

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Studijní program: Kvantitativní metody v ekonomice

Studijní obor: Statisticko-pojistné inženýrství

Autor: Bc. Ondřej Lepša

Vedoucí diplomové práce: Mgr. Milan Bašta, Ph.D

**VYUŽITÍ BOOTSTRAPU A KŘÍŽOVÉ VALIDACE V ODHADU PREDIKČNÍ
CHYBY REGRESNÍCH MODELŮ**

Akademický rok 2014/2015

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze kterých jsem čerpal.

V Praze dne 7. ledna 2015

.....

Ondřej Lepša

Poděkování

Rád bych tímto poděkoval Mgr. Milanu Baštovi, Ph.D. za vedení mé diplomové práce a za podnětné návrhy, které tuto práci obohatily. Dále bych chtěl poděkovat Mgr. Jakubu Večeřovi za konzultace ohledně programovací části.

Abstrakt

Název práce: Využití bootstrapu a křížové validace v odhadu predikční chyby regresních modelů

Autor: Bc. Ondřej Lepša

Katedra: Katedra statistiky a pravděpodobnosti

Vedoucí práce: Mgr. Milan Bašta, Ph.D.

Nalezení modelu s dobrými předpovědními vlastnostmi je jeden z hlavních cílů regresní analýzy. I přesto se v běžné praxi k vyhodnocení predikčních schopností využívá kritérií, která se k tomuto účelu buď nehodí, nebo nejsou dostatečně spolehlivá. Alternativu představují relativně nové metody, které využívají opětovných simulací k odhadnutí vhodné ztrátové funkce – predikční chyby. Do této kategorie patří křížová validace a bootstrap. Tato práce popisuje, jak lze s využitím těchto metod vybrat takový regresní model, který co nejlépe předpovídá hodnoty vysvětlované proměnné.

Klíčová slova: bootstrap, křížová validace, výběr modelu, vícenásobná regrese, R

Abstract

Title: Utilizing Bootstrap and Cross-validation for prediction error estimation in regression models

Author: Bc. Ondřej Lepša

Department: Department of Statistics and Probability

Supervisor: Mgr. Milan Bašta, Ph.D.

Finding a well-predicting model is one of the main goals of regression analysis. However, to evaluate a model's prediction abilities, it is a normal practice to use criteria which either do not serve this purpose, or criteria of insufficient reliability. As an alternative, there are relatively new methods which use repeated simulations for estimating an appropriate loss function – prediction error. Cross-validation and bootstrap belong to this category. This thesis describes how to utilize these methods in order to select a regression model that best predicts new values of the response variable.

Keywords: Bootstrap, Cross-validation, model selection, multiple regression, R

Seznam obrázků

Obrázek 1: Rozdělení datového souboru metody Hold-out.....	13
Obrázek 2: Schéma K-násobné křížové validace	15
Obrázek 3: Schéma naivního bootstrapového odhadu.....	24
Obrázek 4: Odhad predikční chyby.	30
Obrázek 5: Složení predikční chyby z hlediska rozptylu a vychýlení.	32
Obrázek 6: Krabičkový graf odhadnutých optimálních rozměrů regresního modelu podle R^2	51
Obrázek 7: Krabičkový graf odhadnutých optimálních rozměrů regresního modelu podle R_{adj}^2	51
Obrázek 8: Krabičkový graf odhadnutých optimálních rozměrů regresního modelu podle AIC.	53
Obrázek 9: Krabičkový graf odhadnutých optimálních rozměrů regresního modelu podle BIC.	53
Obrázek 10: 5-násobná křížová validace pro $h = 1$	55
Obrázek 11: 10-násobná křížová validace pro $h = 1$	55
Obrázek 12: Křížová validace LOOCV pro $h = 1$	56
Obrázek 13: Vylepšený bootstrap pro $h = 1$	56
Obrázek 14: 5-násobná křížová validace pro $h = 2$	57
Obrázek 15: 10-násobná křížová validace pro $h = 2$	57
Obrázek 16: Křížová validace LOOCV pro $h = 2$	58
Obrázek 17: Vylepšený bootstrap pro $h = 2$	58

Seznam Tabulek

Tabulka 1: Odhad predikční chyby K-násobnou křížovou validací.	22
Tabulka 2: Parametry simulací	49
Tabulka 3: Skutečný optimální počet vysvětlujících proměnných.	49
Tabulka 4: Odmocnina z $MSEDim$ (klasická kritéria).	50
Tabulka 5: Odmocnina z $MSEDim$ (simulační kritéria).	54
Tabulka 6: Odhady predikční chyby v optimálním regresním submodelu.....	59

Obsah

1	Úvod.....	1
2	Klasické přístupy	2
2.1	Vymezení pojmů a značení	2
2.2	Úvod do problematiky.....	4
2.3	Metody selekce regresního submodelu	4
2.4	Klasická kritéria	6
3	Simulační Metody.....	9
3.1	Vymezení pojmů a značení	9
3.2	Odhad predikční chyby	10
3.3	Křížová validace.....	13
3.3.1	Hold-out křížová validace.....	13
3.3.2	K-násobná křížová validace.....	15
3.3.3	Leave-one-out křížová validace.....	17
3.3.4	Ukázka v programu R	18
3.4	Bootstrap	23
3.4.1	Naivní bootstrapový odhad predikční chyby	24
3.4.2	Vylepšený bootstrapový odhad predikční chyby.....	25
3.4.3	Ukázka v programu R	26
3.5	Počet validačních podmnožin a bootstrapových výběrů.....	28
3.5.1	Počet validačních podmnožin K-násobné křížové validace	28
3.5.2	Počet bootstrapových výběrů.....	28
3.6	Tréninková chyba a predikční chyba	29
3.7	Kompromis mezi rozptylem a vychýlením	30
4	Aplikace na datech.....	33
4.1	Úvod.....	33
4.2	Vymezení pojmů	34
4.3	Datový soubor	34
4.4	Simulační schéma.....	36
4.5	Sledované charakteristiky	37
4.6	Simulace v programu R.....	40
4.6.1	Parametry funkcí.....	42
4.6.2	Funkce odhadu predikční chyby	43
4.7	Výsledky simulací.....	49
4.7.1	Hledání optimálního regresního submodelu	49
4.7.2	Odhad predikční chyby a její MSE.....	54
5	Závěr	60

6	Přehled literatury a zdrojů.....	61
6.1	Knižní zdroje.....	61
6.2	Články	61
7	Přílohy.....	63

1 Úvod

Jedním z hlavních cílů regresní analýzy je sestavit takový regresní model, který by na základě datového souboru, který je v současné chvíli k dispozici, uměl předpovědět nové hodnoty vysvětlované proměnné. K posouzení predikčních schopností regresního modelu byla vytvořena kritéria, pomocí kterých se regresní modely vyhodnocují. Mezi v současné době často používaná kritéria se řadí koeficient determinace (R^2), korigovaný koeficient determinace (R^2_{adj}), Akaikeho informační kritérium (AIC) nebo Bayesovo informační kritérium (BIC). V případě většího počtu potenciálních modelů obsahujících různý počet proměnných lze pak jednoduše vybrat model s takovou hodnotou podle předem zvoleného kritéria, která reflektuje nejlepší predikční vlastnosti. Bohužel se ukazuje, že tato kritéria nejsou vždy schopna nalézt takový model. Častým důvodem je, že byla původně navržena pro jiný účel (např. popisné vlastnosti regresního modelu), anebo postrádají dostatečnou přesnost.

S rozvojem informačních technologií zaznamenaly rozvoj přístupy, které k odhadu predikčních schopností regresního modelu využívají opakovaných simulací. Mezi tyto metody se řadí křížová validace a bootstrap.

Cílem této práce je ukázat, že v současnosti používaná kritéria (R^2 , R^2_{adj} , AIC, BIC), která jsou běžně implementována ve statistickém software, nejsou vhodná pro nalezení regresního modelu z pohledu predikce, a naopak, že aplikací křížové validace nebo bootstrapu na předem zvolenou ztrátovou funkci lze docílit přesnějších výsledků. Tuto ztrátovou funkci nazýváme odhadem predikční chyby regresního modelu. Dalším cílem této práce je ukázat, jak lze jednoduše naprogramovat jednotlivé metody bootstrapu a křížové validace.

Práce je rozdělena na dvě části. V první části jsou popsána klasická kritéria pro výběr modelu a důvody, proč jsou tato kritéria v některých situacích nedostačující. Následuje představení simulačních přístupů, jejich konkrétních metod a ukázka implementace na jednoduché datové struktuře v programovém vybavení R.

V druhé části je věnována aplikaci metod bootstrapu a křížové validace na odhad predikční chyby pro vygenerovaná data a nalezení nejvhodnější metody k odhadu predikční chyby. Výsledky těchto metod jsou konfrontovány s výsledky získanými pomocí R^2 , R^2_{adj} , AIC a BIC.

2 Klasické přístupy

2.1 Vymezení pojmů a značení

Pro snazší porozumění obsahu je vhodné nadefinovat hned v úvodu pojmy, se kterými se v tomto textu pracuje.

Plný model je model, který využívá k odhadu regresní funkce všechny vysvětlující proměnné v datovém souboru. Submodel je model, který využívá k odhadu regresní funkce menší počet proměnných ze stejného datového souboru než plný model.

Tréninková podmnožina jsou pozorování, která jsou určena k odhadu regresní funkce. Validační podmnožina jsou pozorování, na kterých je ověřována predikční schopnost podle konkrétní metody.

Regresním modelem je v této práci chápán lineární regresní model aditivního typu. Tento model je určen teoretickou regresní funkcí značenou η . Tou je myšlena střední hodnota vysvětlované proměnné podmíněná kombinací hodnot vysvětlujících proměnných (Hebák a kol. 2005). Proměnné jsou v této práci značeny velkými písmeny, hodnoty, kterých proměnné nabývají, malými písmeny. Vysvětlovaná proměnná (v této práci je uvažována vždy pouze jedna) je značena Y . Vysvětlujících proměnných může být libovolný počet. Jsou značeny X_k , kde dolní index identifikuje konkrétní proměnnou. Hodnoty indexu k nabývají od jedné do K . Index i značí pořadí pozorování dané proměnné a nabývá hodnot od jedné do n , kde n je rozsah datového souboru. Regresní funkci zapisuji jako

$$\eta = E(Y|x_1, x_2, \dots, x_K). \quad (1)$$

Kromě regresní funkce jsou hodnoty vysvětlované proměnné závislé i na náhodné složce, která je značena ϵ . Náhodnou, respektive rušivou složkou, je myšlena náhodná veličina pocházející z normálního rozdělení se střední hodnotou nula a rozptylem rovným jedné. Jelikož je model aditivního typu, její hodnoty jsou přičítány k hodnotám regresní funkce. Regresní model je pak definován

$$y = \eta + \epsilon. \quad (2)$$

Malá písmena bez indexů značí vektory konkrétních proměnných.

Teoretickou regresní funkci se snažíme odhadnout nějakou konkrétní funkcí lineární v parametrech, například přímkou. Odhady regresních koeficientů jsou v této práci získávány vždy metodou nejmenších čtverců. Konstanta v modelu není uvažována. Odhady vysvětlované proměnné, též nazývané vyrovnané hodnoty, jsou značeny se stříškou. Vektor vyrovnaných hodnot je značen $\hat{y}$, jeho složky $\hat{y}_i$. Rozdíl mezi skutečnou hodnotou vysvětlované proměnné a vyrovnanou hodnotou odhadnuté regresní funkce je reziduum. Vektor reziduí je značen e , jeho složky e_i .

$$e = y - \hat{y}. \quad (3)$$

Celkový součet čtverců je myšlen součet čtvercových odchylek hodnot vysvětlované proměnné od její průměrné hodnoty. Značí se TSS (z angl. Total Sum of Squares) nebo $Q(y)$ (Hebák a kol. 2005, str. 87 a James a kol. 2013, str. 214).

$$TSS = Q(y) = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (4)$$

Regresním součtem čtverců je myšlen součet čtvercových odchylek vyrovnaných hodnot od průměrné hodnoty vysvětlované proměnné. Značí se ESS (z angl. Explained Sum of Squares) nebo $Q(\hat{y})$ (Hebák a kol. 2005, str. 87 a James a kol. 2013, str. 214).

$$ESS = Q(\hat{y}) = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 \quad (5)$$

Reziduálním součtem čtverců je myšlen součet čtvercových odchylek vyrovnaných hodnot od napozorovaných hodnot vysvětlované proměnné. Značí se RSS (z angl. Residual Sum of Squares) nebo $Q(e)$ (Hebák a kol. 2005, str. 87 a James a kol. 2013, str. 214).

$$RSS = Q(e) = \sum_{i=1}^n (e_i)^2 \quad (6)$$

Pomocí y_i^{bud} značím hodnoty vysvětlované proměnné z fiktivního datového souboru, který v současné chvíli není k dispozici. Na základě datového souboru, který je v tento moment k dispozici, se snažím sestavit takový regresní model, aby dokázal tyto nové hodnoty co nejlépe předpovědět.

Napříč touto prací se vyskytují pojmy, které pochází z anglicky psané literatury. V některých případech mají svůj český ekvivalent a univerzální značení (např. reziduální součet čtverců lze značit RSS nebo $Q(e)$). V některých případech ale český ekvivalent chybí. Z toho důvodu se při značení vzorců budu jednotně držet značení, které vychází z anglického názvosloví. Veškeré pojmy budou vysvětleny a anglické názvy přeloženy.

2.2 Úvod do problematiky

V regresní analýze je častou úlohou nalezení takového regresního modelu, na jehož základě by bylo možné učinit kvalitní předpověď pro nová pozorování. To si lze představit tak, že v datovém souboru hledáme kombinaci vysvětlujících proměnných, která by co nejlépe vystihla chování vysvětlované proměnné jak v tento moment, tak i pro případy, které teprve nastanou. Počet proměnných této kombinace je zpravidla menší než celkový počet vysvětlujících proměnných.

Pro volbu správné kombinace proměnných lze využít předem zvolených kritérií, které porovnávají jednotlivé submodely. Model, který dosáhne optimální (nejnižší nebo nejvyšší hodnoty v závislosti na typu kritéria) hodnoty daného kritéria je následně vyhodnocen jako nejlepší z posuzovaných modelů. Poměřovat všechny možné kombinace vysvětlujících proměnných může být ale výpočetně velmi náročné, zvláště při větším počtu vysvětlujících proměnných v datovém souboru. Z toho důvodu byly vyvinuty metody selekce modelu (resp. submodelu), které na základě předem určitého postupu vysvětlující proměnné do modelu buď přidávají, nebo je naopak z něj ubírají.

Otázku představuje volba správného kritéria, podle kterého je nejlepší submodel. Kromě v praxi často využívaných kritérií jako je například Akaikeho informační kritérium nebo koeficient determinace existují metody, které jako kritérium používají odhad určité ztrátové funkce. K tomu účelu využívají velkého počtu simulací. V následujících kapitolách budou rozebrána jak stávající, používaná kritéria, tak jejich protějšky využívající simulace.

2.3 Metody selekce regresního submodelu

Za předpokladu, že je k dispozici jedna vysvětlovaná proměnná a větší počet vysvětlujících proměnných, může být zajímavou otázkou, zda některé vysvětlující proměnné nemají větší vliv na vysvětlovanou proměnnou než jiné. Proces výběru vhodných proměnných se nazývá selekce regresního submodelu (z angl. model selection).

Jednou možností, jak k selekci regresního submodelu přistoupit, je vyhodnotit veškeré kombinace proměnných a vzájemně je porovnat s pomocí některého z kritérií (koeficient determinace - R^2 , korigovaný koeficient determinace - R_{adj}^2 , Akaikeho informační kritérium - AIC, Bayesovské informační kritérium - BIC). Tento způsob není příliš efektivní, protože takových kombinací je 2^K (James a kol. 2013, str. 78). Zatímco pokud jsou v datovém souboru dvě vysvětlující proměnné, lze mezi sebou porovnávat čtyři modely, pokud jich je v souboru deset, modelů k porovnání je 1024.

Za účelem snížení výpočetní náročnosti byly navrženy přístupy, které jsou v tomto ohledu úspornější. Jedná se například o takzvaný Shrinkage přístup. Do tohoto přístupu spadají metody hřebenové regrese nebo metoda lasso (James a kol. 2013, str. 216 - 231). Populární jsou metody přístupu Stepwise selekce, a to především Forward, Backward a jejich vzájemná kombinace. Jejich princip stručně přiblížím.

Metoda Forward (popředu - přeloženo z angl.) začíná s regresním modelem, který neobsahuje žádnou vysvětlující proměnnou. Pro všechny proměnné zvlášť je následně sledována změna příslušného kritéria při jejich zařazení do regresního modelu (AIC, BIC, R^2 , R_{adj}^2). Toto kritérium může být nahrazeno F-testy a do modelu jsou následně zařazovány pouze statisticky významné proměnné na předem zvolené hladině významnosti. Pokud přidání konkrétní vysvětlující proměnné zlepší hodnotu tohoto kritéria, je v modelu ponechána (James a kol. 2013, str. 209). Proces přidávání proměnných končí ve chvíli, kdy bylo dosaženo nejlepší hodnoty kritéria. Detailnější rozbor těchto kritérií a jejich interpretace je v následující kapitole.

Metoda Backward (pozadu - přeloženo z angl.) začíná s regresním modelem, který naopak obsahuje všechny vysvětlující proměnné. V každém kroku je následně vyřazena proměnná, jejíž vyřazení nejvíce zlepší hodnotu sledovaného kritéria nebo která je dle F-testu nejméně statisticky významná (Hebák a kol. 2005, str. 112). Takto se pokračuje, dokud lze vyřazením některé z proměnných kritérium zlepšit nebo dokud model obsahuje statisticky nevýznamné proměnné.

Další možností je kombinace obou přístupů. Tato metoda, někdy nazývaná hybridní (James a kol. 2013, str. 212), je charakterizována tím, že po přidání nové proměnné do modelu jinou proměnnou může ubrat, pokud příspěvek této proměnné v nově vzniklém regresním modelu už dané kritérium nezlepšuje. I zde může být kritérium nahrazeno F-testy.

Kritériím, pomocí kterých je každý ze sekvenční regrese modelů vyhodnocován, je věnována následující kapitola. Ty představují pouze jeden z přístupů, jak zhodnotit regresní model.

2.4 Klasická kritéria

Jedním z nejběžněji používaných kritérií, které pro regresní model aditivního typu lze využít, je koeficient determinace R^2 . Tento ukazatel lze velmi jednoduše zkonstruovat. V čitateli se nachází regresní součet čtverců, ve jmenovateli celkový součet čtverců.

$$R^2 = \frac{ESS}{TSS} \quad (7)$$

Jmenovateli se někdy říká celková variabilita, čitateli vysvětlená variabilita. Koeficient determinace lze totiž rozepsat stejně jako ve vzorci (8).

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2 / n}{\sum_{i=1}^n (y_i - \bar{y})^2 / n} \quad (8)$$

Z toho je patrné, že jmenovatel není nic jiného než rozptyl hodnot vysvětlované proměnné. Čítec udává rozptyl odhadnutých, vyrovnaných, hodnot od stejné průměrné hodnoty. A pojem variabilita je synonymem pojmu rozptyl. Vysvětlená potom znamená, že je vysvětlená regresní funkcí, tedy jak dobře se regresní funkce přibližuje skutečnosti. Vzorce (7) a (8) platí pro regresi s konstantou v modelu. Pro případ bez konstanty je používán vzorec (9).

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i)^2}{\sum_{i=1}^n (y_i)^2} \quad (9)$$

V modelu s konstantou nabývá koeficient determinace R^2 od nuly do jedné. V modelu bez konstanty může nabýt i hodnoty větší než jedna. Interpretuje se tak, že čím více se jeho hodnota přibližuje jedné, tím je větší podíl vysvětlené variability a tím lépe zvolený regresní model popisuje aktuální data.

Koeficient determinace má své nedostatky. Významným nedostatkem je, že se hodnota koeficientu determinace nikdy nesníží se zařazením nové vysvětlující proměnné do už existujícího regresního modelu (Greene 2003). Z toho vyplývá, že koeficient determinace upřednostňuje modely s vyšším počtem proměnných. (Takové modely nemusí být nutně na škodu, ale s rostoucím počtem proměnných je interpretace značně ztížena. Tato skutečnost komplikuje vzájemné porovnávání regresních modelů. Částečné řešení se nabízí použitím korigovaného koeficientu determinace R_{adj}^2 (10).

$$R_{adj}^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2 / n - K}{\sum_{i=1}^n (y_i)^2 / n - 1} \quad (10)$$

K odráží počet proměnných v modelu, n je počet pozorování. R_{adj}^2 vzroste jen v případě, pokud vzroste průměrná vysvětlená variabilita vzhledem k počtu vysvětlujících proměnných zařazených do modelu (James a kol. 2013, str. 214). Tím dochází k tomu, že pokud jsou do modelu přidávány nadbytečné proměnné, hodnota korigovaného koeficientu determinace by neměla vzrůst.

Dalším známým kritériem, se kterým lze vzájemně porovnávat regresní modely, je *Akaikeho Informační Kritérium* (AIC). Způsobů, jak toto kritérium vyjádřit, existuje více. V úlohách regresní analýzy a ve statistickém software je často používán výraz (11).

$$AIC = \log \left(\frac{RSS}{n} \right) + \frac{2K}{n} \quad (11)$$

Někdy bývá člen uvnitř logaritmu nahrazen věrohodnostní funkcí a celý logaritmus vynásoben -2 (Efron, Tibshirani 1993, str. 242 a Greene 2003, str. 160).

Kritérium AIC vychází z teorie informace. Velmi obecně ho lze chápat jako míru informace ztracené tím, že se snažíme popsat náhodnou veličinu regresní funkcí. Tuto ztrátu kvantifikuje první člen výrazu (11). Stejně jako korigovaný koeficient determinace i Akaikeho informační kritérium zavádí penalizaci modelů s větším počtem vysvětlujících proměnných (druhý člen ve výrazu (11)). Tato penalizace zvyšuje výslednou hodnotu kritéria s rostoucím počtem proměnných (K). Kritérium tedy hledá kompromis mezi ztrátou

informace a počtem vysvětlujících proměnných v regresním modelu (James a kol. 2013, str. 214).

Posledním klasickým kritériem, kterému se v této práci věnuji, je *Bayesovo Informační Kritérium (BIC)*. To je podobné AIC, ale s rozdílem v předpokladech jeho odvození, kterými se zde nebudu zabývat. Rozdíl mezi AIC a BIC je ve výsledku takový, že BIC uplatňuje větší penalizaci modelů s větším počtem proměnných. Zatímco AIC používá hodnotu 2, BIC používá logaritmus počtu pozorování (James a kol. 2013, str. 214).

$$BIC = \log\left(\frac{RSS}{n}\right) + \frac{\log(n)K}{n} \quad (12)$$

Pro AIC i BIC platí, že ze skupiny regresních modelů vybíráme ten, který má nejnižší hodnotu kritéria AIC nebo BIC, tedy nejnižší informační ztrátu vzhledem k počtu vysvětlujících proměnných v modelu.

Výše uvedené statistiky jsou plně využívány statistickými softwary a s výjimkou koeficientu determinace se jedná o relativně spolehlivá kritéria pro výběr správného popisného modelu (James a kol. 2013, str. 215). Nicméně jejich použití podléhá jistým omezením. Pro jejich aplikaci je zpravidla nezbytné, aby model splňoval požadavky klasického lineárního modelu (Hebák a kol. 2005, str. 40):

- Vysvětlující proměnné jsou nenáhodné a bez funkční závislosti
- Na vektor regresních koeficientů nejsou kladena žádná omezení
- Hodnoty náhodné složky ϵ_i pochází z normálního rozdělení se střední hodnotou $\mu_i = 0$ a rozptylem $\sigma_i^2 = 1$.

V případě hledání regresního submodelu z hlediska predikce už tato kritéria neposkytují tak přesné výsledky. Daná kritéria totiž spíše než predikční schopnost hodnotí to, jak moc dobře daný model vystihuje všechna data. Detailně jsou této problematice věnovány kapitoly 3.6 a 3.7.

Nedostatky těchto metod vedly k vývoji simulačních metod, pomocí kterých by šlo přesněji odhadnout predikční schopnosti modelů. Do této skupiny se řadí i simulační metody, jež jsou hlavním tématem mé práce – bootstrap a křížová validace.

3 Simulační Metody

3.1 Vymezení pojmů a značení

Nad rámec značení vymezeného v kapitole 1.1 je potřeba vymezit některé další pojmy pro lepší porozumění následujícího textu. V kapitole 1.1 jsem hovořil o tom, že některé pojmy pocházející z anglicky psané literatury nemají v češtině svůj ekvivalent. V případě křížové validace se jedná přímo o názvosloví spojené s jednotlivými metodami tohoto teoretického přístupu. Jedná se o metody Hold-out, K-Fold a LOOCV. Vysvětlení názvů metod je vysvětleno přímo v textu. V případě metod Hold-out a LOOCV si nejsem vědom toho, že existuje český ekvivalent. Odhady predikční chyby pomocí těchto metod jsou značeny jako PE_{HO} nebo PE_{LOOCV} . U metody K-Fold Čeština umožňuje překlad na K-násobná křížová validace. Odhady predikční chyby K-násobnou křížovou validací jsou kvůli konzistenci zachovány v angličtině (PE_{K-Fold}).

V rámci křížové validace dochází k rozdělování datového souboru na dvě podmnožiny: tréninkovou a validační. Pozorování v jednotlivých podmnožinách je nutné od sebe odlišit. Pomocí n se značí celkový počet pozorování v datovém souboru. Označení počtu pozorování ve validační podmnožině se liší podle aplikované metody. Pro metodu Hold-out používám značení n_h , pro metodu K-násobná n_k . Pro metodu LOOCV není nutné značení zavádět, protože se vždy vypouští pouze jedno pozorování. Tréninková podmnožina je doplněk validační podmnožiny do n pozorování. Mínus v indexu označuje tu část, jež byla z původního datového souboru vypuštěna a jež představuje validační podmnožinu.

Vyrovnané hodnoty ve smyslu křížové validace nabývají trochu odlišného významu. Regresní koeficienty modelu se odhadují na tréninkové podmnožině. Vyrovnané hodnoty, které značím $\hat{y}_i$, jsou získány aplikací regresních koeficientů (získané na tréninkové podmnožině) na data validační podmnožiny.

Hodnoty vysvětlované proměnné ve validační podmnožině jsou rozlišovány podle aplikované metody. Pro metodu Hold-out používám značení y_i^{-h} , pro metodu K-Fold (K-násobná) y_i^{-k} . V případě metody LOOCV se vypouští pouze jediné pozorování, které je označeno y_i^{-i} .

Stejně jako u křížové validace i zde je nutné vymezit některé dodatečné pojmy, se kterými metody bootstrapu pracují.

Malé n značí rozsah datového souboru, se kterým pracujeme. Bootstrapové výběry mají stejný rozsah. Bootstrapových výběrů je celkem B . Hodnoty vysvětlované proměnné v datovém souboru jsou značeny y_i , hodnoty vysvětlované proměnné v bootstrapových výběrech jsou značeny y_i^b . Index b označuje konkrétní bootstrapový výběr a nabývá hodnot od 1 do B . Vyrovnané hodnoty, získané aplikací regresních koeficientů na původní datový soubor, jsou značeny $\widehat{y}_{b,i}$. Vyrovnané hodnoty, získané aplikací regresních koeficientů na bootstrapový výběr, jsou značeny $\widehat{y}_{b,i}^b$.

Názvosloví odhadů predikční chyby pomocí metod bootstrapu je z důvodu konzistence zachováno v angličtině (např. PE_{naive} , $PE_{improved}$).

3.2 Odhad predikční chyby

Kritéria, která jsem zmínil v předchozí kapitole, slouží tedy k především k hodnocení modelu jako takového, tj. jejich účel je hodnotit popisnou schopnost modelu. Hodnotící hledisko tkví ve shodě odhadnutých hodnot a těch skutečných, v některých případech i v počtu proměnných zařazených do modelu. Pro potřeby této diplomové práce je ale relevantním kritériem především predikční schopnost modelu, tj. na základě existujících dat sestavit model, podle kterého jsme schopni učinit předpověď pro nová pozorování.

Uvažuji datový soubor, obsahující jednu vysvětlovanou proměnnou a několik vysvětlujících proměnných. Tento datový soubor je k dispozici nyní. Dále uvažuji, že mám k dispozici regresní model určený k predikci, jehož vyrovnané hodnoty jsou $\hat{y}_i$. Nabízí se otázka s čím porovnat tyto hodnoty, aby byla predikční schopnost modelu nejlépe zhodnocena. Jednou možností je například hodnotit úhrn odchylek vyrovnaných hodnot od skutečných (13).

$$RSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (13)$$

To není nic jiného než reziduální součet čtverců, který byl zmiňován v předchozí kapitole. Dalo by se tedy říct, že reziduální součet čtverců měří určitou ztrátu informace (stejně jako AIC nebo BIC). V případě RSS je měřena celková ztráta informace přes veškerá pozorování. Často se vzorec (13) dělí počtem pozorování n , a potom je tento odhad nazýván

jako tzv. tréninková chyba (Hastie a kol. 2009, str. 221). Více o tréninkové chybě v kapitole 3.6.

Způsobů, jak vyjádřit tuto ztrátu informace, je celá řada a obecně se funkcím, které ji kvantifikují, říká ztrátové funkce (Hastie a kol. 2009, str. 219). Výraz pochází z anglického loss function. Závorka ve vzorci (13) představuje kvadratickou ztrátovou funkci a lze ji nahradit například absolutní ztrátovou funkcí (14).

$$\sum_{i=1}^n |y_i - \hat{y}_i| \quad (14)$$

Nicméně v této práci využívám kvadratickou ztrátovou funkci, konkrétně její průměrnou hodnotu na jedno pozorování. Obecně se této charakteristice se také říká střední čtvercová odchylka (z angl. mean squared error) a značí se MSE. Střední čtvercová odchylka vyrovnaných hodnot regresního modelu od napozorovaných hodnot vysvětlované proměnné je uvedena v (15).

$$MSE(\hat{y}) = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} = E(y_i - \hat{y}_i)^2 \quad (15)$$

Zbývá vyřešit, zda je správně porovnávat vyrovnané hodnoty regresního modelu $\hat{y}_i$ se současnými, napozorovanými hodnotami vysvětlované proměnné y_i . Tento postup by byl správný při nalezení takového regresního modelu, který co nejlépe vystihl současné chování vysvětlované proměnné. V případě posouzení predikčních schopností by kritérium bylo značně optimistické, protože schopnost predikce by byla hodnocena popisnou schopností modelu. Vhodnější by bylo vyrovnané hodnoty regresního modelu porovnat s budoucími hodnotami vysvětlované proměnné y_i^{bud} .

$$MSE(\hat{y}) = \frac{\sum_{i=1}^n (y_i^{bud} - \hat{y}_i)^2}{n} \quad (16)$$

Vzorec (16) je v literatuře nazýván odhadem predikční chyby regresního modelu (Efron, Tibshirani 1993). Název pochází z anglického označení prediction error, a proto je tento odhad někdy označován PE. Predikční chyba je počítaná jako střední čtvercová

odchylka rozdílu mezi budoucími hodnotami vysvětlované proměnné y_i^{bud} a vyrovnanými hodnotami současného regresního modelu $\hat{y}_i$. Čím menší je její hodnota, tím lepší je regresní model z pohledu predikce. Otázku představuje, co dělat v případě, že nedisponuji novým datovým celkem. Tento problém se snaží řešit teoretické přístupy bootstrap a křížová validace, které k odhadu predikční chyby využívají odlišný postup.

Základní charakteristiky Cross-validation neboli křížové validace, lze vystopovat až do 30. let 20. století (Arlot, Celisse 2010). Základní myšlenkou tohoto přístupu je rozčlenění datového souboru na tréninkovou podmnožinu a validační podmnožinu. Bootstrap už spadá do kategorie moderních přístupů a první zmínky o něm pocházejí ze 70. let 20. století. Podstata spočívá v opakovaných náhodných výběrech s vrácením z datového souboru.

Obecně řečeno, jedná se především o datově orientované přístupy, které nekladou tak velký důraz na pravděpodobnostní předpoklady regresních modelů a jejich síla leží především v jejich jednoduchosti a snadné aplikaci. Zároveň je nutné podotknout, že uplatnění těchto přístupů není limitované pouze regresní analýzou. Nacházejí uplatnění v různých oblastech statistiky, využívají se například u Metody hlavních komponent. Jejich podstata je detailněji vysvětlena v následujících oddílech.

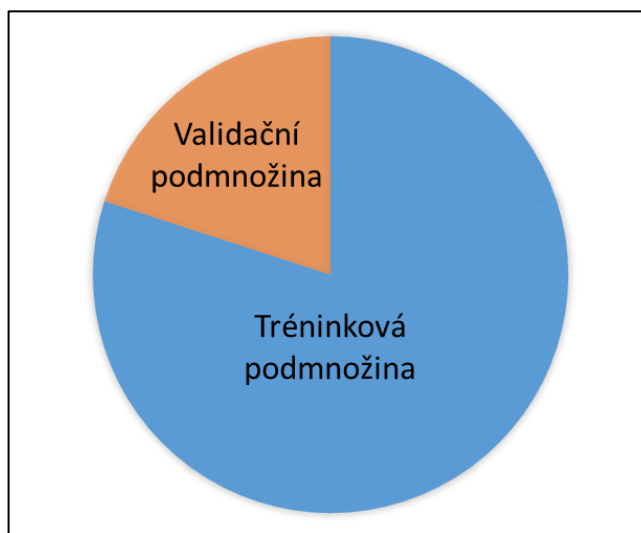
3.3 Křížová validace

Křížová validace tedy představuje skupinu metod, které využívají v jádru velmi podobný přístup. Jak už bylo zmíněno, její podstata tkví v rozdělení datového souboru na dvě podmnožiny: tréninkovou a validační (testovací). Tréninková podmnožina dat slouží k odhadnutí regresních koeficientů modelu, na validační podmnožině je proveden odhad predikční chyby. Díky tomu, že se validační podmnožina nepodílí na odhadu koeficientů modelu, je možné ji považovat za nová data. Tento úsporný postoj k datům je výhodný, protože získat dostatečné množství dat bývá zpravidla nákladné.

Variací na metody křížové validace existuje velký počet. Mezi často zmiňované metody patří Hold-out, K-Fold, Leave-one-out. Mezi náročnější verze křížové validace náleží Leave-P-Out, Repeated Learning Testing, Monte Carlo CV nebo Generalized CV (Arlot, Celisse 2010). V rámci této práce jsem pozornost věnoval prvním třem zmíněným typům metody křížové validace.

3.3.1 Hold-out křížová validace

Základní metodou křížové validace je tzv. metoda Hold-out. Podle obecného principu křížové validace je datový soubor rozdělen na dvě vzájemně se nepřekrývající podmnožiny: tréninkovou a validační (Liu 2009).



Obrázek 1: Rozdělení datového souboru metody Hold-out. Zdroj: (James a kol. 2013)

Validační podmnožině se v angličtině říká Hold-out část. Princip vychází z toho, že jsou tato data jakoby vyjmuta (výraz Hold-out by mohl být z angličtiny přeložen jako „zdržet se něčeho“) a nevyužívají se k odhadu koeficientů regresního modelu.

Regresní koeficienty modelu jsou tedy odhadnuty na tréninkové podmnožině. Následně jsou poté aplikovány na validační podmnožinu za účelem získání vyrovnaných hodnot $\hat{y}_i$. Odhad predikční chyby, tj. predikční schopnost modelu je následně určena vzorcem (17).

$$PE_{HO} = \frac{\sum_{i=1}^{n-h} (y_i^{-h} - \hat{y}_i)^2}{n-h} \quad (17)$$

Velikost podmnožin nemusí být v poměru jedna ku jedné, rozdělení dat závisí čistě na tom, kdo provádí statistickou analýzu. Je však nutné brát v potaz, že nevhodné rozdělení dat může vést k zavádějícím výsledkům. V případě zahrnutí většiny pozorování (například 90%) do tréninkové podmnožiny, může být odhad predikční chyby menší, než je skutečná predikční schopnost dat.

Další problematickou oblast metody Hold-out je výběr dat do obou podmnožin. Původně se v případě křížové validace nepočítalo s opakovanými simulacemi, a tak se metoda prováděla pouze na jednou rozdělených datech. Tímto mohlo dojít ke vzniku dvou naprosto nekorespondujících podmnožin. Při opakování výpočtu dochází k tomu, že pro například pět různých náhodných výběrů je získáno pět výrazně odlišných hodnot PE_{HO} (James a kol 2013, str. 179). Za takových okolností není jasné, který výsledek je správný a jestli je použitá metoda užitečná.

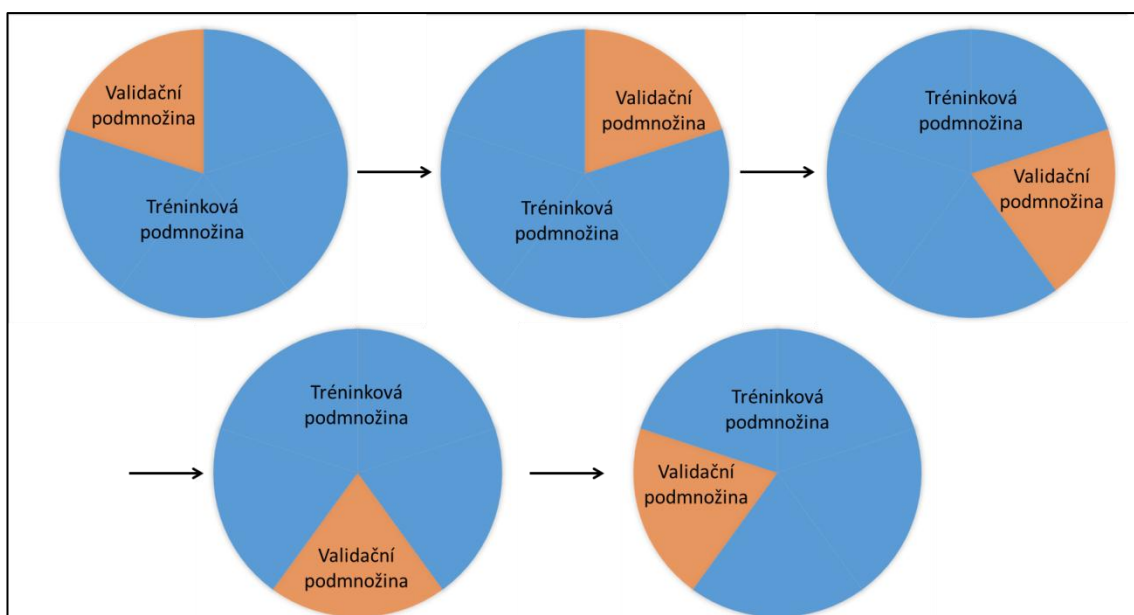
Možné řešení tohoto problému spočívá ve využití statistického softwaru a použití opakovaného simulování ze souboru, tzv. random subsampling (Kohavi 1995, str. 2). Takto je metoda Hold-Out opakována Z-krát a za výslednou predikční chybu se bere průměrná hodnota Z opakování (18).

$$\overline{PE}_h^{RS} = \frac{\sum_{i=1}^Z PE_{HO}^i}{Z} \quad (18)$$

Ani tato korekce nemusí vést k vylepšení odhadu, protože není zaručeno, že při opakování náhodného výběru pozorování do tréninkové (resp. validační) podmnožiny nedojde k opakovanému vynechání některých hodnot. Takové opomenutí má vliv na přesnost odhadu predikční chyby. Tento problém se snaží řešit metody křížové validace K -násobná a Leave-one-out.

3.3.2 K -násobná křížová validace

Momentálně jedna z nejdoporučovanějších metod odhadu predikční chyby se nazývá K -násobná křížová validace. Název pochází z anglického K -Fold. Podstata této metody spočívá v tom, že se datový soubor nejprve rozdělí do K vzájemně nepřekrývajících se podmnožin.¹



Obrázek 2: Schéma K -násobné křížové validace (ilustrováno na 5-násobné křížové validaci). Zdroj: (James a kol. 2013)

Na $K - 1$ množinách je odhadnut regresní model a na K -té množině se ověří jeho predikční schopnost odhadem predikční chyby PE (James a kol. 2013, str. 183). Při odhadu regresního modelu je důležité si uvědomit, že ačkoliv data dělíme na K částí, model je vždy odhadován ze všech pozorování $K - 1$ podmnožin. V každém kroku tedy figurují dvě podmnožiny, stejně jako u metody Hold-out: tréninková a validační. Postup je opakován K -krát, tedy dokud se každá z podmnožin neocitne v roli validační.

¹ Nezaměňovat K s počtem proměnných. V tomto případě K odpovídá počtu validačních podmnožin.

Predikční chyba každého kroku odhadována stejně jako u metody Hold-out.

$$PE_{-k} = \frac{\sum_{i=1}^{n-k} (y_i^{-k} - \hat{y}_i)^2}{n-k} \quad (19)$$

Rozdíl oproti předchozí zmíněné metodě je ten, že odhad predikční chyby je průměrná hodnota predikčních chyb v jednotlivých krocích PE_{-k} (Efron, Tibshirani 1993, str. 240).

$$PE_{K-Fold} = \frac{\sum_{k=1}^K (PE_{-k})}{K} \quad (20)$$

Důležitou součástí je rozložení pozorování do jednotlivých podmnožin. Tato otázka vyvstává především v případech, ve kterých pozorování v datovém souboru podléhají nějakému systému, například pokud jsou hodnoty proměnných seřazené od nejmenšího k největšímu. To nebyl případ u Hold-out, protože tato metoda vycházela z náhodného výběru pozorování z datového souboru do tréninkové a validační. V případě K-násobné křížové validace je vhodné, aby pozorování v jednotlivých podmnožinách byla stratifikována, tj. aby jejich zastoupení v množinách bylo co nejvíce reprezentativní (Liu 2009). Další otázku představuje, jak správně rozdělit pozorování do jednotlivých podmnožin v případě, že n není dělitelné K . Vhodné řešení je rozdělit pozorování tak, aby se podmnožiny navzájem lišily maximálně o jedno pozorování. Pokud by $n = 11$ a $K = 5$, pozorování by byla rozdělena tak, že čtyři podmnožiny obsahují dvě pozorování, jedna podmnožina obsahuje tři. Pro $n = 12$, tři podmnožiny obsahují dvě pozorování, dvě podmnožiny tři pozorování atd.

Metoda je v současné době velmi populární, a to především proto, že oproti ostatním metodám (např. Hold-out nebo LOOCV) je stále relativně výpočetně nenáročná (v praxi se používá maximálně desetinásobná, tj. $K = 10$) zároveň produkuje nejméně vychýlené odhady (Mevik, Cederkvist 2005).

3.3.3 Leave-one-out křížová validace

Poslední metoda křížové validace, se kterou jsem pracoval v této práci, je tzv. Leave-one-out (LOOCV z angl.. Leave One Out Cross Validation). Název, který v češtině znamená „vypustit jedno“ naznačuje princip tohoto přístupu. Datový soubor o rozsahu n je opět rozdělen na dvě podmnožiny, tréninkovou a validační, s tím rozdílem oproti K -násobné křížové validaci, že tréninková podmnožina obsahuje $n - 1$ (proto „vypustit jedno“) a podmnožina validační obsahuje zbylé jediné pozorování (Efron, Tibshirani 1993, str. 240). Postup je opakován n -krát tak, že v prvním opakování je vynechané první pozorování, v druhém druhé a postupně až nakonec v n -tém opakování je vynecháno n -té. Schéma tohoto postupu je shodné se schématem K -násobné křížové validace v předchozí kapitole.

Následný postup je shodný s předchozími metodami. Na tréninkovou podmnožinu je aplikován regresní model za účelem odhadu regresních koeficientů. Validační podmnožina, v tomto případě jediné pozorování, slouží k výpočtu vyrovnané hodnoty a odhadu predikční chyby modelu. V každém kroku je vypočtena predikční chyba PE_{-i} (James a kol. 2013, str. 181).

$$PE_{-i} = (y_i^{-i} - \hat{y}_i)^2 \quad (21)$$

Index $-i$ odpovídá i -tému pozorování, které bylo z datového souboru vypuštěno a které reprezentuje validační část. Oproti Hold-out a K -násobné křížové validace ze vzorce predikční chyby jednoho kroku odpadl jmenovatel, protože validační podmnožina obsahuje pouze jediné pozorování. Odhadů predikční chyby v jednotlivých krocích je tedy n . Celková predikční chyba je opět průměrnou hodnotou predikčních chyb PE_{-i} .

$$PE_{\text{LOOCV}} = \frac{\sum_{i=1}^n PE_{-i}}{n} \quad (22)$$

LOOCV poskytuje několik výhod. Model je odhadnutý na téměř celém datovém souboru, takže problematika zařazení hodnot do tréninkové a validační množiny nehraje prakticky roli. Zároveň je tím zajištěno, že při opakování metody LOOCV je dosaženo vždy totožných výsledků, protože volba pozorování už není náhodná, ani subjektivní, ale systematická.

Odhady predikční chyby metodou LOOCV jsou téměř nevychýlené (Mevik, Cederkvist 2005). Naopak v porovnání s například metodou K-násobná je rozptyl odhadu predikční chyby metodou LOOCV značně vysoký (Liu 2009). Přesnost odhadů metodou LOOCV lze díky tomu diskutovat. Další nedostatek je výpočetní náročnost v případě souborů s velkým počtem pozorování. Pro např. $n = 1000$ je proces opakován tisíckrát a značně tím prodlužuje výpočetní čas i za využití statistických softwarů. Kdyby se k tomu mělo ještě připojit hledání optimálního regresního submodelu (např. *Stepwise* metodami), je výpočetní čas výrazně delší. Tento problém lze naštěstí překonat aplikováním modifikovaného vzorce odhadu predikční chyby pro metodu LOOCV (James a kol. 2013).

$$PE_{LOOCV} = \frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{1 - h_i} \right)^2 \quad (23)$$

K určení regresního modelu se využijí veškerá data. Odhad predikční chyby se upraví vydělením $1 - h_i$. Proměnná h_i v literatuře je nazývána jako efekt (anglicky pak leverage) a zastupuje diagonální prvky projekční matice H .

Projekční matice je v regresi používána jako nástroj k posouzení vlivu jednotlivých pozorování na odhady regresních koeficientů modelu (Hebák a spol. 2005, str. 94-95). Pokud je matici pozorování vysvětlujících proměnných definována jako $\mathbf{X}$ a její transpozice jako $\mathbf{X}^T$, projekční matici H lze psát jako

$$H = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}. \quad (24)$$

3.3.4 Ukázka v programu R

V následujícím textu v obecnosti naznačím, jak si sestavit skript, který by pomocí metod křížové validace odhadl predikční chybu. Z předchozího textu vyplývá, že minimálně v obecné rovině nepředstavuje Křížová validace nikterak obtížnou disciplínu. Metody Hold-out, K-násobná i LOOCV jsou jak snadno pochopitelné a, co se týče programování, tak i snadno i implementovatelné ve statistickém softwaru. Jak jsem už v úvodu zmínil, veškerá programovací část této diplomové práce je vypracována v programu R.

Tento software je povahou open source, mnoho uživatelů do něj přispívá a je možné si stáhnout nadstavbové verze, které dále rozšiřují funkcionalitu a možnosti použití R. Pro účely křížové validace lze například využít balíček `cvTools`. Pro účely diplomové práce jsem se rozhodl tyto balíky nevyužít a funkce jsem napsal sám.

Pro základní pochopení metod křížové validace je nejlepší aplikovat je na jednoduchou datovou strukturu. K tomuto účelu jsem si vybral datový soubor s názvem `felicie`, který obsahuje celkem $n = 20$ pozorování a tři proměnné. Vysvětlovaná proměnná `cena` udává cenu automobilu v tisících Kč. Vysvětlující proměnné jsou `stari` (proměnná `stari`) a `najeto` (počet najetých kilometrů v tisících (najeto)). Soubor je uvedený v přílohách.

Pro metodu Hold-out jsem uvedl důvod, proč nekonstruovat pouze jediný výběr, a proto jsem se věnoval přímo její modifikaci `random subsampling`, tj. opakované simulaci výběru ze základního souboru. V R jsem za tímto účelem vytvořil funkci, do které lze vkládat parametry a která vrací odhad predikční chyby. Příkazy uvnitř funkce jsou okomentovány tučně přímo ve skriptu a jsou označeny `#`.

```
Holdout = function(sim, P, dataset) { #prikaz function vytvari funkci
  data1=as.data.frame(dataset) #vlozeny datovy soubor je ulozen jako
data.frame
  colnames(data1) = c("y",paste("x", 1:(ncol(data1)-1), sep = ""))
  #prikaz colnames prejmeneje promenne na y, x1, x2, atd.
  PE = matrix(NA,ncol=1,nrow=sim) #alokace matice, do ktere budou ukladany
hodnoty odhadu predikcni chyby
  for (i in 1:sim){ #smycka for opakuje odhad predikcni chyby v rozsahu 1 az
    pocet simulaci (sim)
    sub = sample(nrow(data1), floor(nrow(data1) * P)) #nahodne vybrana
    pozorovani, funkce floor zaokrouhluje na cela cisla
    training = data1[sub, ] #treninkova podmnozina
    testing = data1[-sub, ] #validacni podmnozina
    fit = lm(y ~ .-1, data=training) # regresni model
    pred = predict(fit, testing) # vyrovnane hodnoty v testovaci podmnozine
    PE[i,] = sum((testing[, "y"] - pred)^2)/(nrow(testing)) # predikcni chyba
    jedne simulace
  }
  PEHO = mean(PE) # prumerna predikcni chyba

  return(PEHO)
}
```

Chtěl bych zmínit příkaz `as.data.frame`, který ukládá datový soubor do formátu `data.frame`. Tento formát umožňuje uchování různých datových typů ve sloupcích, což lze při regresní analýze využít například pro práci s nejen číselnými, ale i kategoriálními vysvětlujícími proměnnými. Některé funkce (např. `lm`) tento formát preferují. Příkaz `colnames(data1)` překóduje názvy sloupců na proměnné. První sloupec je zpravidla vyhrazen vysvětlované proměnné, a proto tento příkaz přepíše název prvního sloupce na `y`, ostatní sloupce označí `x` s indexem označujícím pořadí. V rámci `for` cyklu se z tréninkové podmnožiny odhadne lineární regresní model (funkce `lm`), který je následně aplikován na validační podmnožinu (funkcí `predict`). Na závěr cyklu je odhadnuta hodnota predikční chyby `PE[i,]`. Tento cyklus opakuje tolikrát, kolik je nastaveno simulací a na samém konci je vrácena predikční chyba jako průměr predikčních chyb jednotlivých simulací.

Když je funkce uložena, stačí ji jen zavolat pomocí příkazu:

```
set.seed(100)
Holdout(sim = 50,P= 0.4, dataset = felicie).
```

Parametry této funkce jsou následující. Hodnota `sim` udává počet simulací. Hodnota `P` představuje poměr, ve kterém jsou pozorování rozdělena do tréninkové a validační podmnožiny. `P = 0,4` znamená, že 40% všech pozorování je v tréninkové podmnožině. Jelikož datový soubor `felicie` obsahuje 20 pozorování, v tréninkové podmnožině je 8 pozorování, ve validační 12. Parametr `dataset` odpovídá datovému souboru, na kterém chceme odhad predikční chyby provést. Před funkcí `Holdout` je zmíněna funkce `set.seed`. Ta nastavuje generátor pseudonáhodných čísel. Je užitečné mít nastavenou nějakou konkrétní hodnotu tak, aby bylo možné daný pokus zopakovat stejným způsobem vícekrát (například při úpravách skriptu).

Lze se přesvědčit, jak se predikční chyba (průměr predikčních chyb jednotlivých simulací) liší při různých nastavení poměru `P`. Zatímco při `P = 0,9` vyšla predikční chyba 5285,85, pro `P = 0,5` byla hodnota predikční chyby 4690,6 a pro `P = 0,1` vyšla predikční chyba 198694,6.

Zápis `K`-násobné metody je v kódu o něco složitější než byla metoda `Hold-out`. Zatímco předchozí přístup vyžadoval rozdělení do dvou vzájemně nekryjících se podmnožin, u `K`-násobné je potřeba zabezpečit, aby těchto podmnožin bylo `K`. Například pro `K = 5` se sice na odhadu regresního modelu podílí vždy čtyři pětiny dat. Podstatné je zajistit, aby konkrétní

validační podmnožina neobsahovala žádné pozorování z žádné jiné validační podmnožiny. S tím souvisí i následující problém v podobě datového souboru určeného k analýze.

Už jsem zmiňoval, že je důležité, aby před samotnou analýzou byla zajištěna dostatečná stratifikace souboru. Pokud se nejedná o generovaná data, ale o nějaký datový soubor, je možné, že pozorování budou obsahovat nějakou systematickост. Při pohledu na prvních několik pozorování datového souboru felicie je patrné, že hodnoty vysvětlující proměnné stáří jsou uskupeny od nejmenších k největším. Pokud bych nejprve neprovedl stratifikaci souboru, riskoval bych ovlivnění výsledků.

Stratifikaci ale není jednoduché provést. Balíky softwaru R obsahující nástroje pro křížovou validaci využívají náhodného výběru z datového souboru, u kterého je zaručeno, že ve validačních podmnožinách nebudou opakovaná pozorování. Pro účely této diplomové práce jsem tento proces částečně obešel tím, že jsem provedl náhodný výběr bez vracení o rozsahu datového souboru. Vytvořil jsem tím nový, stejný soubor, který má ale náhodně proházená pozorování. Tento krok je zaznamenán pomocí objektů `sub` a `data1re` (re zkráceně z angl. re-sample). Díky tomu, že byl proveden náhodný výběr ze základního souboru, jsem si mohl dovolit tento nový soubor rozdělit na K validačních podmnožin jen posunutím se o délku kroku (`step`), která se přičítá k počáteční a koncové hodnotě výběrového souboru (`sta`, `konec`).

```
Kfold = function(sim, K, dataset) { # prikaz function vytvari funkci
  PEK = matrix(NA, ncol = K, nrow = sim) # alokace matice, do ktere se ukladaji
  odhady predikcni chyby
  data1 = as.data.frame(dataset) # format datoveho souboru je data.frame
  colnames(data1) = c("y", paste("x", 1:(ncol(data1)-1), sep = "")) # prikaz
  colnames prejmenuje promenne na y, x1, x2, atd.
  step = floor(nrow(data1)/K) # delka kroku, podle teto hodnoty se rozrazuji
  hodnoty do validacni podmnoziny
  for (i in 1:sim){
    sub = sample(nrow(data1), replace = FALSE)
    data1re = data1[sub, ] # resubstituce datoveho souboru
    sta=1 # pocatecni hodnota
    konec=step # konecna hodnota
    for (j in 1:K){
      testing = data1re[sta:konec, ] # treninkova podmnozina
      training = data1re[-(sta:konec), ] # validacni podmnozina
      fit = lm(y ~ .-1, data=training)
      PEK[i,j] = sum((testing[, "y"] - predict(fit, testing))^2)/(nrow(testing))
    }
  }
}
```

```

    #odhad predikcni chyby v jednom K.
    sta = sta+step # nova pocatecni hodnota validacni podmnoziny
    konec = konec+step # nova koncova hodnota validacni podmnoziny
  }
}
PEKFOLD = mean(PEK) #odhad predikcni chyby
return(PEKFOLD)
}

```

Oproti Hold-out je rozdíl v matici, do které se ukládají predikční chyby v jednotlivých krocích. V jedné simulaci je totiž pro každou validační podmnožinu proveden odhad predikční chyby. Rozměr matice, do které se ukládají tyto jednotlivé odhady, je tedy K krát počet simulací. Odhad predikční chyby každé simulace je potom průměrnou hodnotou přes K odhadů. Zbývající funkce jsou už shodné s těmi, které jsem použil u Hold-out.

Metodu K -násobnou jsem použil pro $K = 5$ a $K = 10$ a sto simulací. Tato funkce vypadá velmi podobně jako u metody Hold-out. Parametry jsou počet simulací (*sim*), o jaký typ K -násobné křížové validace se jedná (κ) a datový soubor, který je analyzován (*dataset*)

```
Kfold(sim = 100, K = 10, dataset = felicie).
```

Hodnoty predikční chyby a rozptylu jsem uvedl v tabulce 1.

K	Odhad Predikční chyby
5	4436,179
10	4424,711

Tabulka 1: Odhad predikční chyby K -násobnou křížovou validací. Zdroj: datový soubor felicie

Jako poslední jsem na data aplikoval metodu LOOCV. Ta je v zásadě shodná s metodou K -násobná, protože představuje pouze extrémní případ této metody, ve kterém $K = n$. Ve validační podmnožině se nachází pouze jediné, neopakující se pozorování, v tréninkové podmnožině jsou všechna zbylá. V tomto případě jde skript postupně od prvního k poslednímu pozorování. Díky tomu odpadá potřeba opakování simulací a nutnosti znáhodnit základní soubor.

```

loocv = function(dataset) { # prikaz function vytvari funkci
  PEj = matrix(NA, ncol = 1, nrow = nrow(data1)) #alokace matice, do ktere se
    ukladaji odhady predikcni chyby

```

```

data1=as.data.frame(dataset) #format datoveho souboru je data.frame
colnames(data1) = c("y",paste("x", 1:(ncol(data1)-1), sep = ""))
#prikaz colnames premenuje promenne na y, x1, x2, atd.
for (j in 1:nrow(data1)){
  testing = data1[j, ] #treninkova podmnozina
  training = data1[-j, ] #validacni podmnozina
  fit = lm(y ~.-1, data=training)
  PEj[j,] = sum((testing[, "y"] - predict(fit, testing))^2)
}
PELOOCV = mean(PEj)
return(PELOOCV) } #Odhad predikcni chyby

```

Odhad predikční chyby vyšel 4414,664. Funkce metody LOOCV obsahuje pouze jediný parametr. Jedná se pouze o datový soubor, na kterém je odhad predikční chyby proveden

```
loocv(dataset = felicie).
```

3.4 Bootstrap

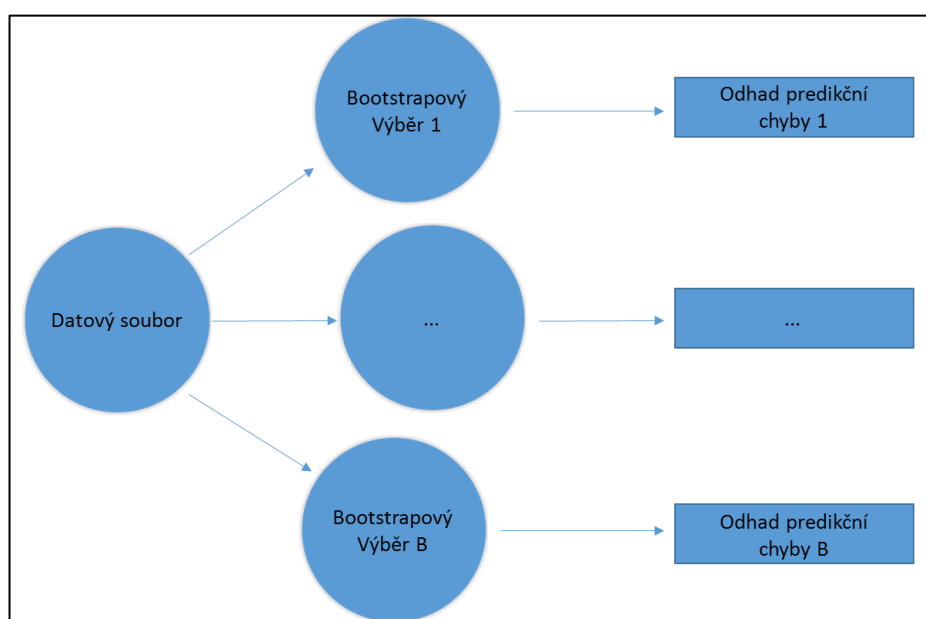
Dalším obecný přístup k odhadu predikční chyby zastupuje pojem bootstrap (někdy bootstrapping), který byl s největší pravděpodobností poprvé zmíněn v práci Bradleyho Efrona z roku 1979. Ačkoliv odhadují stejnou statistiku, bootstrap přistupuje k hodnocení predikčních vlastností modelu odlišně než křížová validace.

Pojem bootstrap se odkazuje na vyprávění o baronu Prášilovi, který se v jednom ze svých příběhů vytáhl z močálu pomocí vlastních přezek na botách (Efron, Tibshirani 1993). Myšlenka této metody spočívá v práci s vlastními prostředky, tedy daty, a to i v momentě, kdy je množství dat nedostačující.

Z technického hlediska stojí bootstrapové metody na opakovaných simulacích, často o počtu několik set až tisíc, a proto stejně jako u křížové validace zaznamenaly rozvoj paralelně s rozvojem informačních technologií. V současné době je bootstrap jednou z nejvíce používaných metod, převážně i proto, že tyto metody nelpí na předpokladech o pravděpodobnostním rozdělení základního souboru nebo rozdělení rušivých složek (Efron, Tibshirani 1993). K dalším přednostem patří především její jednoduchost a funkčnost především v případech, kdy odvození složitých matematických vzorců představuje problém.

3.4.1 Naivní bootstrapový odhad predikční chyby

Bootstrap je ve své nejzákladnější formě bývá nazýván jednoduchým nebo i naivním bootstrapovým odhadem (Mevik, Cederkvist 2005). Stejně jako křížová validace, i bootstrap jsem v této práci použil k odhadu predikční chyby. Princip naivního bootstrapu je následující. Z datového souboru o rozsahu n je učiněno B náhodných výběrů s vracením o rozsahu n (viz obrázek 3). Význam výběru s vracením je dát každému pozorování stejnou pravděpodobnost vybrání, a tak zajistit jak nezávislost jednotlivých tahů, tak jednotlivých B bootstrapových výběrů.



Obrázek 3: Schéma naivního bootstrapového odhadu. Zdroj: Efron, Tibshirani 1993

V každém bootstrapovém výběru se odhadnou koeficienty regresního modelu (například metodou nejmenších čtverců). Koeficienty modelu jsou poté zpětně aplikovány na původní datový soubor za účelem výpočtu predikční chyby jednoho bootstrapového výběru.

$$PE_b = \frac{1}{n} \sum_{i=1}^n (y_i - \widehat{y}_{b,i})^2 \quad (25)$$

Celkový naivní odhad predikční chyby se počítá jako aritmetický průměr B predikčních chyb.

$$PE_{naive} = \frac{1}{B} \sum_{b=1}^B PE_b \quad (26)$$

Tato metoda se v praxi příliš nedoporučuje. Jednotlivé bootstrapové výběry slouží jako testovací podmnožina, původní datový soubor jako validační podmnožina. Kvůli výběru s vrácením do jednotlivých bootstrapových výběrů může vzniknout překrytí pozorování původního datového souboru a bootstrapových výběrů. To může způsobit, že hodnota predikční chyby může být v průměru menší než by byla její skutečná hodnota. Takový odhad predikční chyby je příliš optimistický (Hastie a kol. 2009, str. 250).

3.4.2 Vylepšený bootstrapový odhad predikční chyby

Optimistický odhad naivního bootstrapového odhadu predikční chyby se snaží napravit tzv. vylepšený bootstrap (z angl. improved bootstrap). Ten odhaduje predikční chybu jako hodnotu RSS, vydělenou počtem pozorování n , opravenou o tzv. optimismus, který je značen $\bar{\omega}$ (27).

$$PE_{improved} = \frac{RSS}{n} + \bar{\omega} \quad (27)$$

Prvnímu členu se v terminologii bootstrapu někdy říká „zjevná chyba“ (z angl. apparent error) nebo též tzv. tréninková chyba. V kapitole 1.5 jsem hovořil o tom, že tento člen neposkytuje dobrou informaci o predikčních schopnostech regresního modelu. Z toho důvodu se provádí korekce v podobě přičtení hodnoty odhadnutého optimismu $\bar{\omega}$. Optimismus v sobě zahrnuje bootstrapový odhad a získává se podle následujícího postupu. Z datového souboru je vygenerováno B bootstrapových výběrů. V každém bootstrapovém výběru jsou odhadnuty regresní koeficienty modelu. Ty se nejdříve aplikují na původní datový celek. Tím je vlastně získán naivní bootstrapový odhad predikční chyby PE_b (28).

$$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_{b,i})^2 \quad (28)$$

Zároveň je ten samý model aplikován na bootstrapový výběr samotný. Dostáváme tím vlastně zjevnou chybu bootstrapového výběru (29).

$$\frac{1}{n} \sum_{i=1}^n (y_i^b - \widehat{y}_{b,i}^b)^2 \quad (29)$$

Průměrná hodnota rozdílu těchto dvou členů přes všech B bootstrapových výběrů je tedy optimismus $\bar{\omega}$, o který se opravuje průměrná hodnota RSS přes všechna pozorování (vzorec 30).

$$\bar{\omega} = \frac{1}{B} \sum_{i=1}^B \left(\frac{1}{n} \sum_{i=1}^n (y_i - \widehat{y}_{b,i})^2 - \frac{1}{n} \sum_{i=1}^n (y_i^b - \widehat{y}_{b,i}^b)^2 \right). \quad (30)$$

Interpretace optimismu je taková, že představuje hodnotu, o kterou zjevná chyba datového souboru podhodnocuje skutečnou predikční chybu modelu. Existují i další bootstrapové metody. Jedná se například o metodu 0,632 nebo bootstrapem vylepšená křížová validace (Mevik, Cederkvist 2005). Těm jsem v této práci pozornost nevěnoval.

3.4.3 Ukázka v programu R

Opět názorně ukáži jak si odhady predikčních chyb podle zmíněných bootstrapových metod sestavit v R. Stejně jako u křížové validace i bootstrap je v softwaru R dávno zahrnutý v podobě stažitelných balíků. Lze doporučit například `boot` nebo `bootstrap`.

Jednotlivé metody jsem se opět rozhodl prezentovat na souboru `felicie`, který jsem použil i pro potřeby křížové validace. Naivní bootstrapový odhad nepředstavuje v R žádný zásadní problém. Nadefinoval jsem $B = 100$ simulací a předem alokoval matici predikčních chyb jednotlivých bootstrapových výběrů (`PEi`). Potom následuje cyklus `for`. V něm je uskutečněn náhodný výběr s vracením o rozsahu n . Nejprve funkce `sample` náhodně vybere pozorování do objektu `sub`. Následně je pomocí `data1[sub, ]` vytvořen bootstrapový výběr odpovídající vybraným pozorováním. Na nově vzniklém datovém souboru (`boot`) je proveden odhad regresní funkce. Nakonec je odhadnuta predikční chyba jednoho bootstrapového výběru. Cyklus `for` se opakuje B -krát a na jeho konci je odhadnuta celková predikční chyba jakožto průměrná hodnota predikčních jednotlivých bootstrapových výběrů. Celý tento proces jsem uložil do samostatné funkce s parametry `B` a `dataset`

```
naiveboot(B = 10, dataset = felicie).
```

První parametr odpovídá počtu bootstrapových výběrů, druhý parametr datovému souboru. Funkce naiveboot společně s okomentovanými příkazy je uvedena níže.

```
naiveboot = function(B, dataset){ #funkce naivního bootstrapového odhadu
  data1 = as.data.frame(dataset) #data uložena do formátu data.frame
  colnames(data1) = c("y",paste("x", 1:(ncol(data1)-1), sep = ""))
  #překodování názvu sloupce na y, x1,x2 atd
  PEi = matrix(NA,ncol = B, nrow = 1) #alokace datové matice

  for (j in 1:B){
    sub = sample(nrow(data1),replace = TRUE) # vyber s vracením
    boot = data1[sub, ] #bootstrapový vyberový soubor
    fit = lm(y ~.-1, data = boot)
    PEi[,j] = sum((data1[, "y"] - predict(fit, data1))^2)/(nrow(boot))
    #predikční chyba jednoho bootstrapového vyberu
  }
  PEnaive=mean(PEi) #celková predikční chyba
  return(PEnaive)
}
```

Pro sto simulací vyšel odhad predikční chyby 3743,5.

Vylepšený bootstrapový odhad je o něco složitější. Místo alokování matice pro predikční chybu jsem alokoval matici, do které se ukládá optimismus každého bootstrapového výběru. Následoval odhad modelu s použitím původního datového souboru a výpočet zjevné chyby datového souboru. Stejně jako u naivního bootstrapového odhadu i zde na toto navazuje for cyklus, v němž jsem získal optimismus každého bootstrapového výběru Opt[, j]. Poznámám, že v tomto případě je odhad optimismu jednoho bootstrapového výběru rozepsán na dva řádky kvůli přehlednosti.

```
improvedboot = function(B, dataset){ #funkce vylepseného bootstrapového
  odhadu
  data1 = as.data.frame(dataset) #data uložena do formátu data.frame
  colnames(data1) = c("y",paste("x", 1:(ncol(data1)-1), sep = ""))
  #překodování názvu sloupce na y, x1,x2 atd
  Opt = matrix(NA,ncol=B,nrow=1) #alokace datové matice
  fit = lm(y ~ .-1, data=data1)
  avgRSS = sum(residuals(fit)^2)/nrow(data1) # „zjevná chyba“ datového
  souboru
```

```

for (j in 1:B){
  sub = sample(nrow(data1),replace=TRUE) # vyber s vracenim
  boot = data1[sub, ] #bootstrapovy vyberovy soubor
  fit2 = lm(y ~ .-1, data = boot)
  Opt[,j] = sum((data1[, "y"] - predict(fit2, data1))^2)/nrow(data1)-
    sum(residuals(fit2)^2)/nrow(boot) #optimismus jednoho bootstrapového vyberu
}
PEimproved = avgRSS + mean(Opt) #odhad predikcni chyby
return(PEimproved)
}

```

Pro sto simulací byl odhad predikční chyby 4122,83

3.5 Počet validačních podmnožin a bootstrapových výběrů

3.5.1 Počet validačních podmnožin K-násobné křížové validace

V případě K-násobné křížové validace je správnou otázkou, na kolik podmnožin soubor vlastně rozdělit. V praxi se nejvíce používají $K = 5$ nebo $K = 10$, ale je vhodné brát v potaz i jiné faktory jako je třeba rozsah souboru. Pokud by datový soubor neměl dostatek pozorování (například 60 pozorování a více), nemá smysl používat příliš velká K . Validační podmnožiny by v tom případě obsahovaly málo pozorování a hrozilo by, stejně jako u LOOCV, že odhad predikční chyby bude sice nevychýlený, ale jeho přesnost by byla snížena vysokým rozptylem. Pro menší soubory lze tedy použít K nižší než 5 nebo zkusit jinou metodu odhadu predikční chyby, například vylepšený bootstrapový odhad. Pro datové soubory s dostatečným počtem pozorování se nejčastěji používá $K = 5$ nebo $K = 10$, a to proto, že bylo empiricky dokázáno, že odhady predikční chyby jsou za těchto podmínek téměř nevychýlené (Kohavi 1995 a Breiman, Spector 1992).

3.5.2 Počet bootstrapových výběrů

Stejně jako počet validačních podmnožin u K-násobné křížové validace, i počet bootstrapových výběrů, který by zajistil spolehlivý odhad predikční chyby, je neméně důležitou otázkou při aplikaci bootstrapových metod.

Autoři v různých volně dostupných člancích využívají sice odlišných datových souborů, ale vesměs se na počtu bootstrapových výběrů shodují. Například v (Breiman, Spector 1992) je uvažován datový soubor se čtyřiceti proměnnými a 160 pozorováními. Pro tento soubor se údajně odhad predikční chyby zlepšuje jen minimálně, jakmile je počet

bootstrapových výběrů vzroste nad $B = 20$. V jiném článku autor udává jednoduché pravidlo, jak se správně rozhodnout a říká, že už $B = 25$ bootstrapových výběrů dává dobré odhady a že jen velmi zřídka se provádí nad $B = 200$ (Efron, Tibshirani 1993, str. 52).

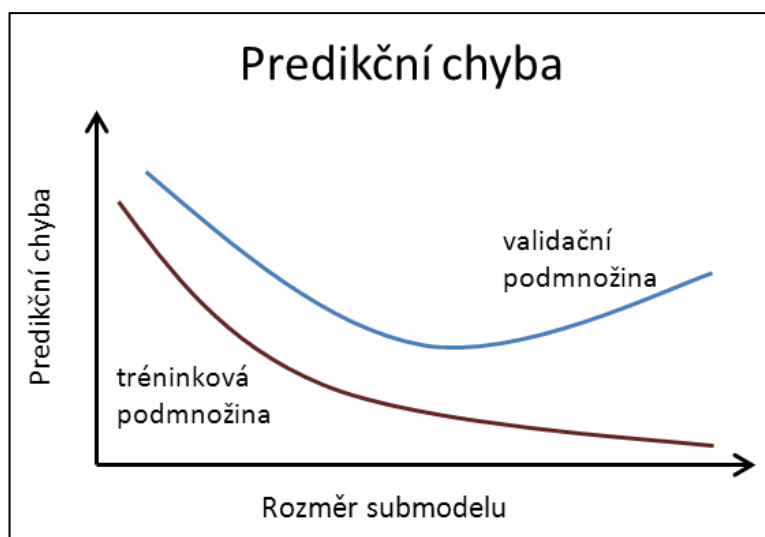
3.6 Tréninková chyba a predikční chyba

V kapitolách 1.4 a 1.5 jsem zmiňoval, že kritéria AIC, BIC, R^2 nebo R^2_{adj} neposkytují příliš dobré výsledky při hledání regresního submodelu s dobrými predikčními vlastnostmi. Důvodem je, že odhady klasických kritérií v sobě zahrnují tzv. tréninkovou chybu (Hastie a kol. 2009) – TE (z angl. training error).

$$TE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} = \frac{RSS}{n} \quad (31)$$

Tréninková chyba proto, že na rozdíl od odhadů predikční chyby metodami bootstrap nebo křížovou validací, není u datového souboru rozlišována tréninková a validační podmnožina. Na odhad koeficientů regresního modelu $\hat{y}_i$ jsou využita veškerá pozorování datového souboru. Predikční vlastnost modelu je poté ověřována na stejných datech.

Nevýhodou tréninkové chyby je, že její hodnota neroste s každou další vysvětlující proměnnou zařazenou do regresního modelu, a to i v případě, že daná vysvětlující proměnná nesouvisí s ostatními proměnnými v modelu (Hebák a kol. 2005, str. 88). To má za následek, že kritéria, která v sobě zahrnují tréninkovou chybu, mají tendenci upřednostňovat regresní modely s vyšším než optimálním počtem proměnných. Pokud by v uvažovaném hypotetickém datovém souboru byly pouze tři statisticky významné proměnné, kritéria postavená na tréninkové chybě by zvolila regresní model obsahující více než tyto tři proměnné. Této situaci se říká přeurčení regresního modelu (z angl. over-fitting). Jejich nevýhodou je snížená interpretace regresního modelu a jen málo přesné předpovědi nových hodnot vysvětlované proměnné (Hastie a kol. 2009). Situace znázorňující obecné chování tréninkové chyby je na obrázku 4.



Obrázek 4: Odhad predikční chyby. Červená čára znázorňuje průběh odhadu predikční chyby pomocí tréninkových dat pro různý počet vysvětlujících proměnných v regresním modelu. Modrá čára popisuje průběh odhadu predikční chyby za situace, ve které je odhad prováděn na validační podmnožině. Zdroj: Hastie a kol. 2009

Výše uvedené nedostatky jsou důvodem, proč je lepší používat odhady predikční chyby pomocí metod křížové validace nebo bootstrapu, které rozlišují tréninkovou a validační podmnožinu (bootstrapový výběr je svým způsobem také tréninková podmnožina). Průběh predikční chyby, která je odhadována na validační podmnožině, je znázorněn na obrázku 4 společně s tréninkovou chybou. Detailnější rozbor této predikční chyby je v následující kapitole.

3.7 Kompromis mezi rozptylem a vychýlením

V předchozí kapitole jsem hovořil o tom, že odhady predikční chyby, které využívají tréninkové chyby, dobře vystihují tréninková data. Tím je myšleno, že takto pořízené odhady se téměř shodují s napozorovanými hodnotami vysvětlované proměnné uvnitř datového souboru. O takových odhadech se říká, že jsou nevychýlené. To je sice dobrá vlastnost odhadů predikční chyby, ale není to jediná vlastnost. Situaci přiblížím obecně.

Uvažuji situaci, ve které se pro hypotetickou populaci snažím nalézt takový regresní model, se kterým bych mohl učinit předpovědi pro vysvětlovanou proměnnou. Z této populace mohu učinit libovolný počet náhodných výběrů. Na každém výběru aplikuji některou ze Stepwise metod selekce modelu a jako hodnotící kritérium použiji některé z kritérií, které v sobě zahrnuje tréninkovou chybu. V každém výběru zvolím regresní model (resp. submodel) s nejlepší hodnotou tohoto kritéria (pro AIC, BIC nejnižší, pro koeficienty determinace nejvyšší hodnota). Vyrovnané hodnoty takto získaných regresních modelů

značím $\hat{y}$. Dále uvažuji, že mám k dispozici i budoucí hodnoty vysvětlované proměnné této populace y^{bud} .

Přesnost odhadu budoucích hodnot y^{bud} regresní funkcí lze poté vyhodnotit pomocí MSE (32). Jedná se vlastně o odhad predikční chyby, ale s tím rozdílem, že predikční chybu odhaduji na jednom datovém souboru. Zde mám k dispozici libovolný počet náhodně vybraných datových souborů pocházející z jedné populace.

$$MSE(y^{bud} - \hat{y}) = D(\hat{y}) + B^2(\hat{y}) + D(\epsilon) \quad (32)$$

MSE je tedy vlastnost, kterou lze zjistit při opakovaných simulacích a popisuje, jak se odhad vyrovnaných hodnot regresní funkce zvolené podle konkrétního kritéria v průměru liší od skutečných budoucích hodnot. MSE lze rozložit na tři členy (32).

První člen je rozptyl vyrovnaných hodnot od jejich průměrné hodnoty. Vysoké hodnoty rozptylu nejsou žádoucí, protože to znamená, že malá odchylka v datech má velký vliv na vyrovnané hodnoty $\hat{y}$ (James a kol. 2013, str. 34). Druhý člen je čtverec vychýlení odhadu. Vychýlení představuje odchylku střední hodnoty odhadu od jeho skutečné hodnoty.

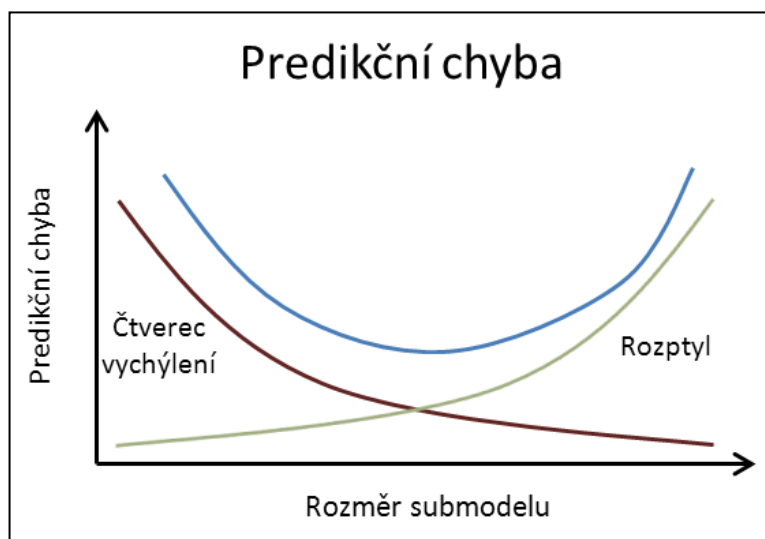
$$B(\hat{y}) = E(\hat{y}) - y^{bud} \quad (33)$$

Obecně platí, že odhad je považován za nevychýlený, pokud jsou tyto hodnoty v rovnosti. Třetí člen je tzv. nesnižitelná chyba (z angl. irreducible error). Nesnižitelná chyba odpovídá rozptylu náhodné složky a je součástí variability odhadů předpovědí jakožto forma penalizace za neznalost budoucnosti.

MSE v případě metod, které k odhadu predikční chyby využívají tréninkovou chybu, nesnižitelnou chybu neobsahuje (Hebák a kol. 2005, str. 54). To je důvod, proč je odhad predikční chyby těmito metodami téměř vždy nižší, a tedy optimističtější, než v případě metod, které vyhodnocují predikční chybu na validační podmnožině. To je jeden z důvodů, proč se nedoporučuje takové metody používat pro odhad predikční chyby.

Jestliže je poslední člen na pravé straně výrazu (32) označován jako nesnižitelná chyba, ostatní členy lze tedy nazývat snižitelnou chybou (z angl. reducible error). A právě o snížení těchto hodnot usilují různé metody odhadu predikční chyby.

Lze ukázat, že rozptyl vyrovnaných hodnot je v průměru nižší u modelů s menším počtem vysvětlujících proměnných v modelu (James a kol. 2013, str. 36). Naopak vychýlení vyrovnaných hodnot od skutečných budoucích hodnot vysvětlované proměnné se snižuje s rostoucím počtem vysvětlujících proměnných v modelu (James a kol. 2013, str. 36). Rozptyl a vychýlení jsou vzájemně provázané a snížení jedné charakteristiky vede ke zvýšení druhé. Situaci zachycuje obrázek 5.



Obrázek 5: Složení predikční chyby z hlediska rozptylu a vychýlení.
Zdroj: Hastie a kol. 2009

To odpovídá na otázku, proč se přeurčené modely nehodí k odhadu predikční chyby. Jejich výhodou je sice, že se mnohem lépe přizpůsobují datům bez ohledu na to, jestli se jedná o data ze současného datového souboru nebo z budoucího. To vše ale za cenu vysokého rozptylu vyrovnaných hodnot, který je u přeurčených modelů přítomný a který snižuje přesnost těchto odhadů (Hastie a kol. 2009). Metody křížové validace a bootstrapu berou v potaz jak vychýlení, tak rozptyl, a proto jsou obecně považovány jako lepší nástroje hodnocení predikčních schopností regresního modelu.

Jak už bylo řečeno, rozptyl a vychýlení odhadu jsou navzájem provázané a bohužel nelze nalézt takový submodel, v němž by klesly obě charakteristiky najednou. Rozptyl lze snižovat například rozšířením datového souboru o nová pozorování. To ale znamená zvýšení nákladů a další vynaložený čas na získání dalších pozorování. V jiném případě už nemusí být možné další data získat. Vychýlení je ovlivněno částečně použitými metodami, ale také (stejně jako rozptyl) závisí na tom, jestli je regresní model přeurčen nebo podurčen.

Nezbývá, než dosáhnout jistého kompromisu mezi těmito hodnotami. Tomu se v anglické literatuře říká *Bias-Variance Trade-Off* a při hledání optimálního regresního submodelu je tento kompromis hlavním cílem. Pointa tohoto hledání spočívá v tom, že nevychýlený odhad nemusí nutně být ten nejlepší, protože může mít vysoký rozptyl. Odhad s nejnižší predikční chybou připouští sice přítomnost jak zkreslení, tak rozptylu, ale v takové kombinaci, která poskytuje nejlepší předpovědní vlastnosti. To je důvod, proč se při hledání optimální dimenze modelu spoléhá na minimum predikční chyby.

4 Aplikace na datech

4.1 Úvod

V teoretické části jsem hovořil o klasických kritériích, jejich vlastnostech a důvodech, proč je lepší tato kritéria nepoužívat při výběru regresního submodelu určeného k předpovědím. Jako alternativu jsem postavil kritéria, která vyhodnocují predikční schopnosti regresních modelů pomocí opakovaných simulací. Fungování těchto metod bylo vysvětleno společně s ukázkou, jak tyto funkce napsat v software R. Metody jsem ilustroval na datovém souboru felicie, který má pouze dvacet pozorování. Na tomto datovém souboru nebyla provedena selekce regresního submodelu.

Datové soubory jako felicie se v praxi příliš často nevyskytují. Běžné je pracovat s datovými soubory čítající desítky proměnných a mající řádově stovky pozorování. Z datových souborů je pak potřeba vybrat optimální regresní submodel, který by poskytl nejpřesnější předpovědi hodnot vysvětlované proměnné.

Tento oddíl je věnován právě hledání optimálního regresního submodelu. Za tímto účelem jsem vygeneroval datový soubor, na který jsem aplikoval jednu z metod selekce modelu Stepwise s využitím kritérií, o kterých jsem hovořil v předchozích oddílech. Z klasických kritérií jsem využil koeficient determinace, korigovaný koeficient determinace, AIC a BIC. Z kritérií, která využívají opakovaných simulací, jsem využil vylepšený bootstrapový odhad predikční chyby, 5-násobnou a 10-násobnou křížovou validaci a LOOCV.

Celá praktická část byla (stejně jako simulační metody v předchozích oddílech) naprogramována v softwaru R s pomocí vlastních funkcí.

4.2 Vymezení pojmů

Maticí pozorování $\mathbf{X}$ je myšlena matice, ve které jsou uložena pozorování vysvětlujících proměnných. Matice je rozměru $n \times K$. To znamená, že má tolik řádků, kolik je pozorování, a tolik sloupců, kolik je vysvětlujících proměnných (Hebák a kol. 2005). Vysvětlujících proměnné jsou značeny X_k , kde index k nabývá hodnot od 1 do K .

Kovarianční maticí $\mathbf{\Sigma}$ je myšlena symetrická, čtvercová matice o rozměru $K \times K$, která má na diagonále rozptyly vysvětlujících proměnných X_1 až X_K , mimo diagonálu se nachází kovariance těchto proměnných. Kovariancí je myšlena intenzita vztahu vysvětlujících proměnných (Bílková a kol, 2009). Je definována vzorcem (34).

$$C(X_k, X_{k'}) = E[(X_k - E(X_k))(X_{k'} - E(X_{k'}))], k \neq k' \quad (34)$$

4.3 Datový soubor

Při generování datového souboru jsem vycházel z teoretických předpokladů publikovaného článku Leo Breimana a Philipa Spectora z časopisu International Statistical Review z roku 1992. Tento článek jsem využil z toho důvodu, že obsahoval grafické výstupy, podle kterých jsem se orientoval při programování funkcí pro odhad predikční chyby. Článek sám o sobě žádný kód neobsahuje. Datový soubor, na který byla selekce regresního submodelu aplikována, byl vytvořen podle následujícího schématu.

Vycházel jsem z klasického lineárního regresního modelu definovaného podle

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \quad (35)$$

Vzorec (35) obsahuje vektor napozorovaných hodnot vysvětlované proměnné $\mathbf{y}$. Ty jsou určeny součinem matice pozorování $\mathbf{X}$ a vektoru skutečných regresních koeficientů $\boldsymbol{\beta}$, ke kterému je přičten vektor hodnot náhodné složky $\boldsymbol{\epsilon}$ (Hebák a kol. 2005, str. 39).

Matice pozorování $\mathbf{X}$ byla vygenerována z vícerozměrného normálního rozdělení

$$\mathbf{X} \sim N_K(\boldsymbol{\mu}, \mathbf{\Sigma}). \quad (36)$$

První parametr je vektor středních hodnot $\boldsymbol{\mu}$, který je pro účely generování matice pozorování nulový. Druhým parametrem je kovarianční matice $\boldsymbol{\Sigma}$. Hodnoty kovarianční matice (včetně rozptylů na diagonále) byly definovány podle vzorce (37)

$$C(X_k, X_{k'}) = \rho^{|k-k'|} \quad (37)$$

Koeficient ρ v tomto kontextu nepředstavuje korelační koeficient, ale libovolně volitelnou konstantu. Její hodnota byla nastavena na $\rho = 0,7$. Pro lepší pochopení uvedu několik příkladů. Hodnota v prvním řádku a prvním sloupci je jedna, protože exponent je v tomto případě nula. Hodnota v prvním řádku a druhém sloupci je 0,7 atd. Počet vysvětlujících proměnných (K) byl stanoven na 40.

Důležitou roli hraje skutečný vektor regresních koeficientů β . Tento vektor zpravidla nebývá k dispozici a jeho koeficienty se odhadují pomocí například metody nejmenších čtverců. Jelikož je datový soubor generován, jeho hodnoty jsou předem známy. Hodnota jednotlivých regresních koeficientů je definována podle (38).

$$\begin{aligned}\beta_{10+j} &= (h-j)^2, |j| < h \\ \beta_{20+j} &= (h-j)^2, |j| < h \\ \beta_{30+j} &= (h-j)^2, |j| < h\end{aligned}\tag{38}$$

[illegible]

Před aplikací vektoru regresních koeficientů na matici pozorování $\mathbf{X}$ je třeba tento vektor vynásobit určitou konstantou tak, aby hodnota teoretického koeficientu determinace byla rovna 0,75 (39). Tuto konstantu je třeba spočítat pro $h = 1$ a $h = 2$ zvlášť. Ve vzorci pro h

= 2 stačí nahradit vektor $\beta_{h=1}$ novým vektorem $\beta_{h=2}$. Odvození tohoto vzorce lze nalézt v příloze.

$$q = \sqrt{\frac{3}{\beta_h^t \Sigma \beta_h}} \quad (39)$$

Aplikování konstanty q na vektory regresních koeficientů je $\beta_{h=1}$ a $\beta_{h=2}$ jsou získány nové vektory regresních koeficientů $\beta_{h=1}^*$ a $\beta_{h=2}^*$.

$$\begin{aligned} \beta_{h=1}^* &= q\beta_{h=1} \\ \beta_{h=2}^* &= q\beta_{h=2} \end{aligned} \quad (40)$$

Takto vzniklé vektory regresních koeficientů jsou aplikovány na matici pozorování X za účelem získání vektoru hodnot vysvětlované proměnné (vzorec (35)). Na závěr je přičten vektor náhodné složky ϵ . Tím je tvorba datového souboru u konce.

4.4 Simulační schéma

V rámci hledání kritéria, které by spolehlivě našlo regresní submodel s nejlepšími predikčními vlastnostmi, byl pro každé z nich datový soubor (definovaný v kapitole 3.3) vygenerován celkem 250krát. V každé simulaci byla aplikována metoda Stepwise selekce Backward s využitím F-testu jako kritéria, podle kterého jsou eliminovány proměnné. Na každý takto získaný regresní submodel jsem postupně spočetl koeficient determinace, korigovaný koeficient determinace, AIC a BIC. Z alternativních kritérií pak vylepšený bootstrapový odhad predikční chyby, 5-násobnou a 10-násobnou křížovou validací a LOOCV.

Detailní postup jedné z 250 simulací je následující. V prvním kroku je provedena regrese s plným modelem, tj. s modelem, který obsahuje všechny vysvětlující proměnné v datovém souboru. Podotýkám, že se jedná o regresi bez konstanty. Následuje výpočet předem určeného kritéria (např. AIC, odhadu predikční chyby vylepšeným bootstrapovým odhadem nebo jednou z metod křížové validace). Poté by měla následovat eliminace proměnné, jejíž vyřazení nejvíce zlepší hodnotu sledovaného kritéria.

Nevýhodou metod Stepwise selekce, jež jsou součástí statistických software, je, že pokud je nalezeno minimum zvoleného kritéria, metoda už dále nepokračuje. Pokud by tedy bylo nalezeno minimum určitého kritéria v modelu, který obsahuje tři vysvětlující proměnné, nebylo by možné získat hodnoty kritéria pro model se dvěma nebo pouze jednou vysvětlující proměnnou. Z toho důvodu byla selekce proměnných modifikována tak, že je vždy vyloučena vysvětlující proměnná s nejnižší statistickou významností měřenou na základě F-testu, tj. s nejvyšší p-hodnotou. Klasická a simulační kritéria se tedy na samotném procesu selekce nepodílí. Nicméně se podle optimálních hodnot těchto kritérií orientují při hledání vhodného regresního submodelu.

Proces selekce začne s plným modelem a skončí v momentě, kdy model neobsahuje žádnou vysvětlující proměnnou. Z toho vyplývá, že v jedné simulaci je pro metody bootstrapu a křížové validace vždy 41 odhadů predikční chyby (model, ve kterém není obsažena žádná vysvětlující proměnná, je také brán v potaz). Pro klasická kritéria je odhadů pouze 40, protože pro regresní model s žádnou vysvětlující proměnnou nejsou definovány standardní odhady.

4.5 Sledované charakteristiky

Předtím, než začnu hovořit o sledovaných charakteristikách u zvolených kritérií, je nutné vysvětlit některé pojmy.

V každé simulaci je v každém kroku selekce regresního submodelu zjištěna hodnota zvoleného kritéria (AIC, BIC, odhad predikční chyby atd.). Každé hodnotě pak odpovídá počet proměnných, které daný regresní submodel obsahuje. Sleduji počet vysvětlujících proměnných, které dané kritérium pro regresní model zvolilo jako optimální, tj. kolik vysvětlujících proměnných je například v modelu s nejnižší hodnotou AIC či v modelu s maximální hodnotou koeficientu determinace.

Z toho plyne, že pokud Akaikeho informační kritérium nalezne v procesu selekce minimum pro model se dvěma vysvětlujícími proměnnými, lze prohlásit, že z pohledu tohoto kritéria se jedná o model s nejlepšími predikčními vlastnostmi. Stejně to platí i pro ostatní kritéria. Odhadů optimálního počtu vysvětlujících proměnných je stejně pro každé kritérium stejně jako počet simulací, tj. 250.

Díky tomu, že je datový soubor simulovaný, je možné nalézt i tzv. skutečný optimální počet vysvětlujících proměnných. Ten v každé simulaci odpovídá počtu proměnných, při

kterém je minimalizována skutečná predikční chyba. Skutečná predikční chyba PE_{true} je pro $h = 1$ definována vzorcem (41). Název skutečná je volným překladem z anglického true, který je používán v (Breiman, Spector 1992).

$$PE_{true} = (\hat{\beta} - \beta_{h=1}^*)\Sigma(\hat{\beta} - \beta_{h=1}^*)^T + \sigma^2 \quad (41)$$

Pomocí $\hat{\beta}$ je vzorci (41) značen vektor odhadnutých regresních koeficientů v každém kroku Backward eliminace proměnných. $\beta_{h=1}^*$ je vektor skutečných regresních koeficientů určených vzorcem (40). Chybějící hodnoty ve vektoru $\hat{\beta}$ jsou nahrazeny nulami. Skutečnou predikční chybu regresního modelu lze chápat jako skutečnou predikční schopnost daného regresního submodelu. Tuto schopnost se snažím nalézt pomocí odhadů křížové validace a bootstrapu. Minima nabývá pro regresní submodel, který v dané simulaci nejlépe předpovídá hodnoty vysvětlované proměnné.

Právě skutečný optimální počet vysvětlujících proměnných se snažím nalézt aplikací různých kritérií na metodu selekce Backward. Jelikož náhodná složka způsobuje, že je generovaný datový soubor při každé z 250 simulací trochu odlišný, jako skutečný optimální počet vysvětlujících proměnných se považuje průměrná hodnota počtu vysvětlujících proměnných odpovídající minimu skutečné predikční chyby. Tento průměr je pro každé kritérium počítán přes všechny simulace a značím ho Dim_{opt} .

Nyní zpět k charakteristikám, o které jsem se v rámci simulací zajímal. První sledovanou charakteristikou je střední čtvercová odchylka počtu vysvětlujících proměnných odpovídajících optimálnímu regresnímu submodelu každé simulace (podle daného kritéria) od skutečného optimálního počtu vysvětlujících proměnných. Pokud bych označil počet simulací n_{sim} , počet vysvětlujících proměnných odpovídající optimu určitého kritéria $\widehat{Dim}_t$ a skutečný optimální počet vysvětlujících proměnných jako Dim_{opt} , hledaná střední čtvercová odchylka od skutečného optimálního počtu vysvětlujících proměnných MSE_{Dim} lze vyjádřit vzorcem (42). Situaci ilustruji pro kritérium AIC.

$$MSE_{Dim}^{AIC} = \frac{\sum_{i=1}^{n_{sim}} (\widehat{Dim}_t - Dim_{opt})^2}{n_{sim}} \quad (42)$$

Tuto charakteristiku lze nahradit její odmocninou z důvodu lepší interpretace. Ta kritéria, která dosáhnou nižší hodnoty MSE_{Dim} , jsou hodnocena jako vhodnější při hledání skutečného optimálního počtu vysvětlujících proměnných (pro generovaná data). Tuto charakteristiku jsem zjišťoval pro všechna kritéria.

Další charakteristiky jsem sledoval už pouze pro metody bootstrapu a křížové validace. Pro tyto metody mne zajímala průměrná hodnota minima odhadu predikční chyby $\overline{PE}_{min}$. Situaci pro metodu LOOCV ilustruje vzorec (43). To reprezentuje nalézt v každé z 250 simulací (pro dané kritérium bootstrapu nebo křížové validace) minimum odhadu predikční chyby a spočítat jejich průměrnou hodnotu.

$$\overline{PE}_{min}^{LOOCV} = \frac{\sum_{i=1}^{n_{sim}} PE_{min,i}}{n_{sim}} \quad (43)$$

Čím nižší je tato hodnota $\overline{PE}_{min}^{LOOCV}$, tím přesnější jsou předpovědi modelu vybraného na základě zvolené metody.

Posledním sledovaným kritériem (opět pouze pro odhady křížové validace a bootstrapu) byla RMSE odhadu predikční chyby. MSE odhadu je obecně definována jako střední čtvercová odchylka odhadnutých parametrů od skutečné hodnoty parametru. RMSE je definována jako odmocnina z MSE (RMSE – z angl. root mean squared error). Jelikož je v každé simulaci skutečná hodnota predikční chyby PE_{true} mírně odlišná, tak je místo její skutečné hodnoty použita její průměrná hodnota (přes všechny simulace) v každém kroku metody selekce submodelu Backward. To znamená, že pro každou metodu je vždy celkem 41 průměrných hodnot průměrné skutečné predikční chyby (pro plný model, pro model bez jedné proměnné atd.) Můžeme ji označit $\overline{PE}_{true}$.

$$\overline{PE}_{true} = \frac{\sum_{i=1}^{n_{sim}} PE_{true,i}}{n_{sim}} \quad (44)$$

Pokud by byl odhad predikční chyby v daném kroku Backward selekce submodelu označen $\widehat{PE}_l$, tak by bylo možné RMSE odhadu predikční chyby vyjádřit vzorcem (45).

$$RMSE(PE) = \sqrt{\frac{\sum_{i=1}^{n_{sim}} (\widehat{PE}_i - \overline{PE}_{true})^2}{n_{sim}}} \quad (45)$$

Pro každé kritérium bootstrapu a křížové validace je obdrženo opět 41 hodnot. Každá z nich odpovídá $RMSE(PE)$ v každém kroku Backward selekce. Význam RMSE odhadu predikční chyby spočívá v tom, že umožní posoudit vlastnosti odhadu predikční chyby pomocí zvolené metody křížové validace a bootstrapu. Jedná se rozptyl a především o vychýlení odhadu predikční chyby.

4.6 Simulace v programu R

Oproti kapitolám, které jsem věnoval ilustrování principu metod v software R, bylo naprogramování metod ztíženo o zahrnutí metody selekce regresního submodelu Backward. A proto jsem došel k závěru, že je lepší ho rozdělit do dvou částí. K práci jsou přiloženy dva soubory s kódem. První soubor obsahuje kód funkcí, které generují data, druhý soubor obsahuje funkce samotné.

Při psaní kódu jsem se orientoval podle grafických výstupů v článku (Breiman, Spector 1992). Abych zajistil přesnost výsledků, bylo nutné odhad predikční chyby přizpůsobit tomu, který je v článku používán.

První odlišnost spočívala v odhadování rozdílné charakteristiky. Zatímco se u metod křížové validace a bootstrapu zabývám odhadem predikční chyby, v článku je snaha odhadnout vlastnost, která je nazývána chybou modelu (z angl. model error) a značena ME. Rozdíl mezi odhadem chyby modelu a odhadem predikční chyby spočívá v tom, že zatímco odhad predikční chyby obsahuje nesnižitelnou chybu (viz (32) jakožto penalizaci za neznalost budoucnosti, odhad chyby modelu je o tuto hodnotu snížen. Obecně lze skutečnou chybu modelu vyjádřit pomocí (46).

$$ME = PE - D(\epsilon) \quad (46)$$

V praxi se pak nejedná o přesnou hodnotu $D(\epsilon)$, ale o její odhad, který je definován vzorcem (47).

$$est(D(\epsilon)) = \frac{RSS_M}{(N - M)} \quad (47)$$

Důvod, proč je v článku využíván odhad chyby modelu a ne predikční chyb, spočívá v tom, že nesnižitelná chyba je přítomna v každém odhadu predikční chyby a nelze ji snížit sebelepším regresním modelem. Jako taková nic nevypovídá o předpovědních vlastnostech daného regresního modelu a lze ji tedy vypustit.

Druhá odlišnost spočívala v samotných odhadech predikční chyby (resp. chyby modelu). Situaci ilustruji na odhadu predikční chyby K-násobnou křížovou validací. Oproti odhadu, který využívám já a který je definovaný vzorcem (20), je odhad predikční chyby K-násobnou křížovou validací sumou, nikoliv průměrem přes počet validačních podmnožin K (48).

$$PE_{K-Fold} = \sum_{k=1}^K (PE_{-k}) \quad (48)$$

Stejná situace se opakuje i pro odhady predikční chyby jinými metodami křížové validace nebo bootstrapu. Z toho vyplývá, že odhady skutečné predikční chyby (nebo chyby modelu) v článku jsou oproti těm, které používám, N -krát vyšší.

Jak jsem řekl v úvodu kapitoly, odhady jsem konstruoval podle modifikací v uvedených v článku. V momentě, kdy byla zajištěna shoda grafických výstupů mých funkcí s grafickými výstupy v článku, vložil jsem do kódu i odhady predikční chyby tak, jak byly vymezeny v teoretické části této práce. Součástí mnou napsaných Rkovských funkcí odhadu predikční chyby je možnost rozlišovat mezi těmito různými odhady nastavením určitého parametru.

Z důvodu shody nejsou odhady způsobem uvedeným v článku (Breiman, Spector 1992) prezentovány. Prezentovány jsou odhady, které jsem definoval v teoretické části této práce.

4.6.1 Parametry funkcí

Za účelem selekce nejlepšího regresního submodelu z hlediska předpovědi jsem sestrojil dvě funkce. První využívá jako kritérium odhad predikční chyby metodou vylepšeného bootstrapového odhadu:

```
bootstr(sim, N, B, mu, G, mod, sigmaEps2, Breiman).
```

Druhá využívá jako kritérium odhad predikční chyby metodami křížové validace (K-násobná, LOOCV):

```
crossval(sim, N, K, mu, G, mod, sigmaEps2, Breiman).
```

Tato kapitola je věnována parametrům těchto funkcí. Následující kapitola je věnována tomu, jak tyto funkce fungují.

Parametr `sim` představuje počet simulací. Simulací jsem zvolil 250. Parametr `N` odpovídá počtu pozorování, které má vygenerovaný datový soubor (datový soubor je generován uvnitř funkcí `bootstr` a `crossval` podle schématu kapitoly 4.3). Parametr `B` funkce `bootstr` odráží počet bootstrapových výběrů, parametr `K` funkce `crossval` říká, koliknásobná křížová validace je provedena. V rámci simulací jsem spouštěl tuto funkci pro parametry $K = 5, 10$ a N . Pokud se $K = N$, spustí se odhad predikční chyby pomocí LOOCV. To je umožněno díky tomu, že metoda křížové validace LOOCV pouze extrémním případem K-násobné křížové validace pro případ, kdy je validačních podmnožin stejně jako pozorování.

Parametr `mu` odpovídá vektoru středních hodnot vícerozměrného normálního rozdělení, ze kterého pochází matice pozorování $\mathbf{X}$. Jedná se o vektor 40 nul a lze ho vytvořit takto: `mu = rep(0, 40)`.

Parametr `G` zastupuje funkci, která generuje kovarianční matici vysvětlujících proměnných vícerozměrného normálního rozdělení, ze kterého pochází matice pozorování $\mathbf{X}$. Tuto matici lze vytvořit funkcí `G = getCovMatrix()`. Její funkce vypadá následovně:

```
getCovMatrix = function() { #funkce kovariancni matice vysvetlujicich  
                           promennych  
  G = matrix(NA, ncol=40, nrow=40) #alokace matice, do ktere jsou hodnoty  
                                     ukladany  
  for (i in 1:40){ #prvni for cyklus  
    for (j in 1:40){ #druhy for cyklus
```

```

        G[i,j] = 0.7^(abs(i-j)) # hodnoty uvnitř kovarianční matice
    } }
    return(G)}

```

Parametr `mod` odpovídá funkci `adjustCoef`, která vrací modifikovaný vektor skutečných regresních koeficientů tak, aby bylo dosaženo hodnoty teoretického koeficientu determinace 0,75 ($R^2 = 0.75$). Tuto funkci lze zavolat takto:

```
mod = adjustCoef(v = v, G = G, R2 = R2, sigmaEps2 = sigmaEps2).
```

Funkce `adjustCoef` je uvedena pod textem. Parametry této funkce jsou kovarianční matice `G`, teoretický koeficient determinace `R2`, rozptyl náhodné složky (je i parametrem funkcí `crossval` a `bootstr`) `sigmaEps2`. Jelikož náhodná složka pochází v těchto simulacích z normálního rozdělení se střední hodnotou nula a rozptylem jedna, je její hodnota nastavena na 1 (`sigmaEps2 = 1`).

```

adjustCoef = function(v, G, R2, sigmaEps2) {
    q = sqrt((sigmaEps2/(1 - R2) - sigmaEps2)/(t(v) %*% G %*% v))
    return(q * v)
}

```

Posledním parametrem funkce `adjustCoef` je `v`. Tento parametr odpovídá vektoru regresních koeficientů před jejich modifikací na nové, pro něž je teoretický koeficient roven 0,75 (38). Tento vektor lze vytvořit funkcí `v = getCoefVector()` a jeho funkce vypadá takto:

```

getCoefVector = function() { #funkce vrati vektor regresnich koeficientu

    v = c(rep(c(rep(0, 9),1), 3), rep(0, 10)) #vektor modelu h=1
    #v = c(rep(0,8),9,4,rep(c(1,rep(0,7),9,4),2),1,rep(0,9)) vektor modelu h=2
    return(v)}

```

Posledním parametrem funkcí `crossval` a `bootstr` je `Breiman`. Jedná se o logickou podmínku, díky které funkce provedou odhad predikční chyby způsobem uvedeným v článku (`Breiman=TRUE`) nebo podle vzorců v této práci (`Breiman=FALSE`). Její účel měl význam při výstavbě kódu. Veškeré odhady jsou provedeny pro nastavení `Breiman=FALSE`.

4.6.2 Funkce odhadu predikční chyby

Tato kapitola je věnována vysvětlení fungování funkcí `crossval` a `bootstr`. Funkce `crossval` provádí opakované simulace, ve kterých je provedena metody selekce regresního

submodelu Backward. Jako kritérium hodnocení predikční schopností regresních submodelů je zde odhad predikční chyby metodou křížové validace. Funkce v sobě zahrnuje tři smyčky for. V první smyčce je vygenerován datový soubor. V druhé smyčce je datový soubor rozdělen na validační podmnožinu a tréninkovou podmnožinu. Pro $K = 5$ je ve validační podmnožině vždy pětina dat, pro $K = 10$ jedna desetina atd. V poslední smyčce je provedena samotná selekce, která vyřazuje vysvětlující proměnné s nejvyšší p-hodnotou. V každém kroku selekce je proveden odhad predikční chyby a jeho hodnota uložena (vzorce (20) a (22)). Celý kód je následuje (včetně popisků).

```
crossval = function(sim, N, K, mu, G, mod, sigmaEps2, Breiman = FALSE) {
  # sim ... pocet simulaci
  # N ... pocet pozorovani v datovem souboru
  # K ... typ krizove validace K-nasobna
  # mu ... vektor strednich hodnot
  # mod ...modifikovany vektor regresnich koeficientu
  # sigmaEps2 ... rozptyl nahodne slozky
  # Breiman ... typ odhadu predikcni chyby (TRUE nebo FALSE)

  n = length(mod)
  PE = matrix(NA, ncol = K, nrow = n + 1)
  err = matrix(NA, ncol = sim, nrow = n + 1)
  trueErr = err

  for (j in 1:sim) { # prvni smycka
    A = mvrnorm(n = N, mu, G, tol = 0, empirical = FALSE) #matice
    vysvetlujicich promennych
    Noise = rnorm(N) # vector nahodne slozky
    yn = (A %*% mod) + noise # vektor hodnot vysvetlovane promenne
    dataset = data.frame(yn,A) #datovy soubor
    fit = lm(yn ~.-1, data=dataset) # regrese plneho modelu
    RSS = sum((resid(fit))^2) # rezidualni soucet ctvercu

    for (m in 1:K) {# druha smycka
      testing = data.frame(dataset[ceiling(nrow(dataset))*
        (m-1)/K+1):floor(nrow(dataset)*(m)/K),])
        #validacni podmnozina
      training = data.frame(dataset[-(ceiling(nrow(dataset))*
        (m-1)/K+1):floor(nrow(dataset)*(m)/K)),])
        # treninkova podmnozina
      for(i in 1:(n + 1)) { # treti smycka - Backward selekce
        fit = lm(yn ~.-1, data=training)
        PE[i,m] = sum((predict(fit, testing, se.fit = TRUE)$fit-
```

```

testing[, "yn"])^2)/nrow(testing) # Odhad predikcni chyby
U = summary(fit)$coef
Training = training[,!(colnames(training) %in%
rownames(U)[which.max(U[,4])]), drop = FALSE]
#novy treninkovy soubor po provedeni selekce
} }
if (Breiman) {# Logicka podminka
err[,j] = N*rowMeans(PE) - N*RSS/(N - n) #Breiman = TRUE: odhad chyby
modelu podle clanku
} else {
err[,j] = rowMeans(PE) #Breiman = FALSE: odhad predikcni chyby podle
teorie v DP
}
trueErr[,j] = getTrueErr(dataset = dataset, G = G, mod = mod,
sigmaEps2 = sigmaEps2, Breiman = FALSE)
#skutecna predikcni chyba
}
return(list(err = err, trueErr = trueErr))}

```

Funkce `bootstr` také provádí opakované simulace, ve kterých je provedena metoda selekce regresního submodelu Backward. Jako kritérium hodnocení predikční schopností regresních submodelů je vylepšený bootstrapový odhad predikční chyby podle vzorce (27). Funkce v sobě zahrnuje čtyři smyčky `for`. V první smyčce je vygenerován datový soubor. V druhé smyčce jsou získány hodnoty RSS v každém kroku Backward selekce. Třetí smyčka provádí bootstrapové výběry. Ve čtvrté smyčce je provedena Backward selekce uvnitř každého bootstrapového výběru a v každém kroku této selekce je proveden výpočet optimismu (E_b). Celý kód je následuje (uvádím jen popisky, které se odlišují od kódu křížové validace).

```

bootstr = function(sim, N, B, mu, G, mod, sigmaEps2, Breiman = FALSE) {
# sim ... pocet simulaci
# N ... rozsah vyberu
# B ... pocet bootstrapovych vyberu
# mu ... vektor strednich hodnot vysvetlujicich promennych
# G ... kovariancni matice vysvetlujicich promennych
# mod ... modifikovany vektor regresnich koeficientu
# sigmaEps2 ... rozptyl nahodne slozky
# Breiman ... typ odhadu predikcni chyby (TRUE nebo FALSE)

n = length(mod)
RSSJ = matrix(NA, ncol=1, nrow = n + 1)
EbJ = matrix(NA, ncol=K, nrow = n + 1)

```

```

err <- matrix(NA, ncol=sim, nrow= n + 1)
trueErr = err

for (j in 1:sim) { #první smyčka
  A = mvrnorm(n = N, mu, G, tol = 0, empirical = FALSE)
  noise = rnorm(N)
  yn = (A %*% mod) + noise
  dataset = data.frame(yn,A)
  fit0 = lm(yn ~.-1, data=dataset)
  RSS = sum((resid(fit0))^2)
  dataset2=dataset

  for (o in 1:(n + 1)) { #druhá smyčka
    fit1 = lm(yn ~.-1, data=dataset2)
    RSSJ[o,] = sum((resid(fit1))^2) #vypocet RSS v kazdem kroku selekce
    U = summary(fit1)$coef
    dataset2 = dataset2[,!(colnames(dataset2) %in%
      rownames(U)[which.max(U[,4])]),drop=FALSE]
  }

  for (m in 1:B) { #třetí smyčka
    dataset1=dataset
    sub = sample(nrow(dataset1),replace=TRUE)
    repdataset = dataset1[sub,drop=F] # bootstrapovy vyberovy soubor

    for(i in 1:(n + 1)) { #čtvrtá smyčka
      fit2 = lm(yn ~.-1, data=repdataset)
      EbJ[i,m] = sum((dataset1[, "yn"]-predict(fit2,dataset1))^2)-
        sum((fit2$residuals)^2) #optimismus bootstrapoveho vyberu
      U = summary(fit2)$coef
      dataset1 = dataset1[,!(colnames(dataset1) %in%
        rownames(U)[which.max(U[,4])]),drop=FALSE]
      repdataset = repdataset[,!(colnames(repdataset) %in%
        rownames(U)[which.max(U[,4])]),drop=FALSE]
      #nový datový soubor po provedení selekce
    }

    if (Breiman) { # Logická podmínka
      err[,j] = RSSJ + rowMeans(EbJ) - N * RSS/(N - n) #Breiman = TRUE: odhad chyby modelu podle članku
    } else { # náš přístup
      err[,j] = (RSSJ + rowMeans(EbJ))/N #Breiman = FALSE: odhad predikční chyby podle teorie v DP
    }

    trueErr[,j] = getTrueErr(dataset = dataset, G = G, mod = mod,
      sigmaEps2 = sigmaEps2, Breiman = FALSE)
  }
}

```

```

        #skutečna predikční chyba
    }
    return(list(err = err, trueErr = trueErr))}

```

Počet bootstrapových odhadů jednotlivých optimismů byl stanoven na $B = 50$. I zde je možné nastavením kritéria Breiman vybrat mezi odhadem predikční chyby a odhadem chyby modelu.

Další funkcí, kterou je nutné zmínit, je `getTrueErr`, která je obsažena ve funkcích `bootstr` a `crossval` a která vrací hodnotu skutečné predikční funkce. Opět je v této funkci obsažena logická podmínka Breiman. Pro `Breiman = FALSE` je obdržena hodnota skutečné predikční chyby podle vzorce (44), pro `Breiman = TRUE` je obdržena hodnota skutečné chyby modelu podle vzorce (46).

```

getTrueErr = function(dataset, G, mod, sigmaEps2, Breiman) {
  # dataset ... dataset
  # G ... kovarianční matice vysvětlujících promenných
  # mod ... vektor regresních parametrů (modifikovaný)
  # sigmaEps2 ... rozptyl náhodné složky
  # Breiman ... typ odhadu predikční chyby (TRUE nebo FALSE)

  n = ncol(dataset) - 1 # počet regresorů
  N = nrow(dataset) # rozsah výběru
  trueErr = rep(NA, n + 1) # sem se ukládají skutečné chyby
  ourNames = colnames(dataset)[-1] # názvy bez vysvětlované proměnné

  for(i in 1:(n + 1)) {
    coefVec = rep(0, n) # vektor s odhady parametrů
    names(coefVec) = ourNames
    fit = lm(yn ~.-1, data = dataset)
    coefVec[colnames(dataset)[-1]] = coef(fit)
    dif = coefVec - mod # rozdíl mezi odhadnutým a skutečným vektorem

    if (Breiman) { # Logická podmínka
      trueErr[i] = N * t(dif) %*% G %*% dif #Breiman = TRUE: skutečná chyba
        modelu (vzorec 46)
    } else {
      trueErr[i] = t(dif) %*% G %*% dif + sigmaEps2 #Breiman = FALSE:
        skutečná predikční chyba (vzorec 44)
    }
  }
  U = summary(fit)$coef

```

```

        dataset = dataset[,!(colnames(dataset) %in%
            rownames(U)[which.max(U[,4])]), drop = FALSE]
        #novy datovy soubor po provedeni selekce
    }
    return(trueErr)
}

```

Na závěr je třeba uvést, jakým způsobem jsem získal hodnoty klasických kritérií (AIC, BIC, R^2 , R^2_{adj}). K tomuto účelu už nebylo nutné sestavit funkci, stačilo pouze načíst hodnoty parametrů *sim*, *nrow*, *mu* a *N* a spustit přes sebe dvě smyčky. V první smyčce je pro každou simulaci (250) generován datový soubor, v druhé smyčce pak probíhá selekce proměnných metodou Backward. V každém kroku selekce jsou poté spočtena všechna klasická kritéria.

```

Rsqr = matrix(NA,ncol= sim, nrow = n) # sem se ukladaji hodnoty koeficientu
determinace
Radjsqr = matrix(NA,ncol= sim, nrow = n) # sem se ukladaji hodnoty
korigovaného koeficientu determinace
AIC1 = matrix(NA,ncol= sim, nrow = n) # sem se ukladaji hodnoty AIC
BIC1 = matrix(NA,ncol= sim, nrow = n) # sem se ukladaji hodnoty BIC

for (a in 1:sim) { #prvni smycka
    A = mvrnorm(n = N, mu, G, tol = 0, empirical = FALSE)
    noise = rnorm(N)
    yn = (A %*% mod) + noise
    dataset = data.frame(yn,A) #datovy soubor

    for (b in 1:n) { #druha smycka
        fit = lm(yn ~.-1, data=dataset)
        Rsqr[b,a] = summary(fit)$r.squared #vypocet koeficientu determinace
        Radjsqr[b,a] = summary(fit)$adj.r.squared #vypocet korigovaneho
koeficientu determinace
        AIC1[b,a] = AIC(fit) #vypocet AIC
        BIC1[b,a] = BIC(fit) #vypocet BIC

        U = summary(fit)$coef
        dataset = dataset[,!(colnames(dataset) %in%
            rownames(U)[which.max(U[,4])]), drop = FALSE]
        #novy datovy soubor po provedeni selekce
    }
}

```

4.7 Výsledky simulací

Nejprve shrnu nastavení simulací. Na datový soubor definovaná podle kapitoly 4.3 byla aplikována metoda selekce regresního submodelu Backward. Bylo posuzováno celkem osm kritérií, jejichž účelem je v daném datovém souboru nalézt optimální regresní submodel z hlediska předpovědi pozorování. Optimální regresní submodel závisel na skutečném vektoru regresních koeficientů. Byly posuzovány varianty $h = 1$ a $h = 2$. Z klasických kritérií se jednalo o koeficient determinace, korigovaný koeficient determinace, Akaikeho informační kritérium a Bayesovo informační kritérium. Ze simulačních kritérií se jednalo o odhad predikční chyby pomocí křížové validace (5-násobná, 10 násobná, LOOCV) a vylepšeným bootstrapovým odhadem. Nastavení parametrů simulací je uvedeno v tabulce 2.

Počet simulací	250
Rozsah generovaného souboru (pozorování)	160
Počet bootstrapových výběrů*	50
K^{**}	5, 10

Tabulka 2: Parametry simulací

Hlavním sledovaným kritériem byla střední čtvercová odchylka počtu vysvětlujících proměnných odpovídajících optimálnímu regresnímu submodelu každé simulace (podle daného kritéria) od skutečného optimálního počtu vysvětlujících proměnných MSE_{Dim} . Pro metody křížové validace a bootstrapu jsem pak zjišťoval, s jakou přesností metody dokážou odhadnout skutečnou predikční chybu.

4.7.1 Hledání optimálního regresního submodelu

Podle kapitoly 4.5 je skutečný optimální počet vysvětlujících proměnných chápán jako průměrná hodnota počtu vysvětlujících proměnných odpovídající minimu skutečné predikční chyby. Tento průměr je pro každé kritérium počítán přes všechny simulace a značím ho Dim_{opt} . Jeho hodnoty jsou pro každou variantu uvedeny v tabulce 3.

h	Dim_{opt}
1	3
2	5,968

Tabulka 3: Skutečný optimální počet vysvětlujících proměnných. Zdroj: Vlastní simulace

* Platí pouze pro vylepšený bootstrapový odhad predikční chyby

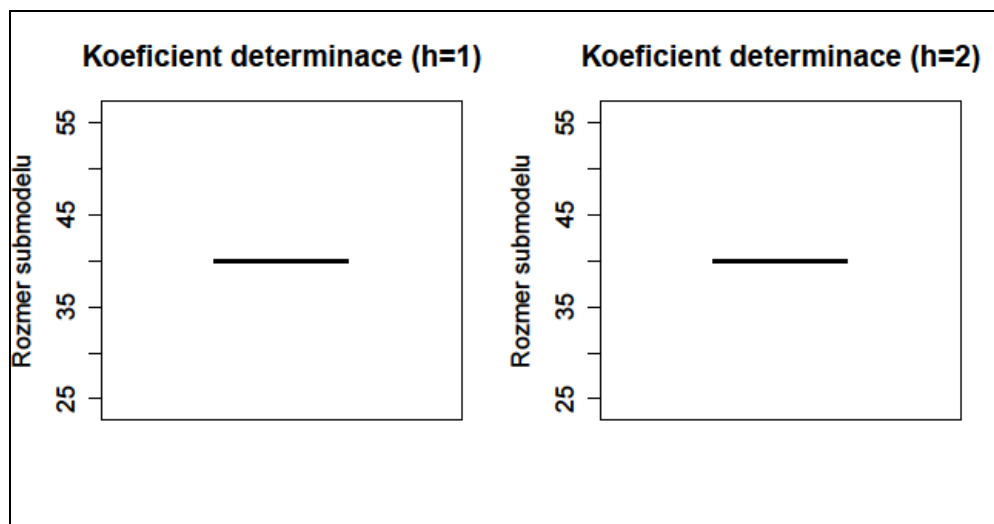
** Platí pouze pro odhad predikční chyby K-násobnou křížovou validací.

Pro variantu $h = 1$ byl skutečný optimální počet vysvětlujících proměnných Dim_{opt} přesně 3. Pro variantu $h = 2$ fluktoval skutečný optimální počet vysvětlujících proměnných kolem šesti vysvětlujících proměnných (v závislosti na hodnotách náhodné složky). Zajímala mě přesnost, s jakou kritéria odhadovala tento skutečný optimální rozměr. K tomuto účelu jsem použil odmocninu ze střední čtvercové odchylky počtu vysvětlujících proměnných odpovídajících optimálnímu regresnímu submodelu každé simulace od skutečného optimálního počtu vysvětlujících proměnných (42). Čím nižší odmocnina MSE_{Dim} tím méně jsou odhady rozptýlené kolem Dim_{opt} a tím jsou při hledání tohoto rozměru přesnější.

Odmocnina z MSE_{Dim}		
Klasické kritérium	$h = 1$	$h = 2$
R^2	37,00	34,03
R^2_{adj}	14,15	10,70
AIC	8,42	5,19
BIC	1,45	2,33

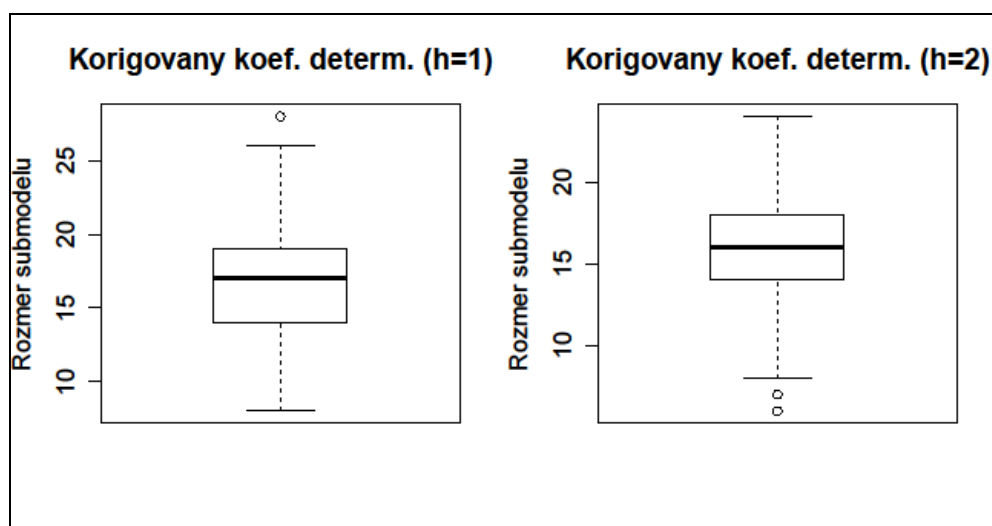
Tabulka 4: Odmocnina z MSE_{Dim} (klasická kritéria). Zdroj: Vlastní simulace

Nejdříve se budu zabývat výsledky klasických kritérií, které jsou uvedené v tabulce 4. V kapitole 2.4 jsem hovořil o tom, že slabina koeficientu determinace R^2 spočívá v tom, že se jeho hodnota nikdy nesníží se zařazením nové vysvětlující proměnné do regresního modelu. Pro mnou generovaná data byl podle tohoto kritéria nejlepší regresní model z hlediska předpovědi vždy plný model, tj. model obsahujících všech čtyřicet proměnných. Krabičkový graf na obrázku 6 ukazuje, že pro obě zvolené varianty ($h = 1$ a $h = 2$) dopadla selekce regresního submodelu stejně v každé simulaci. Tomu odpovídají i hodnoty odmocniny z MSE_{Dim} . Z toho vyplývá, že koeficient determinace není vhodným kritériem při hledání optimálního regresního submodelu z hlediska predikce.



Obrázek 6: Krabíkový graf odhadnutých optimálních rozměrů regresního modelu podle R^2 . Nalevo pro variantu $h = 1$, napravo pro $h = 2$. Zdroj: Vlastní simulace

Korigovaný koeficient determinace R_{adj}^2 definovaný vzorcem (10) přináší zlepšení v hledání optimálního regresního submodelu. Penalizace tohoto koeficientu spočívá v tom, že jeho hodnota vzroste, jen pokud vzroste průměrná vysvětlená variabilita vzhledem k počtu vysvětlujících proměnných v regresním submodelu. Přidáním nadbytečných proměnných by tedy neměla hodnota R_{adj}^2 vzrůst. Krabíkový graf na obrázku 7 ukazuje četnosti, s jakými korigovaný koeficient determinace volil počet proměnných v optimálním regresním submodelu.

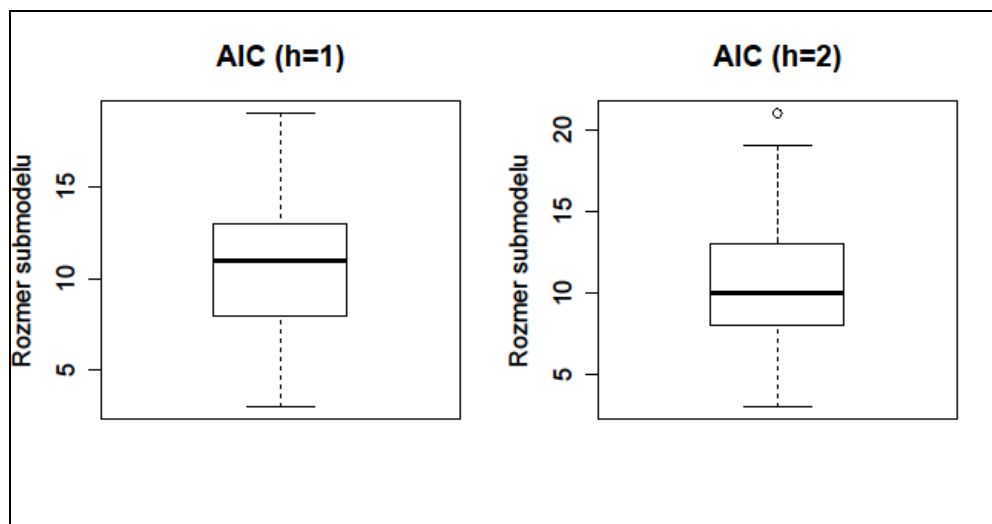


Obrázek 7: Krabíkový graf odhadnutých optimálních rozměrů regresního modelu podle R_{adj}^2 . Nalevo pro variantu $h = 1$, napravo pro $h = 2$. Zdroj: Vlastní simulace

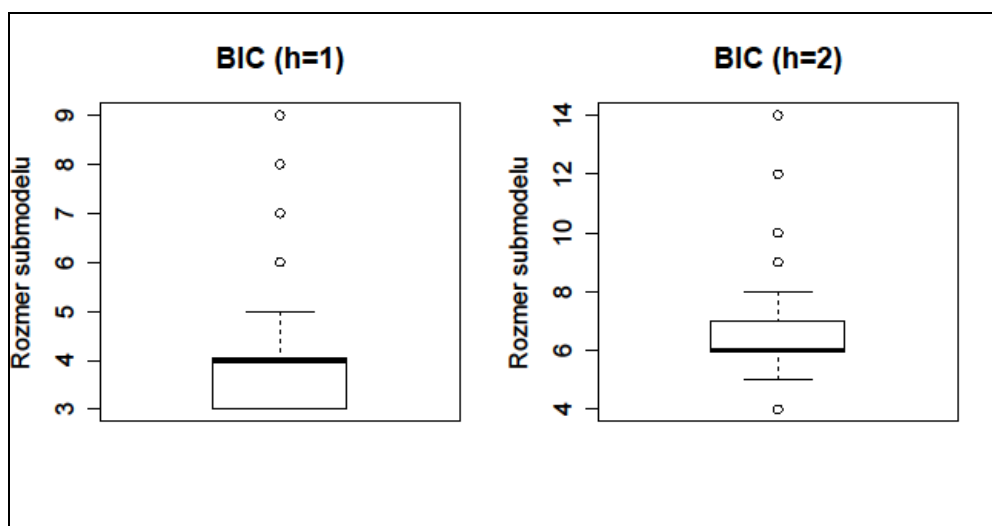
Pro variantu skutečného vektoru regresních koeficientů $h = 1$ byl nejčastěji volený regresní submodel čítající 18 vysvětlujících proměnných. Medián byl 17 proměnných v modelu. Pro variantu $h = 2$ byla modální hodnota také 18 vysvětlujících proměnných, mediánová pak 16 vysvětlujících proměnných. Nižší hodnoty odmocniny z MSE_{Dim} vypovídají o tom, že korigovaný koeficient determinace je přesnějším kritériem než koeficient determinace. Nicméně penalizace tohoto kritéria není dostatečná a při selekci regresního submodelu má toto kritérium tendenci upřednostňovat přeuročené regresní modely. Ani R_{adj}^2 není příliš vhodným kritériem.

Akaikeho informační kritérium (11) se také neukazuje být vhodným kritériem při hledání optimálního regresního submodelu (viz obrázek 8). I přes penalizaci, která je úměrná počtu vysvětlujících proměnných v regresním modelu, byly podle tohoto kritéria nejčastěji voleny regresní modely s 10 ($h = 1$) a 12 ($h = 2$) vysvětlujícími proměnnými. Mediánové hodnoty počtu proměnných byly 11 ($h = 1$) a 13 ($h = 2$). Odchylka od skutečného optimálního rozměru Dim_{opt} je sice nižší než u obou koeficientů determinace, pořád ale nejedná o příliš přesný odhad skutečného rozměru.

Nejzajímavější výsledky z klasických kritérií přineslo Bayesovo informační kritérium (viz obrázek 9). To oproti Akaikeho informačnímu kritériu uplatňuje ještě větší penalizaci modelů s větším počtem proměnných (12). Pro variantu $h = 1$ bylo toto kritérium z hlediska konzistence odhadu skutečného rozměru přesnější než odhad predikční chyby metodou LOOCV. Pro variantu $h = 2$ byla konzistentnější při hledání skutečného rozměru než všechny odhady predikční chyby.



Obrázek 8: Krabičkový graf odhadnutých optimálních rozměrů regresního modelu podle AIC. Nalevo pro variantu $h = 1$, napravo pro $h = 2$. Zdroj: Vlastní simulace



Obrázek 9: Krabičkový graf odhadnutých optimálních rozměrů regresního modelu podle BIC. Nalevo pro variantu $h = 1$, napravo pro $h = 2$. Zdroj: Vlastní simulace

Zatímco klasická kritéria byla posuzována z pohledu četnosti, kritéria odhadu predikční chyby pomocí simulačních metod jsou posuzována z hlediska jejich průměrného chování. Grafické výstupy jsou pak uvedeny v následující kapitole. Přesnost odhadu skutečného optimálního rozměru vyjádřená odmocninou z MSE_{Dim} je uvedena pro každou simulační metodu v tabulce 5.

Odmocnina z MSE_{Dim}		
	$h = 1$	$h = 2$
10-násobná křížová validace	1,42	3,37
5-násobná křížová validace	0,77	2,57
LOOCV	3,77	4,68
Vylepšený bootstrap	0,90	5,07

Tabulka 5: Odmocnina z MSE_{Dim} (simulační kritéria).

Zdroj: Vlastní simulace

Simulační kritéria si při selekci optimálního regresního submodelu počínala dobře bez rozdílu. Počet proměnných, který odpovídal průměrné minimální hodnotě odhadu predikční chyby, byl jak pro $h = 1$ tak pro $h = 2$ blízky Dim_{opt} . Pro variantu $h = 1$ se jednalo o hodnotu 3, pro variantu $h = 2$ byl počet vysvětlujících proměnných 6.

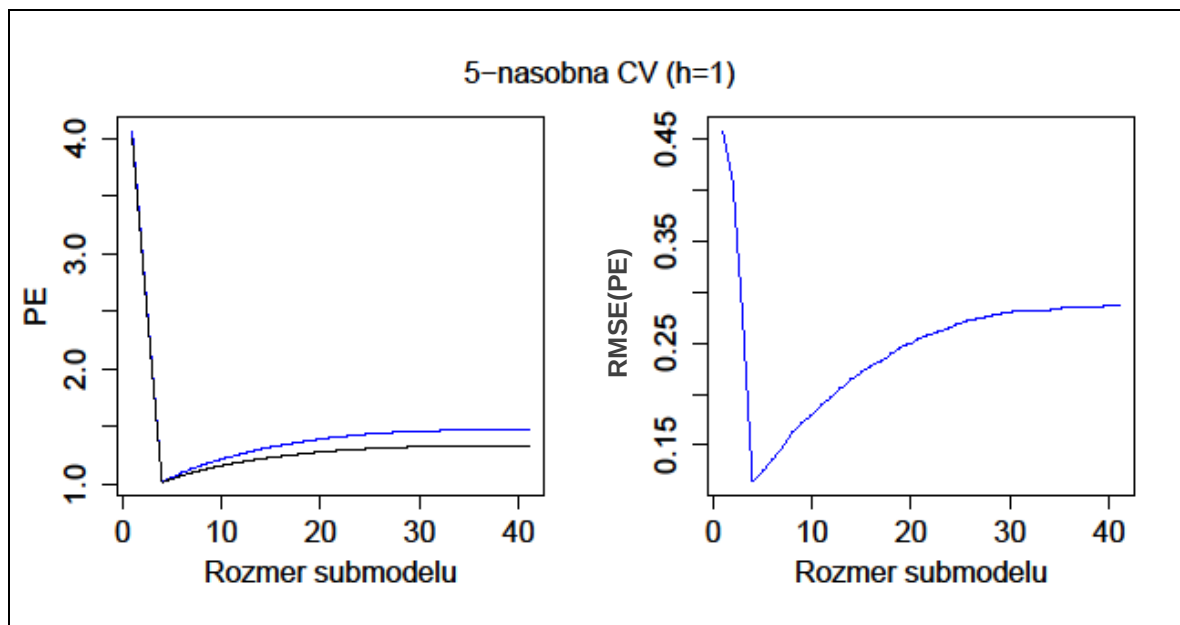
Přesnost simulačních kritérií v odhadu skutečného optimálního rozměru byla téměř vždy signifikantně vyšší než u klasických kritérií. Výjimku tvořilo pouze Bayesovo informační kritérium. Ze simulačních kritérií pak jako nejméně konzistentní vychází metoda křížové validace LOOCV. Nejvíce konzistentní metodou se ukazuje být metoda K-násobná. Překvapivé je, že ačkoliv je v praxi velmi doporučovaná metoda 10-násobná křížová validace, 5-násobná křížová validace dosahovala nižších hodnot odmocniny z MSE_{Dim} pro obě varianty h . Metoda vylepšeného bootstrapového odhadu pak vykazuje proměnlivou konzistenci. Zatímco pro variantu $h = 1$ se zdála být nejpřesnější, pro variantu $h = 2$ zaznamenal MSE_{Dim} signifikantní nárůst.

Na základě simulací, které jsem provedl, vyplývá, že z pohledu nalezení optimálního regresního submodelu je nejvhodnější používat metodu K-násobná křížová validace a vylepšený bootstrapový odhad. Z klasických kritérií je nejvhodnějším Bayesovo informační kritérium.

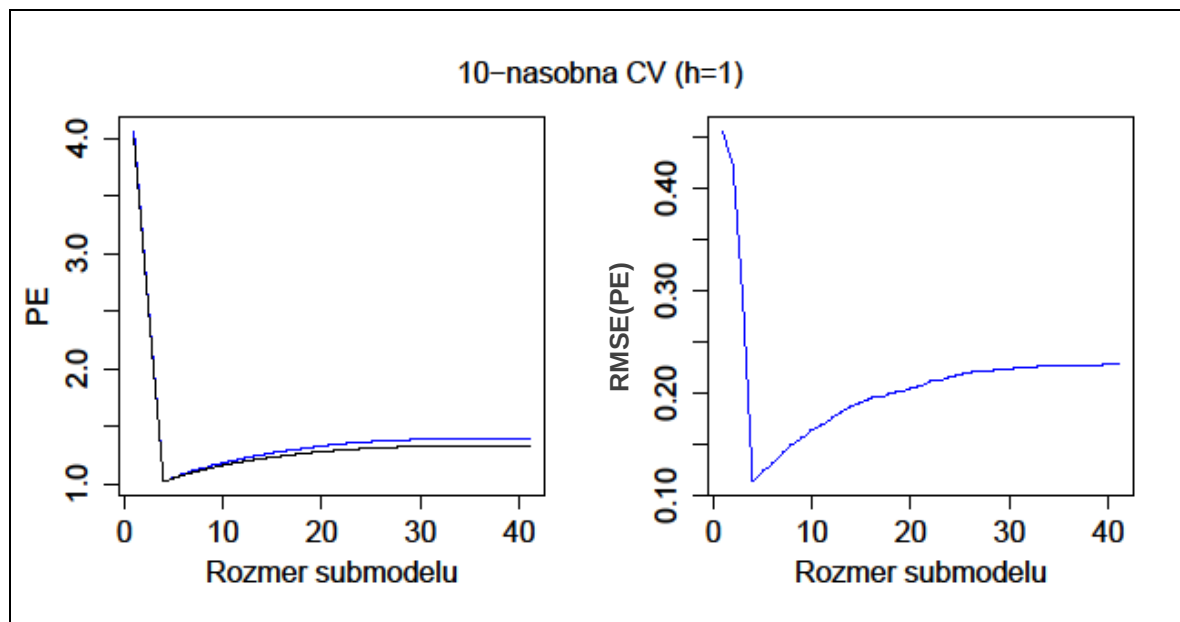
4.7.2 Odhad predikční chyby a její MSE

Metody bootstrapu a křížové validace odhadují stejnou charakteristiku – predikční chybu. Z toho důvodu lze tyto odhady vzájemně porovnávat a na základě toho usoudit, která z uvedených metod nejlépe charakterizuje skutečnou predikční schopnost regresních submodelů. K tomuto účelu jsem využil charakteristiky $RMSE(PE)$ definovanou vzorcem (50). Jedná se o odmocninu ze střední čtvercové odchylky odhadu predikční chyby od skutečné hodnoty predikční chyby $\overline{PE}_{true}$. Tato hodnota je počítána pro každý krok selekce

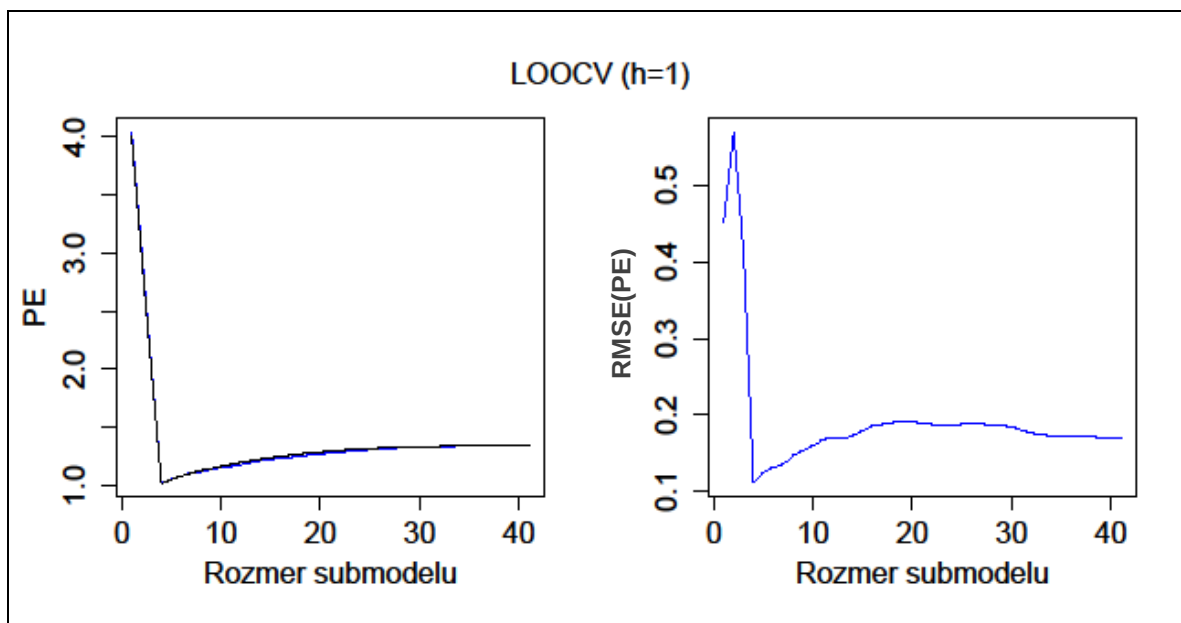
regresního submodelu. Její průběh je pro každou simulační metodu zachycen v pravé části obrázků 10 až 17. V levé části těchto obrázků se nachází průběh průměrné hodnoty odhadu predikční chyby (modrá čára) společně s průběhem skutečné predikční chyby. Obrázky 10 až 13 popisují situaci pro variantu $h = 1$, obrázky 14 až 17 pro $h = 2$.



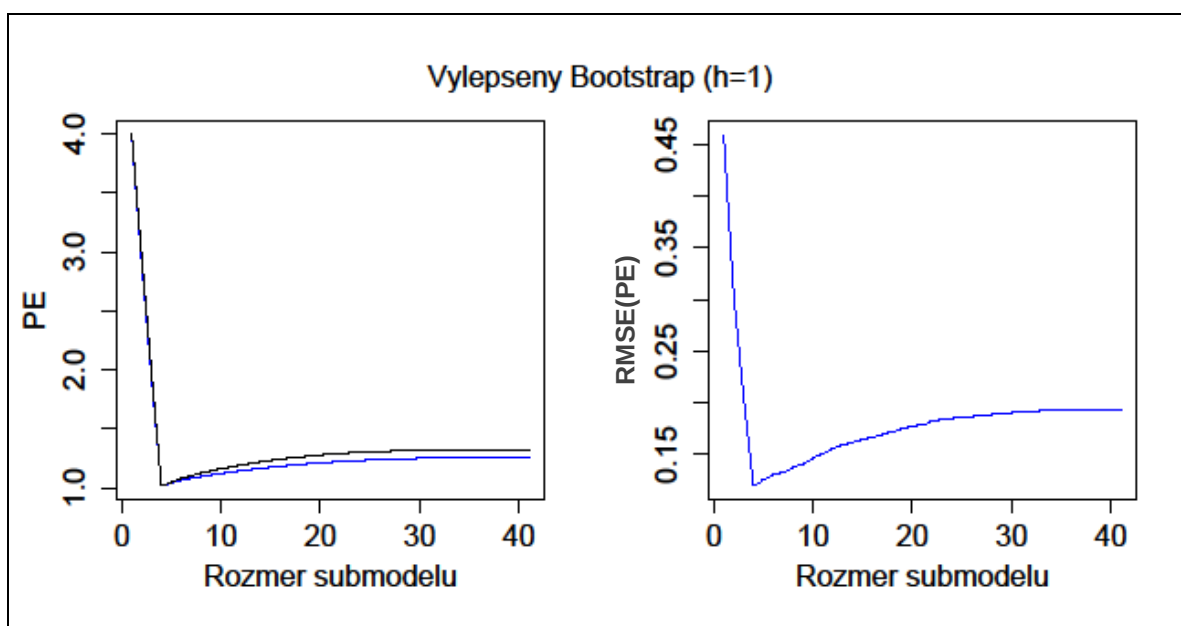
Obrázek 10: 5-násobná křížová validace pro $h = 1$. V levé části obrázku je zachycen průběh odhadu predikční chyby (modrá čára) společně s průběhem skutečné predikční chyby (černá čára). V pravé části je zachycen průběh RMSE(PE). Zdroj: Vlastní simulace



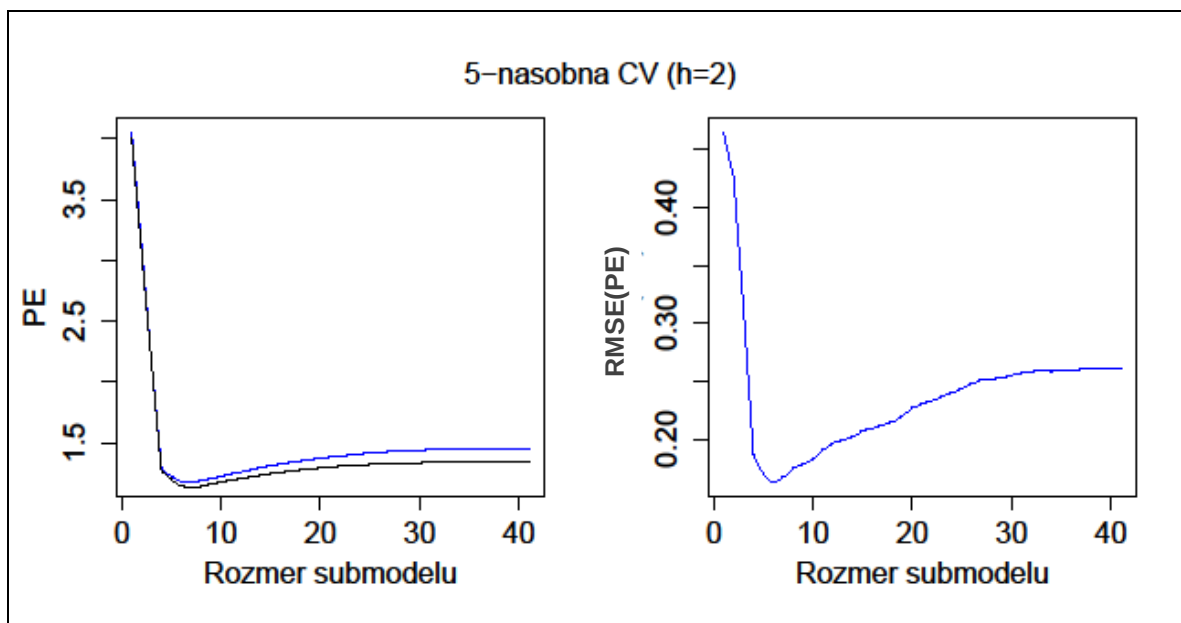
Obrázek 11: 10-násobná křížová validace pro $h = 1$. V levé části obrázku je zachycen průběh odhadu predikční chyby (modrá čára) společně s průběhem skutečné predikční chyby (černá čára). V pravé části je zachycen průběh RMSE(PE). Zdroj: Vlastní simulace



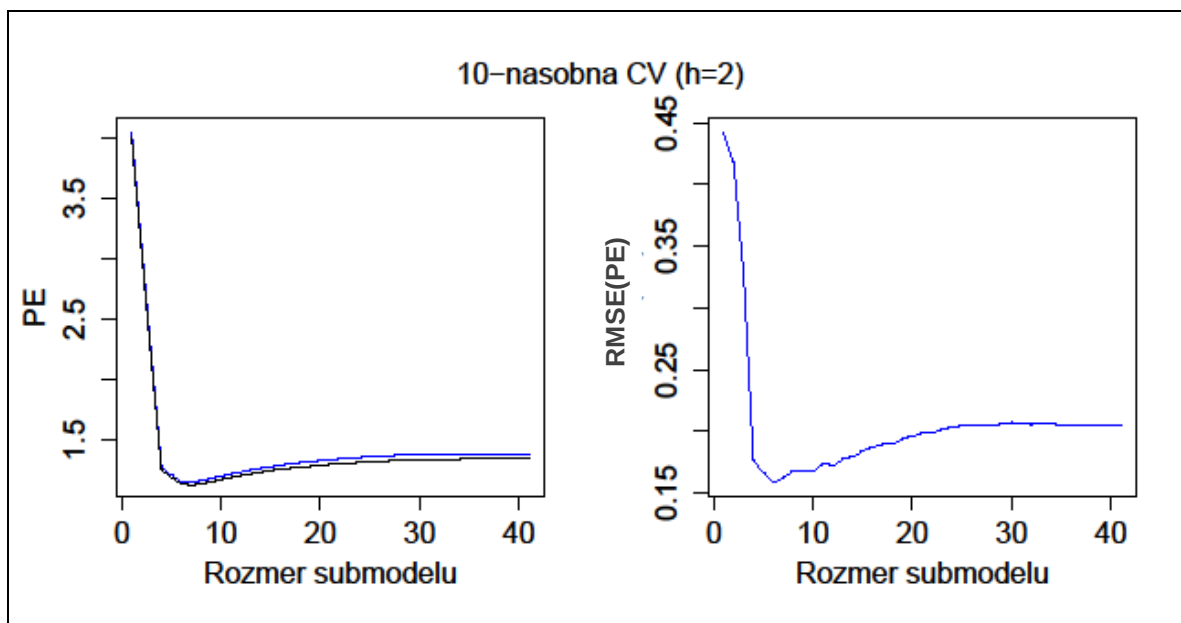
Obrázek 12: Křížová validace LOOCV pro $h = 1$. V levé části obrázku je zachycen průběh odhadu predikční chyby (modrá čára) společně s průběhem skutečné predikční chyby (černá čára). V pravé části je zachycen průběh RMSE(PE). Zdroj: Vlastní simulace



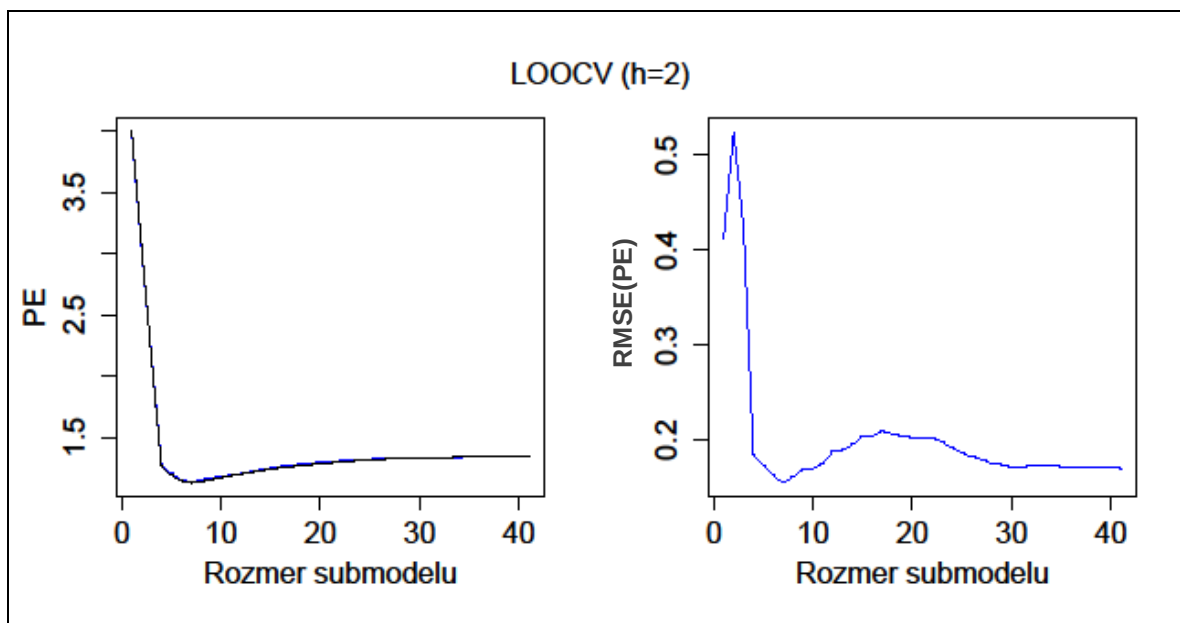
Obrázek 13: Vylepšený bootstrap pro $h = 1$. V levé části obrázku je zachycen průběh odhadu predikční chyby (modrá čára) společně s průběhem skutečné predikční chyby (černá čára). V pravé části je zachycen průběh RMSE(PE). Zdroj: Vlastní simulace



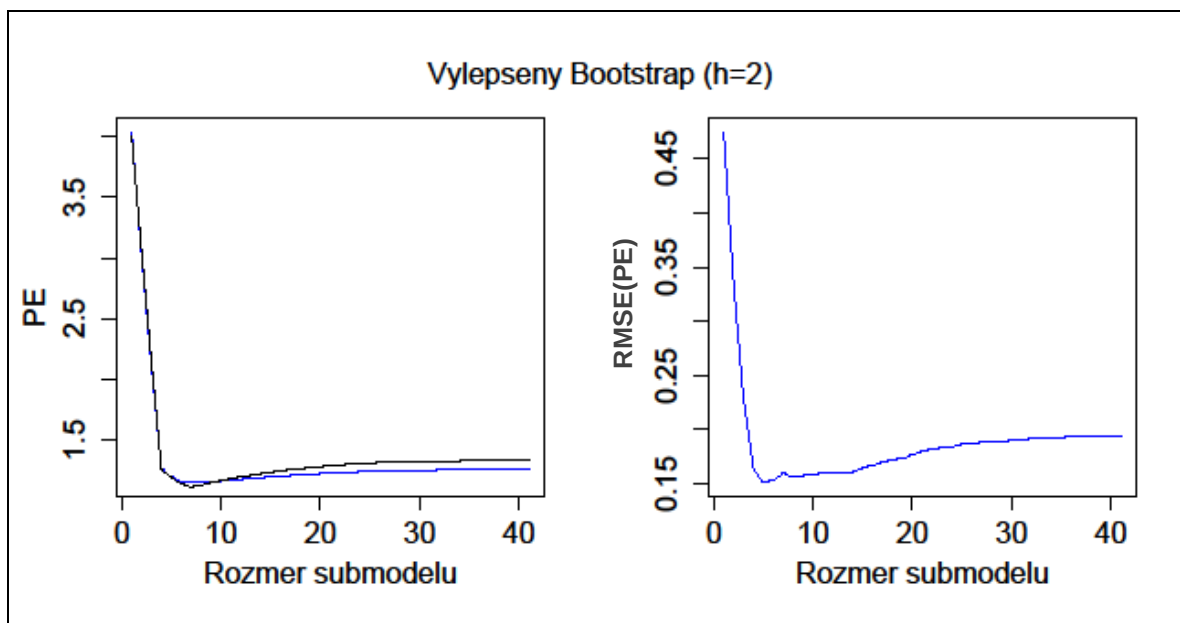
Obrázek 14: 5-násobná křížová validace pro $h = 2$. V levé části obrázku je zachycen průběh odhadu predikční chyby (modrá čára) společně s průběhem skutečné predikční chyby (černá čára). V pravé části je zachycen průběh RMSE(PE). Zdroj: Vlastní simulace



Obrázek 15: 10-násobná křížová validace pro $h = 2$. V levé části obrázku je zachycen průběh odhadu predikční chyby (modrá čára) společně s průběhem skutečné predikční chyby (černá čára). V pravé části je zachycen průběh RMSE(PE). Zdroj: Vlastní simulace



Obrázek 16: Křížová validace LOOCV pro $h = 2$. V levé části obrázku je zachycen průběh odhadu predikční chyby (modrá čára) společně s průběhem skutečné predikční chyby (černá čára). V pravé části je zachycen průběh $RMSE(PE)$. Zdroj: Vlastní simulace



Obrázek 17: Vylepšený bootstrap pro $h = 2$. V levé části obrázku je zachycen průběh odhadu predikční chyby (modrá čára) společně s průběhem skutečné predikční chyby (černá čára). V pravé části je zachycen průběh $RMSE(PE)$. Zdroj: Vlastní simulace

Obrázky 10 a 14 popisují průměrné odhady predikční chyby a $RMSE(PE)$ v každém kroku selekce regresního submodelu. Obrázky 11 a 15 popisují totéž pro 10-násobnou křížovou validaci. Je z nich patrné, že odhady predikční chyby (modrá čára v grafu na obrázku) s využitím metody K-násobné jsou v průměru vyšší, než jsou hodnoty skutečné predikční chyby (černá čára). Toto lze pozorovat jak pro variantu $h = 1$, tak $h = 2$. To znamená, že odhady pořízené těmito metodami vyhodnocují predikční schopnosti regresních submodelů jako horší, než ve skutečnosti jsou. Zatímco u odhadů predikční chyby 5-násobnou křížovou validací je vychýlení relativně významné, u 10-násobné křížové validace je vychýlení výrazně menší. I $RMSE(PE)$ 10-násobné křížové validace je v průměru nižší než u 5-násobné. Díky tomu, lze odhad predikční chyby 10-násobnou křížovou validací považovat za přesnější.

Obrázky 12 a 16 zachycují odhady predikční chyby a $RMSE(PE)$ pro křížovou validaci LOOCV. Z nich je zřejmé, že odhady predikční chyby pořízené touto metodou jsou téměř nevychýlené (modrá čára odhadu predikční chyby téměř dokonale splývá s čarou znázorňující průběh skutečné predikční chyby). Hodnoty $RMSE(PE)$ jsou nižší, než u 5-násobné křížové validace, a velmi podobné hodnotám $RMSE(PE)$ 10-násobné křížové validace. Stejně jako 10-násobná křížová validace, i odhad predikční chyby metodou LOOCV se pro mnou simulovaná data zdá být velmi přesným.

Na obrázcích 13 a 17 se nachází odhady predikční chyby vylepšeným bootstrapem. Ty jsou v průměru nižší, než je skutečná predikční chyba. Bootstrap má tedy tendenci dělat optimističtější odhady predikční chyby než třeba K-násobná křížová validace. Jeho vychýlení je ale relativně nízké. Společně s velmi nízkými hodnotami $RMSE(PE)$, lze odhad predikční chyby pomocí vylepšeného bootstrapu považovat za přesný odhad skutečné predikční chyby.

Na závěr uvádím průměrné hodnoty odhadu predikční funkce v optimálním regresním submodelu, který byl vyhodnocen podle jednotlivých metod.

Odhad predikční chyby	$h = 1$	$h = 2$
10-násobná křížová validace	1,0205	1,1489
5-násobná křížová validace	1,0240	1,1706
LOOCV	1,0235	1,1302
Vylepšený bootstrap	1,0199	1,1506

Tabulka 6: Odhady predikční chyby v optimálním regresním submodelu.
Zdroj: Vlastní simulace

5 Závěr

V této práci jsem se zabýval nalezením vhodného kritéria, pomocí kterého lze vybrat z velkého počtu vysvětlujících proměnných takový regresní submodel, který by byl neoptimálnější z hlediska předpovědi nových hodnot vysvětlované proměnné. Nejdříve jsem se věnoval kritériím, která jsou v praxi velmi využívána. Jednalo se o koeficient determinace, korigovaný koeficient determinace, Akaikeho informační kritérium, Bayesovo informační kritérium. Jako alternativu jsem představil metody bootstrap a křížovou validaci, které vyhodnocují predikční schopnosti regresních submodelů pomocí opakovaných odhadů ztrátové funkce, která se v tomto kontextu nazývá predikční chybou. Názorně jsem ukázal, jak tyto metody naprogramovat v softwaru R.

Ze simulací, které jsem provedl, vyplývá, že koeficient determinace, upravený koeficient determinace a Akaikeho informační kritérium není vhodné používat jako kritéria pro volbu regresního submodelu pro případy, že je účelem hledání regresního submodelu s nejlepšími predikčními schopnostmi. Tato kritéria mají tendenci volit přeuročené modely, tj. modely, které obsahují větší počet vysvětlujících proměnných. Takto zvolené modely pak dobře pasují na datový soubor, ale jejich předpovědi vykazují nižší přesnost.

Zajímavou výjimku tvořilo Bayesovo informační kritérium, které oproti původnímu předpokladu, dokázalo nalézt optimální počet vysvětlujících proměnných pro obě testované varianty $h = 1$ a $h = 2$. Z výsledků simulací vyplývá, že se toto kritérium osvědčuje při hledání regresního submodelu pro účel konstrukce předpovědi.

Dále ze simulací vyplývá, že ačkoliv metody bootstrapu a křížové validace jsou v průměru úspěšné při hledání optimálního regresního submodelu, K-násobná křížová validace a vylepšený bootstrapový odhad mají nejnížší rozptyl odhadovaného počtu proměnných kolem skutečné hodnoty.

Na závěr jsem porovnával to, která z uvedených metod nejlépe odhaduje skutečnou predikční schopnost regresních submodelů. V tomto ohledu se nejlépe osvědčily metody K-násobná, LOOCV a vylepšený bootstrapový odhad.

Z tohoto pohledu považuji cíle této práce za splněné.

6 Přehled literatury a zdrojů

6.1 Knižní zdroje

1. BÍLKOVÁ, Diana, Petr BUDINSKÝ a Václav VOHÁNKA. Pravděpodobnost a statistika. Plzeň. ISBN 978-80-7380-224-0.
2. EFRON, Bradley a Robert TIBSHIRANI. A introduction to the bootstrap. Boca Raton: Chapman, c1993, xvi, 436 s. Monographs on statistics and applied probability. ISBN 04-120-4231-2.
3. GREENE, William H. Econometric analysis. 5th ed. Upper Saddle River, N.J.: Prentice Hall, c2003, xxx, 1026 p. ISBN 01-306-6189-9.
4. HASTIE, Trevor, Robert TIBSHIRANI a J FRIEDMAN. The elements of statistical learning: data mining, inference, and prediction. 2nd ed. New York, NY: Springer, c2009, xxii, 745 p. ISBN 978-038-7848-587.
5. HEBÁK, Petr. Vícerozměrné statistické metody. Vyd 1. Praha: Informatorium, 2005, 239 s. ISBN 80-733-3036-9.
6. JAMES, Gareth, Daniela WITTEN, Trevor HASTIE a Robert TIBSHIRANI. An introduction to statistical learning: with applications in R. 2013, xvi, 426 pages. Springer texts in statistics, 103. ISBN 978-1-4614-7138-7.

6.2 Články

1. ARLOT, Sylvain a Alain CELISSE. A survey of cross-validation procedures for model selection. Statistics Surveys. 2010, vol. 4, issue 0, s. 40-79. DOI: 10.1214/09-SS054. Dostupné z: <http://projecteuclid.org/euclid.ssu/1268143839>
2. BREIMAN, Leo a Philip SPECTOR. International statistical review Revue, Vol. 60, No. 3 (Dec., 1992): Submodel Selection and Evaluation in Regression. The X-Random Case. pp. 291-319. ISBN 1751-5823. Dostupné z: http://www.stat.washington.edu/courses/stat527/s13/readings/BreimanSpector_1992.pdf

3. KOHAVI, Ron. A study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. [online]. roč. 1995, s. 7 [cit. 2014-11-11]. Dostupné z: <http://www.cs.iastate.edu/~jtian/cs573/Papers/Kohavi-IJCAI-95.pdf>
4. LIU, Ling a OZSU, M. Tamer. Encyclopedia of database systems. New York: Springer, c2009, 5 v. (lxix, 3749 p.). ISBN 978-038-7496-160. Dostupné z: <http://leitang.net/papers/ency-cross-validation.pdf>

7 Přílohy

Příloha č. 1: Datový soubor *felicie*

n	cena	stari	najeto
1	167	3	106
2	139	4	134
3	159	4	51
4	135	5	102
5	139	5	125
6	139	6	104
7	139	6	49
8	145	6	74
9	109	7	156
10	119	7	147
11	129	7	59
12	135	7	83
13	99	8	137
14	99	8	91
15	109	8	114
16	119	8	97
17	63	9	298
18	69	9	165
19	76	9	172
20	77	9	145

Příloha č. 2: Odvození koeficientu q

Pro odvození koeficientu q je nutné vyjít z rovnice teoretického koeficientu determinace. Ten je definován vzorcem (49).

$$R_{Teor}^2 = 1 - \frac{D(\epsilon)}{D(y)} \quad (49)$$

Ve vzorci (49) je $D(\epsilon)$ rozptyl náhodné složky. Ten byl předem zvolen roven 1 ($D(\epsilon) = 1$). Ve jmenovateli se nachází rozptyl hodnot vysvětlované proměnné y . Hodnota R_{Teor}^2 je nastavena na 0,75. Z toho plyne, že lze vzorec (49) vyjádřit podle (50).

$$D(y) = 4 \quad (50)$$

Za předpokladu, že je matice pozorování $\mathbf{X}$ nenáhodná, odpovídá rozptyl y rozptylu náhodné složky (Hebák a kol. 2005, str 40). To pro případ generované matice pozorování neplatí a rozptyl vysvětlované proměnné lze rozložit podle vzorce (35):

$$D(y) = D(\mathbf{X}\boldsymbol{\beta}_h^*) + D(\boldsymbol{\epsilon}). \quad (51)$$

$D(\boldsymbol{\epsilon})$ je rovno 1, takže lze vzorec (51) zapsat jako:

$$3 = D(\mathbf{X}\boldsymbol{\beta}_h^*). \quad (52)$$

Ze vzorce je (51) lze vytknout vektor regresních koeficientů $\boldsymbol{\beta}_h^*$. Tento vektor odpovídá již novému, modifikovanému vektoru, který odpovídá hodnotě teoretického koeficientu determinace 0,75. V závorce poté zůstane pouze člen $D(\mathbf{X})$, což je akorát jiné vyjádření pro kovarianční matici vysvětlujících proměnných $\boldsymbol{\Sigma}$.

$$3 = \boldsymbol{\beta}_h^{*T} \boldsymbol{\Sigma} \boldsymbol{\beta}_h^* \quad (53)$$

Podle vzorce (40) volím koeficient q , kterým je třeba vynásobit vektor regresních koeficientů $\boldsymbol{\beta}_h$ za účelem získání $\boldsymbol{\beta}_h^*$. Vzorec (53) lze tedy upravit:

$$3 = (q\boldsymbol{\beta}_h^*)^T \boldsymbol{\Sigma} (q\boldsymbol{\beta}_h^*). \quad (54)$$

Nakonec je třeba koeficient q vytknout (55).

$$q = \sqrt{\frac{3}{\boldsymbol{\beta}_h^T \boldsymbol{\Sigma} \boldsymbol{\beta}_h}} \quad (55)$$