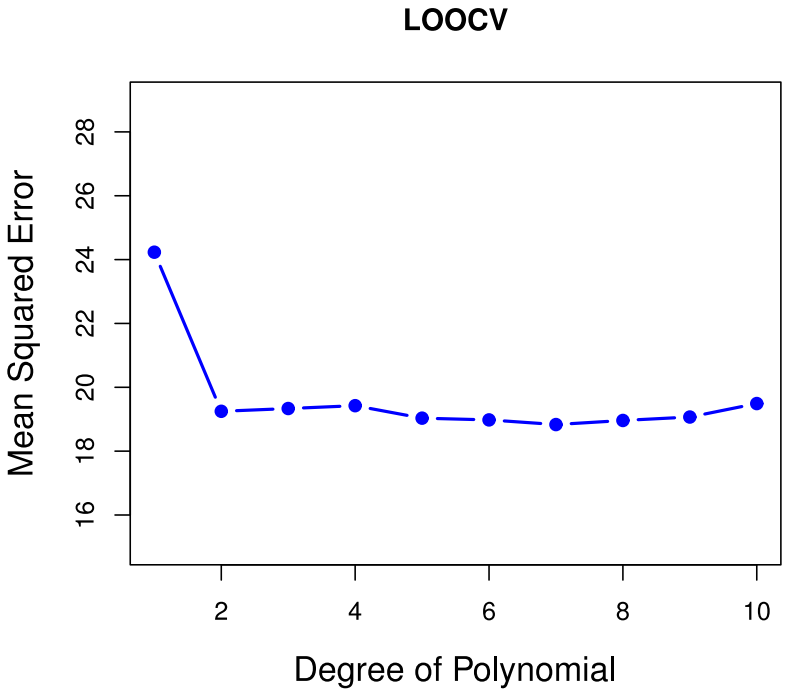
Pěkný speciální případ!

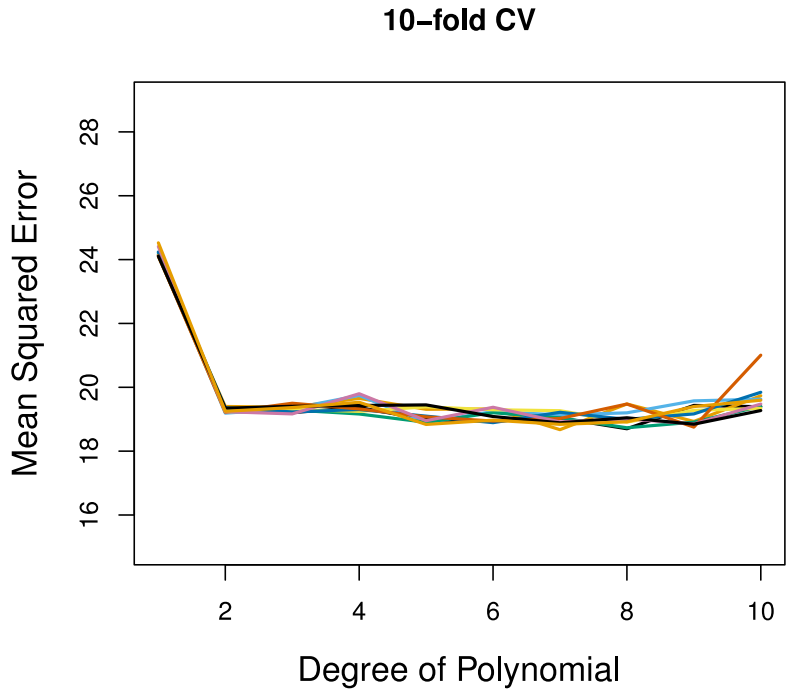
- U lineární nebo polynomiální regrese metodou nejmenších čtverců je tu překvapivý trik, který dělá cenu LOOCV stejnou jako je cena proložení jediným modelem. Platí následující vzorec:

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2,$$

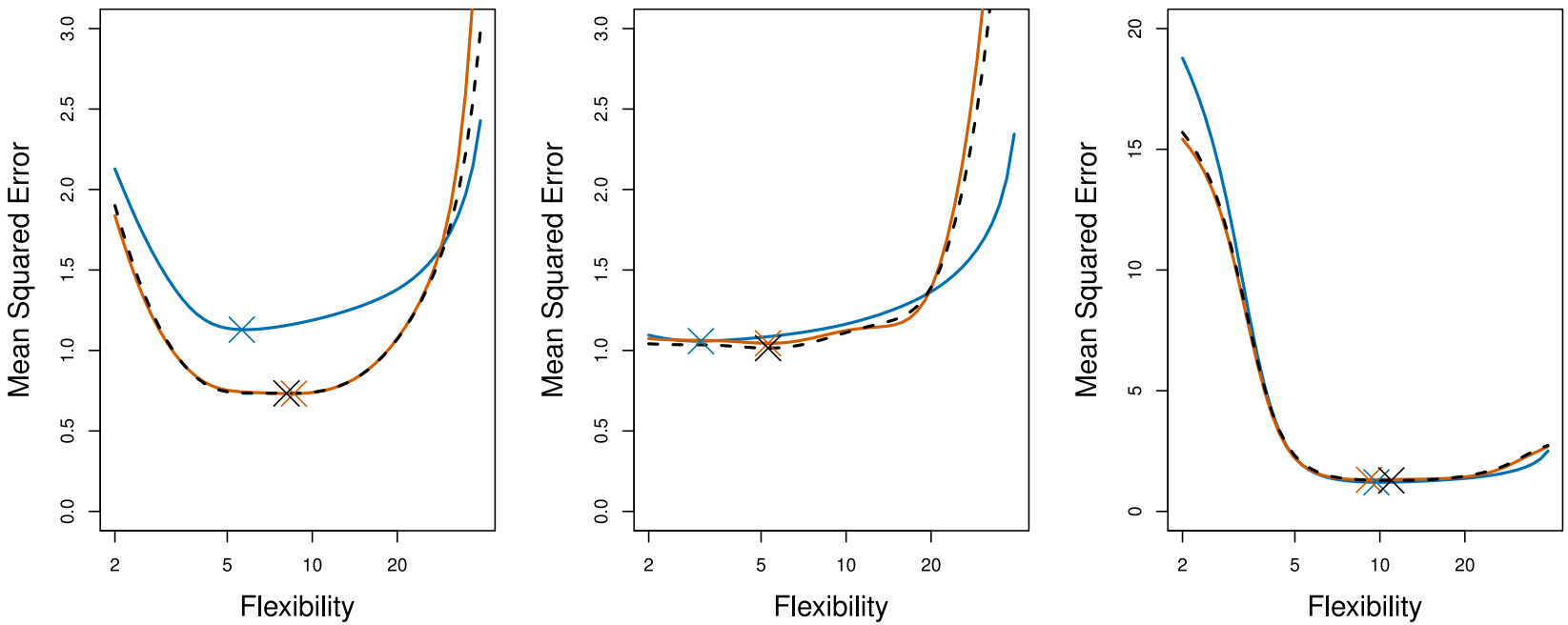
  kde $\hat{y}_i$ je i-tá proložená hodnota z původní aproximace metodou nejmenších čtverců a h_i je účinek (diagonální prvek „stříškové“ matice; podrobnosti jsou v knize). Je to jako obvyklá MSE s tou výjimkou, že i-té reziduum je děleno $1 - h_i$.
- LOOCV je někdy užitečná, ale typicky *neprotřese* data dostatečně. Odhady z jednotlivých složek jsou vysoce korelovány a jejich průměr má tudíž vysoký rozptyl.
- Lepší volba je $K = 5$ nebo 10.

Data o automobilech zkoumaná znovu





Skutečná a odhadnutá testovací MSE pro simulovaná data



Další problémy s křížovou validací

- Jelikož každá trénovací sada je pouze $(K-1)/K$-krát tak velká jako původní trénovací sada, odhady chyby předpovědi budou typicky zkresleny směrem nahoru. *Proč?*
- Toto zkreslení se minimalizuje při $K = n$ (LOOCV), ale tento odhad má vysoký rozptyl, jek jsme uvedli dříve.
- $K = 5$ nebo 10 dává dobrý kompromis pro vyvážení tohoto vztahu zkreslení a rozptylu.

Křížová validace pro klasifikační úlohy

- Rozdělíme data na K zhruba stejně velikých částí $C_1, C_2, \ldots, C_K$. C_k označuje indexy pozorování v části k. V části k je n_k pozorování: jestliže n je násobkem K, pak $n_k = n/K$.
- Vypočítáme

$$\mathrm{CV}_K = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{Err}_k$$

  kde $\mathrm{Err}_k = \sum_{i \in C_k} I(y_i \neq \hat{y}_i)/n_k$.
- Odhadnutá směrodatná odchylka CV_K je

$$\widehat{\mathrm{SE}}(\mathrm{CV}_K) = \sqrt{\sum_{k=1}^{K} (\mathrm{Err}_k - \overline{\mathrm{Err}_k})^2/(K-1)}$$

- Toto je užitečný odhad, ale, přesněji řečeno, není zcela správně. *Proč?*

Křížová validace: správně a chybně

- Uvažujme jednoduchý klasifikátor aplikovaný na nějaká dvoutřídní data:
  1. Začínáme s 5000 prediktory a 50 vzorky a najdeme těch 100 prediktorů, které mají největší korelaci se štítky tříd.
  2. Pak použijeme nějaký klasifikátor, jako je třeba logistická regrese, pouze na těchto 100 prediktorů.

  Jak odhadneme účinnost tohoto klasifikátoru na testovacím souboru?

  Můžeme použít křížovou validaci v kroku 2 a zapomenout na krok 1?

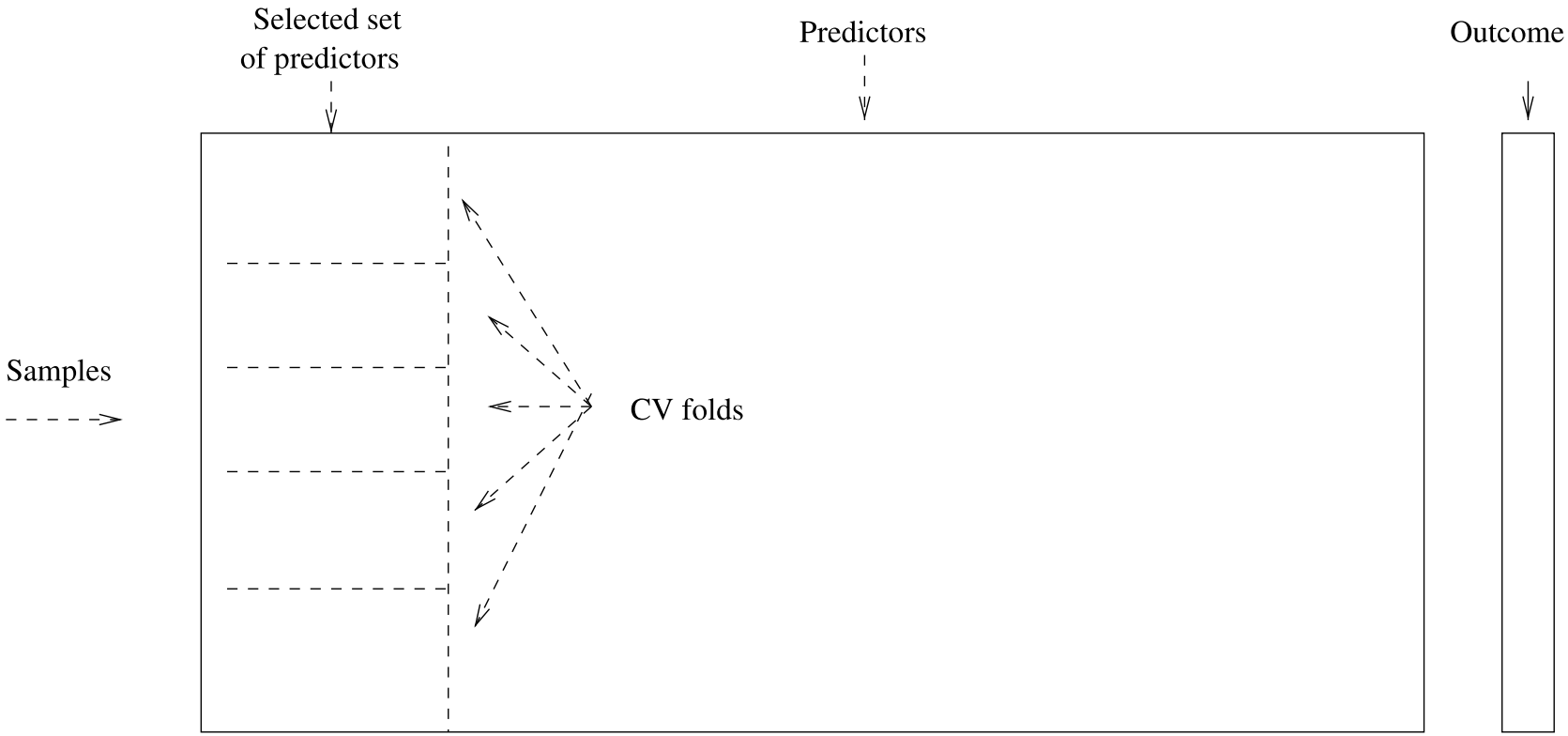
NE!

- To by ignorovalo skutečnost, že naše procedura v kroku 1 *již viděla štítky trénovacích dat* a zužitkovala je. To je forma tréninku a musí to být do validačního procesu zahrnuto.
- Je snadné nasimulovat realistická data se štítky tříd nezávisejícími na výstupu, takže skutečná testovací chyba je 50 %, ale odhad chyby křížovou validací, který ignoruje krok 1, bude nula! *Zkuste to udělat sami.*
- Viděli jsme tuto chybu dělat v mnoha článcích z genomiky ve vysoce profilovaných časopisech.

Chybný a správný způsob

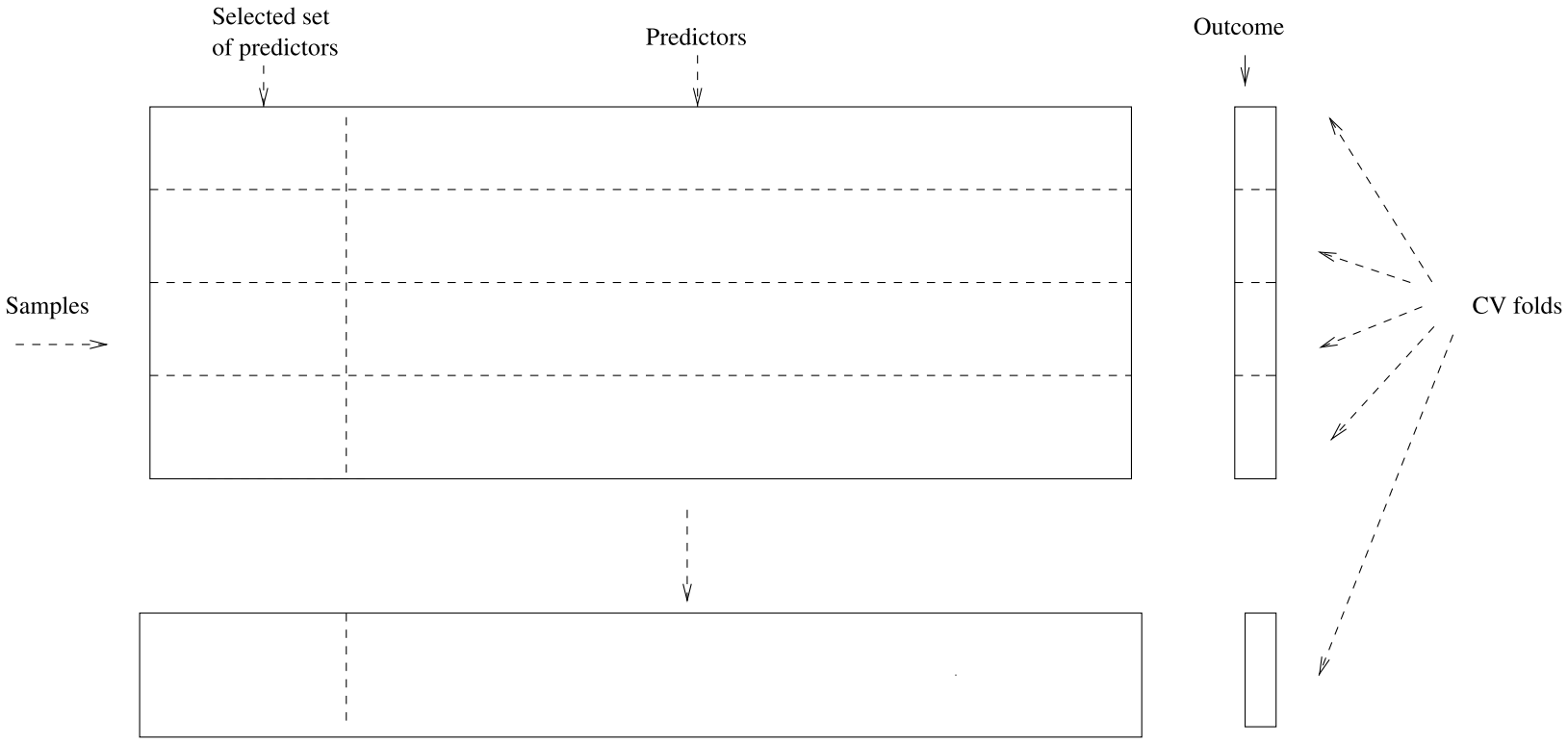
- *Chybně:* Použijeme křížovou validaci v kroku 2.
- *Správně:* Použijeme křížovou validaci v krocích 1 a 2.

Chybný způsob



Texty: Vzorky, Vybraná sada prediktorů, Složky křížové validace, Prediktory, Výsledky

Správný způsob



Texty: Vzorky, Vybraná sada prediktorů, Prediktory, Výsledky, Složky křížové validace

Bootstrap

- *Bootstrap* je flexibilní a mocný statistický nástroj, který se dá použít k posouzení nejistoty spojené s daným estimátorem nebo metodou statistického učení.
- Může například poskytnout odhad směrodatné chyby koeficientu nebo interval spolehlivosti pro ten koeficient.

Odkud pochází ten název?

- Použití termínu bootstrap (poutko na botách) se odvozuje z fráze *obout se do něčeho* (anglicky doslovně vytáhnout se za poutka na svých botách), o níž se soudí, že pochází z příběhu Rudopha Ericha Raspa „Podivuhodná dobrodružství barona Munchausena“ z osmnáctého století:

  Baron spadl na dno hlubokého jezera. Právě když to vypadalo, že vše je už ztraceno, napadlo ho vytáhnout se zpět za svá vlastní poutka u bot.
- Není to totéž jako termín „bootstrap“ používaný u počítačů a vztahující se k „bootování“ počítače pomocí řady primárních instrukcí, i když odvození je podobné.

Jednoduchý příklad

- Předpokládejme, že chceme investovat pevnou částku peněz do dvou finančních aktiv, která poskytují po řadě výnosy X a Y, kde X a Y jsou náhodné veličiny.
- Budeme investovat jistý podíl α našich peněz do X a zbývající část $1 - \alpha$ do Y.
- Přejeme si zvolit α tak, aby se minimalizovalo celkové riziko, neboli rozptyl, naší investice. Jinými slovy, chceme minimalizovat $\mathrm{Var}(\alpha X + (1 - \alpha) Y)$.
- Dá se ukázat, že hodnota, která minimalizuje riziko, je

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}},$$

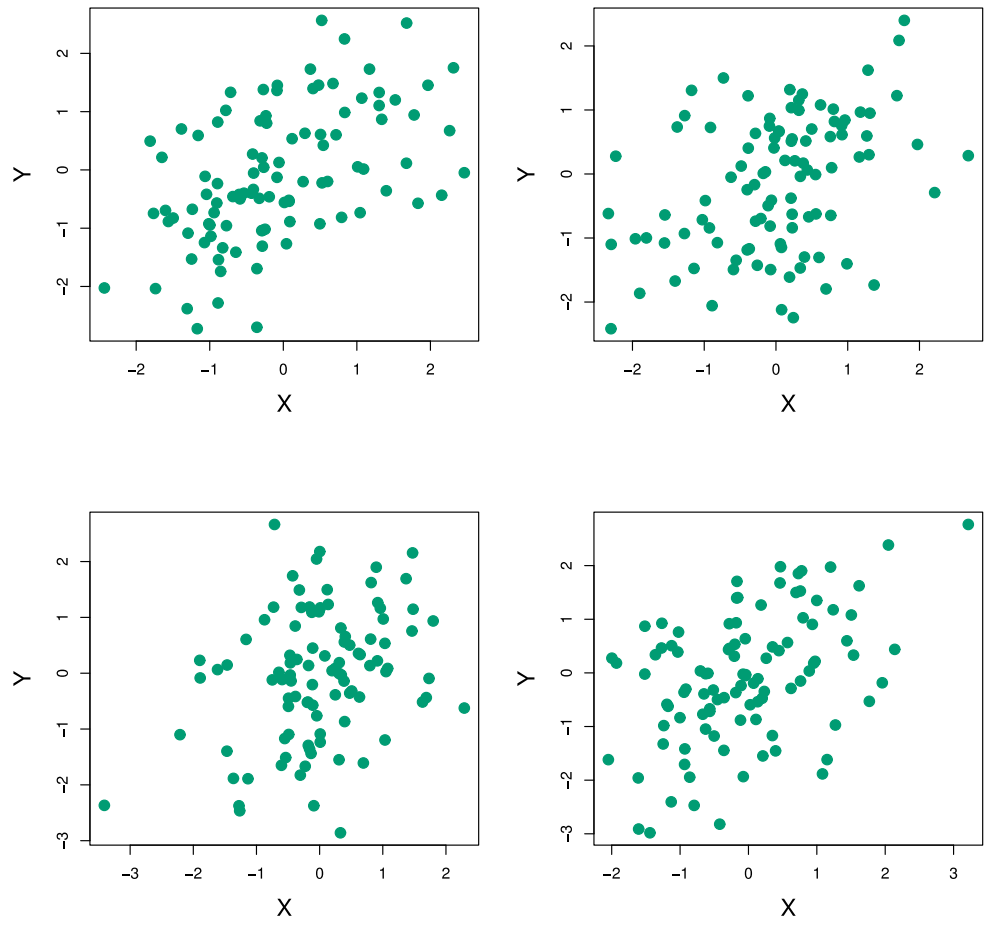
kde $\sigma_X^2 = \mathrm{Var}(X)$, $\sigma_Y^2 = \mathrm{Var}(Y)$, a $\sigma_{XY} = \mathrm{Cov}(X, Y)$.

Pokračování příkladu

- Ale hodnoty σ_X^2, σ_Y^2 a σ_{XY} nejsou známy.
- Můžeme vypočítat odhady těchto veličin, $\hat{\sigma}_X^2$, $\hat{\sigma}_Y^2$ a $\hat{\sigma}_{XY}$, pomocí souboru dat, který obsahuje měření X a Y.
- Pak můžeme odhadnout hodnotu α, která minimalizuje rozptyl naší investice, ze vztahu

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}.$$

Pokračování příkladu



Každý panel zobrazuje 100 *simulovaných výnosů pro investice* X *a* Y*. Zleva doprava a odshora dolů jsou výsledné odhady pro* α *rovny* 0,576, 0,532, 0,657, *a* 0,651.

Pokračování příkladu

- Abychom odhadli směrodatnou odchylku $\hat{\alpha}$, opakovali jsme proces simulace 100 dvojic pozorování X a Y a odhadování α 1000krát.
- Získali jsme tak 1000 odhadů α, které můžeme označit $\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{1000}$.
- Levý panel na obrázku na stránce 29 zobrazuje histogram výsledných odhadů.
- Pro tyto simulace byly parametry nastaveny jako $\sigma_X^2 = 1$, $\sigma_Y^2 = 1{,}25$ a $\sigma_{XY} = 0{,}5$, takže víme, že skutečná hodnota α je 0,6 (označeno červenou čarou).

Pokračování příkladu

- Střední hodnota přes všech 1000 odhadů α je

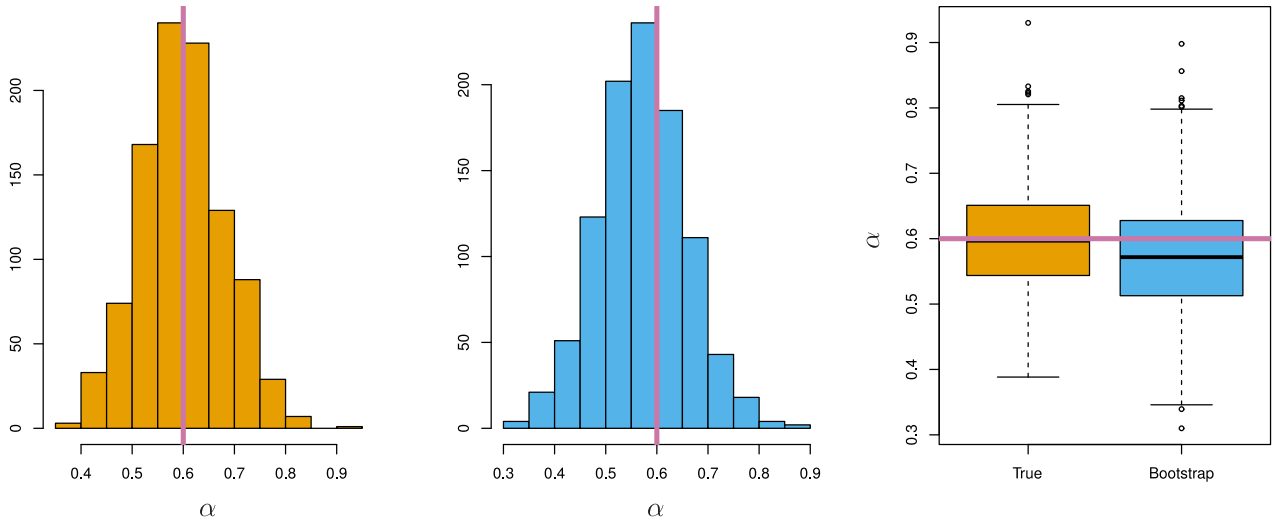
$$\bar{\alpha} = \frac{1}{1000} \sum_{r=1}^{1000} \hat{\alpha}_r = 0{,}5996,$$

velmi blízko k $\alpha = 0{,}6$, a směrodatná odchylka odhadů je

$$\sqrt{\frac{1}{1000-1} \sum_{r=1}^{1000} (\hat{\alpha}_r - \bar{\alpha})^2} = 0{,}083.$$

- To nám dává velmi dobrou představu o přesnosti $\hat{\alpha}$: $\mathrm{SE}(\hat{\alpha}) \approx 0{,}083$.
- Takže bychom zhruba řečeno u náhodného vzorku z populace čekali, že $\hat{\alpha}$ se bude lišit od α v průměru přibližně o 0,08.

Výsledky

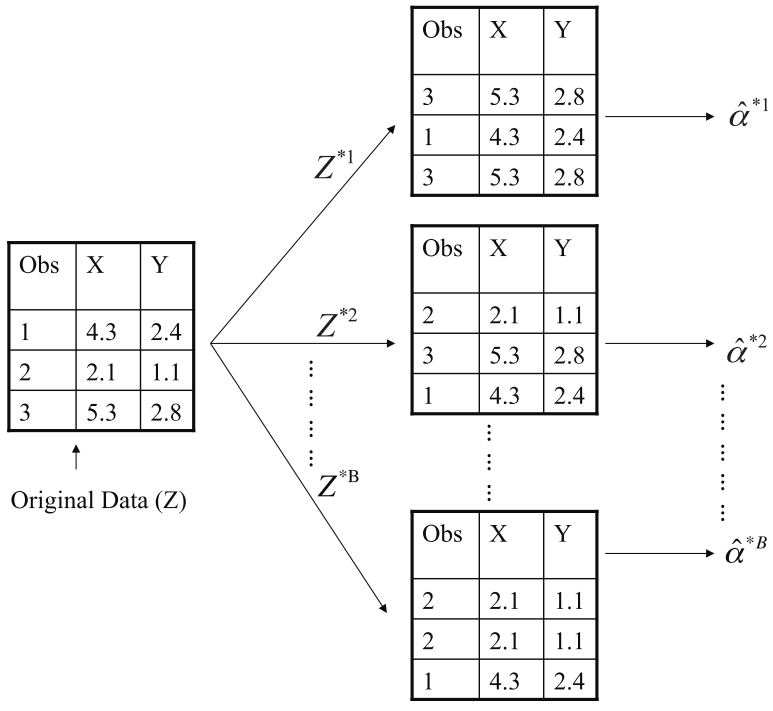


Vlevo: Histogram odhadů α získaných vygenerováním 1000 souborů simulovaných dat ze skutečné populace. *Uprostřed:* Histogram odhadů α získaných z 1000 vzorků z jediného souboru dat bootstrapem. *Vpravo:* Odhady α zobrazené v levém a prostředním panelu jsou zde znázorněny jako krabicové diagramy. Na každém panelu růžová úsečka označuje skutečnou hodnotu α.

Nyní zpět do skutečného světa

- Výše naznačený postup se nedá použít, protože pro reálná data nemůžeme generovat nové vzorky z původní populace.
- Avšak přístup zvaný bootstrap nám umožňuje použít počítač k napodobení procesu získávání nových souborů dat, takže můžeme odhadnout proměnlivost našeho odhadu, aniž bychom generovali dodatečné vzorky.
- Spíše než abychom z populace opakovaně získávali nezávislé soubory dat, získáváme místo toho různé soubory dat tak, že opakovaně vzorkujeme pozorování z původního souboru dat, a to *s vracením*.
- Každý z těchto „bootstrapových souborů dat“ se vytváří vzorkováním *s vracením* a má *stejnou velikost* jako náš původní datový soubor. V důsledku toho se některá pozorování mohou v daném bootstrapovém souboru dat objevit více než jednou a některá nemusí být zahrnuta vůbec.

Příklad s pouhými třemi pozorováními



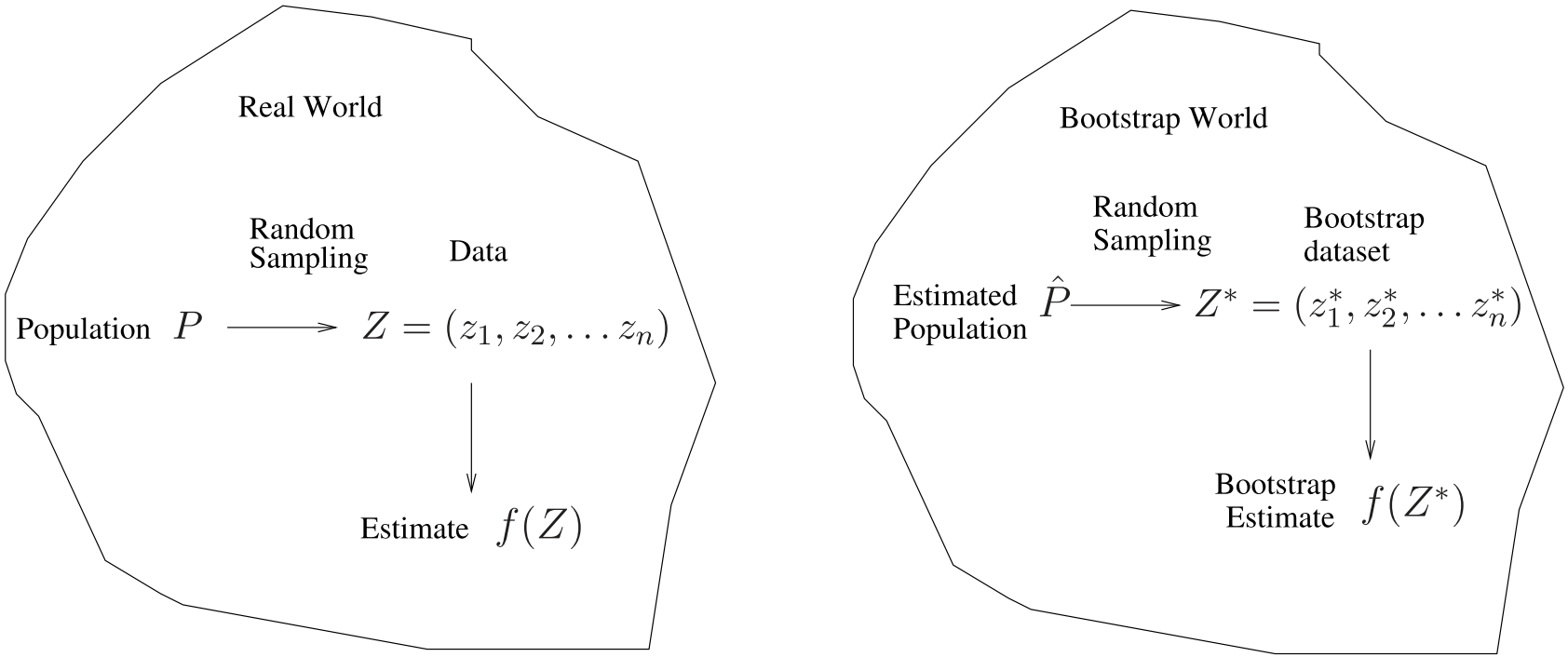
Grafická ilustrace myšlenky bootstrapu na malém vzorku obsahujícím $n = 3$ pozorování. Každý datový soubor bootstrapu obsahuje n pozorování, navzorkovaných s vracením z původního souboru dat. Každý bootstrapový soubor dat se použije k získání jednoho odhadu α.

- Označíme-li první bootstrapový soubor dat jako $Z^{\star 1}$, použijeme $Z^{\star 1}$ v bootstrapu k vytvoření nového odhadu α, který nazveme $\hat{\alpha}^{\star 1}$.
- Tento postup se opakuje B krát pro nějakou velkou hodnotu B (řekněme 100 nebo 1000), abychom tak získali B různých bootstrapových souborů dat, $Z^{\star 1}, Z^{\star 2}, \ldots, Z^{\star B}$, a B odpovídajících odhadů α: $\hat{\alpha}^{\star 1}, \hat{\alpha}^{\star 2}, \ldots, \hat{\alpha}^{\star B}$.
- Směrodatnou chybu těchto odhadů bootstrapu odhadneme pomocí vzorce

$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}(\hat{\alpha}^{\star r} - \bar{\hat{\alpha}}^{\star})^2}.$$

- Toto nám slouží jako odhad směrodatné chyby $\hat{\alpha}$, odhadnutého z původního souboru dat. Viz střední a pravý panel na obrázku na slajdu 29. Výsledky bootstrapu jsou modře. Pro tento příklad je $\mathrm{SE}_B(\hat{\alpha}) = 0{,}087$.

Obecný pohled na bootstrap



Vlevo: Skutečný svět, vpravo: Svět bootstrapu

Bootstrap obecně

- Přijít na vhodný způsob, jak generovat bootstrapové vzorky, může ve složitějších situacích s daty vyžadovat něco přemýšlení.
- Například je-li daty časová řada, nemůžeme jednoduše vzorkovat pozorování s vracením (*proč ne?*).
- Můžeme místo toho utvořit bloky po sobě jdoucích pozorování a vzorkovat je s vracením. Pak navzorkované bloky sesadíme dohromady a dostaneme tak soubor dat bootstrapu.

Jiná užití bootstrapu

- Primárně používán k získání směrodatných chyb nějakého odhadu.
- Poskytuje také přibližné intervaly spolehlivosti u populačních parametrů. Například, podíváme-li se na histogram v prostředním panelu obrázku na slajdu 29, pak 5% a 95% kvantily z 1000 hodnot dávají $(0{,}43, 0{,}72)$.
- To představuje přibližný 90% interval spolehlivosti pro skutečné α. *Jak interpretujeme tento interval spolehlivosti?*
- Výše uvedený interval se nazývá *percentilový interval* spolehlivosti bootstrapu. Je to nejjednodušší metoda (mezi mnoha jinými) pro získání intervalu spolehlivosti z bootstrapu.

Může bootstrap odhadnout chybu předpovědi?

- U křížové validace je každá z K validačních složek odlišná od zbývajících $K-1$ složek použitých pro trénování: *složky se nepřekrývají*. To je podstatné pro její úspěch. *Proč?*
- Abychom odhadli chybu předpovědi pomocí bootstrapu, mohlo by nás napadnout použít každý datový soubor bootstrapu jako náš tréninkový vzorek a původní vzorek jako náš validační vzorek.
- Ale každý vzorek bootstrapu se významně překrývá s původními daty. V každém bootstrapovém vzorku se objevují asi dvě třetiny původních datových bodů. *Můžete to dokázat?*
- To působí, že bootstrap vážně podhodnocuje skutečnou chybu předpovědi. *Proč?*
- Obrácený způsob — původní vzorek jako trénovací vzorek, soubor dat bootstrapu jako validační vzorek — je horší!

Odstranění překryvu

- Můžeme částečně tento problém napravit tím, že se používají pouze předpovědi pro ta pozorování, která se (náhodou) nevyskytla v daném vzorku bootstrapu.
- Ale metoda se stává složitou a nakonec křížová validace poskytuje jednodušší, atraktivnější přístup k odhadování chyb předpovědi.

Pre-validace

- V genomických studiích pomocí microarray a jiných je důležitým problémem porovnat prediktor vypuknutí choroby odvozený z velkého počtu „biomarkerů“ se standardními klinickými prediktory.
- Jejich porovnávání na stejném souboru dat, jaký byl použit k odvození biomarkerového prediktoru, může vést k výsledkům silně zkresleným ve prospěch biomarkerového prediktoru.
- K získání férovějšího porovnání těch dvou sad prediktorů se dá použít *pre-validace*

Motivační příklad

Příklad tohoto problému se objevil v článku van't Veera a dalších v časopise Nature (2002). V jejich microarray datech bylo měřeno 4918 genů v 78 případech vzatých ze studie rakoviny prsu. Je zde 44 případů ve skupině s dobrou prognózou a 34 ve skupině se špatnou prognózou. „Microarray" prediktor byl zkonstruován následovně:

1. Bylo vybráno 70 genů majících největší absolutní korelaci se 78 štítky tříd.
2. Pomocí těchto 70 genů byl zkonstruován klasifikátor $C(x)$ podle nejbližšího centroidu.
3. Použití tohoto klasifikátoru na daných 78 microarray dalo dichotomický prediktor $z_i = C(x_i)$ pro každý případ i.

Výsledky

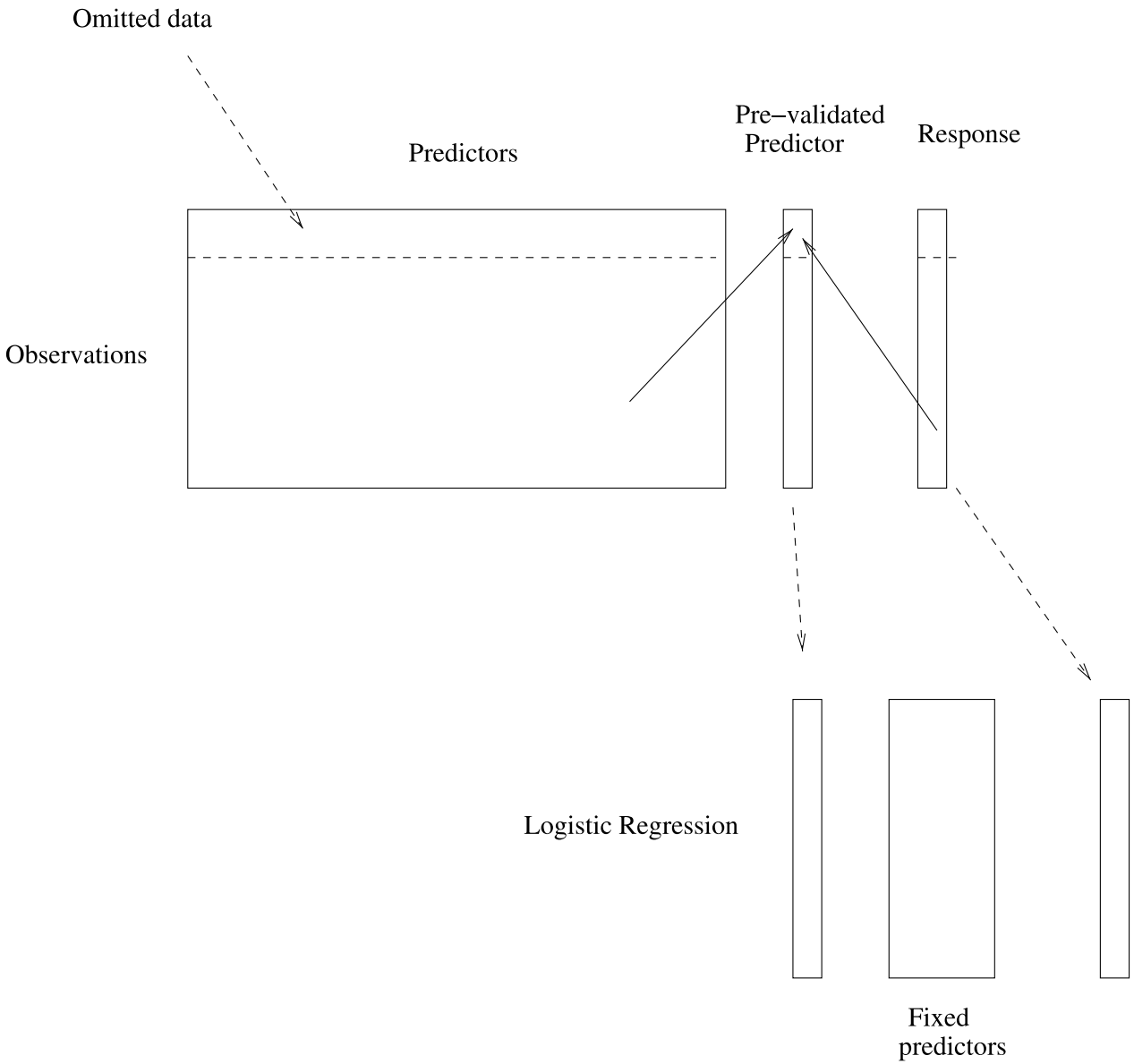
Porovnání microarray prediktoru s některými klinickými prediktory užívajícími logistickou regresi s výstupem `prognosis`:

Model	Coef	Stand. Err.	Z score	p-value
		Re-use		
microarray	4.096	1.092	3.753	0.000
angio	1.208	0.816	1.482	0.069
er	-0.554	1.044	-0.530	0.298
grade	-0.697	1.003	-0.695	0.243
pr	1.214	1.057	1.149	0.125
age	-1.593	0.911	-1.748	0.040
size	1.483	0.732	2.026	0.021
		Pre-validated		
microarray	1.549	0.675	2.296	0.011
angio	1.589	0.682	2.329	0.010
er	-0.617	0.894	-0.690	0.245
grade	0.719	0.720	0.999	0.159
pr	0.537	0.863	0.622	0.267
age	-1.471	0.701	-2.099	0.018
size	0.998	0.594	1.681	0.046

Myšlenka skrytá za pre-validací

- Navrženo pro porovnávání adaptivně odvozených prediktorů s pevnými, předdefinovanými prediktory.
- Myšlenka je vytvořit „pre-validovanou“ verzi adaptivního prediktoru: speciálně, „férovější“ verzi, která ještě „neviděla“ odpověď y.

Proces pre-validace



Pre-validace pro tento příklad detailně

1. Rozděl případy na $K = 13$ stejně velkých částí po 6 případech.
2. Odlož jednu část stranou. Pouze pomocí dat ze zbylých 12 částí vyber charakteristiky mající absolutní korelaci nejméně 0,3 se štítky tříd a utvoř klasifikační pravidlo nejbližšího centroidu.
3. Použij toto pravidlo k předpovědi štítků tříd pro třináctou část.
4. Opakuj kroky 2 a 3 pro každou ze 13 částí, čímž získáme „pre-validovaný“ microarray prediktor $\tilde{z}_i$ pro každý ze 78 případů.
5. Prolož model logistické regrese pre-validovaným microarray prediktorem a těmi 6 klinickými prediktory.

Bootstrap versus permutační testy

- Bootstrap vzorkuje z odhadnuté populace a používá výsledky k odhadům směrodatných chyb a intervalů spolehlivosti.
- Permutační metody vzorkují z odhadnutého *nulového* rozdělení pro data a používají to k odhadování p-hodnot a míry chybných zjištění při testování hypotéz.
- Bootstrap se dá použít v jednoduchých situacích k testování nulové hypotézy. Je-li například nulová hypotéza $\theta = 0$, kontrolujeme, zda interval spolehlivosti pro θ obsahuje 0.
- Lze také přizpůsobit bootstrap ke vzorkování z nulového rozdělení (viz knihu Efrona a Tibshiraniho „An Introduction to the Bootstrap“ (1993), kap. 16), ale není tu žádná skutečná výhoda nad permutacemi.