UNIVERZITA PALACKÉHO V OLOMOUCI

PŘÍRODOVĚDECKÁ FAKULTA

DIPLOMOVÁ PRÁCE

Feature Selection

Katedra matematické analýzy a aplikací matematiky
Vedoucí diplomové práce: Mgr. Jana Vrbková, Ph.D.
Vypracoval: Bc. Luboš Linhart
Studijní program: N1103 Aplikovaná matematika
Studijní obor: Aplikace matematiky v ekonomii
Forma studia: Prezenční
Rok odevzdání: 2016

BIBLIOGRAFICKÁ IDENTIFIKACE

Autor: Bc. Luboš Linhart

Název práce: Feature Selection

Typ práce: Diplomová práce

Pracoviště: Katedra matematické analýzy a aplikací matematiky

Vedoucí práce: Mgr. Jana Vrbková, Ph.D.

Rok obhajoby: 2016

Abstrakt: Tato práce se zabývá teorií metod Feature Selection a jejich aplikací na datech ve statistickém softwaru R. Cílem práce je porovnat jednotlivé přístupy, zhodnotit jejich výhody a nevýhody. V práci byly představeny metody best subset, forward, backward, stepwise selection a lasso regrese. Taktéž jsou zde uvedeny potřebné balíčky softwaru R, včetně popisu jejich důležitých funkcí. Metody byly aplikovány na logistickou a Coxovu regresi. Výsledky jsou slovně popsány a shrnuty v závěrečných tabulkách.

Klíčová slova: Coxovův regresní model, Logistická regrese, Feature Selection, Best Subset, Forward Selection, Backward Selection, Stepwise selection, Lasso regrese, R

Počet stran: 80

Počet příloh: 0

Jazyk: český

BIBLIOGRAPHICAL IDENTIFICATION

Author: Bc. Luboš Linhart

Title: Feature Selection

Type of thesis: Master's

Department: Department of Mathematical Analysis and Application of Mathematics

Supervisor: Mgr. Jana Vrbková, Ph.D.

The year of presentation: 2016

Abstract: This paper deals with theory of Feature Selection methods and their applications on data in statistical software R. The aim is to compare different approaches and assess their advantages and disadvantages. In this work there were presented the best subset, forward, backward, stepwise selection and lasso regression. Also there are listed R software packages, including a description of their important functions. The methods were applied to logistic and Cox regression. The results are verbally described and summarized in the final tables.

Key words: Cox Regression, Logistick Regression, Feature Selection, Best Subset, Forward Selection, Backward Selection, Stepwise selection, Lasso Regression, R

Number of pages: 80

Number of appendices: 0

Language: Czech

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně pod vedením Mgr. Jany Vrbkové, Ph. D. a všechny použité zdroje jsem uvedl v seznamu literatury.

V Olomouci dne ....
podpis

Obsah

Poděkování

Na tomto místě bych rád poděkoval všem, kteří mě během studia podporovali, především rodičům a kamarádům. Nicméně nejvíce děkuji svojí vedoucí diplomové práce Mgr. Janě Vrbkové, PhD, která měla spoustu trpělivosti a společnými silami jsme práci dotáhli až do zdárného konce.

Úvod

Téma diplomové práce jsem si vybral pro tématickou návaznost na mojí bakalářskou práci, ve které jsem se zabýval statistickou analýzou dat. Jelikož je mi statistický software a software obecně velmi blízký, absolvoval jsem v rámci studia několik kurzů zabývajících se právě výukou práce ve statistickém softwaru. Nabyté znalosti jsem chtěl zúročit právě při psaní diplomové práce a proto se v práci zabývám také praktickou aplikací přístupů popsaných v teoretické části.

Data poskystnutá pro diplomovou práci představují anonymizované soubory pacientů s blíže nespecifikovaným nádorovým onemocněním, které jsou složeny z několika částí – výsledek zpracování údajů z mikročipů, klinické údaje, údaje o přežívání a další údaje. Data budou podrobněji popsána v praktické části diplomové práce. Poskytnutý soubor pozorování bohužel nebylo možné k práci přiložit, poněvadž se nejedná o veřejně přístupné informace.

Jelikož tento datový soubor obsahuje celkem 138 pozorování na 151 proměnných, nevystačíme si tak s klasickými přístupy k modelování dat.

Cílem práce je tedy nastudovat teorii k Feature Selection metodám, které se používají k výběru proměnných do statistického modelu v případě, kdy je těchto proměnných velmi mnoho, a pomocí statistického softwaru R tyto postupy aplikovat na reálná data. Následně po otestování jednotlivých metod na datech bude úkolem u každého přístupu zhodnotit výhody, nevýhody a také postupy porovnat mezi sebou. V rámci dat je základní úlohou nalézt významné faktory, které ovlivňují přežívání pacientů.

U čtenáře se předpokládají znalosti ze základních kurzů matematiky, matematické statistiky a statistického softwaru.

Lineární model

Většinu metod, které jsou vyučovány během základních kurzů statistiky, můžeme vztáhnout do jedné velké skupiny - lineární modely. U lineárních modelů používáme regresní analýzu, analýzu rozptylu (ANOVA) a analýzu kovariance. Někteří výzkumníci, ačkoliv si to neuvědomují, mají i své nejjednodušší analýzy založené na modelech. Modely hrají ve statistickém odvozování velmi důležitou roli. Model je matematický způsob, jak vyjádřit vztah mezi vysvětlovanou proměnnou a množinou nezávislých, vysvětlujících proměnných. Statistické modely, narozdíl od deterministických, berou v úvahu i možnost, že takový vztah nemusí být úplně dokonalý – připouštíme nevysvětlený rozptyl ve formě reziduí. Nejčastěji statistické modely zapisujeme jako

vysvětlovaná proměnná = systematická složka + reziduální složka.

Některé modely jsou samozřejmě lepší než jiné. Úkolem statistika je pak najít takový model, který nejlépe popisuje daný problém a zároveň je co nejjednodušší a také snadno interpretovatelný.

V lineárním modelu pozorovanou hodnotu závisle proměnné y pro $i = 1, 2, ..., n$ modelujeme jako lineární funkci $(p-1)$ takzvaných nezávislých proměnných $x_1, x_2, ..., x_{p-1}$ jako

$$y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_{p-1} x_{i(p-1)} + \varepsilon_i .$$

Maticově tento vztah zapisujeme jako

$$y = X\beta + \varepsilon ,$$

kde

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

je vektor pozorování závisle proměnné,

$$X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1(p-1)} \\ 1 & x_{21} & & \\ \vdots & & \ddots & \\ 1 & x_{n1} & & x_{n(p-1)} \end{pmatrix}$$

je známá matice typu $n \times p$, nazýváme ji dizajnová matice a obsahuje hodnoty nezávislých proměnných a sloupec jedniček odpovídající absolutnímu členu,

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{pmatrix}$$

je vektor obsahující p odhadovaných parametrů (zahrnující i absolutní člen) a

$$\varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

je vektor reziduí. Obecně se předpokládá, že rezidua ε jsou nezávislá, stejně rozdělená a mají stejný rozptyl. Některé modely neobsahují absolutní člen β_0, takové modely pak nemají sloupec jedniček v matici X.

Odhady parametrů v lineárních modelech získáváme nejčastěji použitím metody nejmenších čtverců, která je v tomto případě (platí pouze pro lineární modely) ekvivalentní k metodě maximální věrohodnosti. Hodnoty odhadovaných parametrů jsou hodnoty, pro které je součet čtverců reziduí $\sum_i \varepsilon_i^2$ minimální.

Maticově zapíšeme reziduální součet čtverců (RSS) jako

$$\varepsilon'\varepsilon = (y - X\beta)' - (y - X\beta).$$

Minimalizací vzhledem k parametrům β dostaneme normální rovnice

$$X'X\beta = X'y.$$

Pokud není matice $X'X$ singulární, dostáváme odhad parametrů modelu

$$\hat{\beta} = (X'X)^{-1} X'y.$$

Pokud inverze $X'X$ neexistuje, můžeme i přesto získat řešení, avšak toto řešení nemusí být optimální. V takovém případě můžeme použít zobecněnou inverzi a hledat řešení jako

$$\hat{\beta} = (X'X)^{-} X'y.$$

[1, strana 1-4]

Jednoduchá lineární regrese

V regresní analýze má častodizajnová matice X jeden sloupec, který obsahuje samé jedničky (odpovídá nulovému členu – interceptu), zatímco druhý sloupec představuje hodnoty nezávisle proměnné. Tudíž regresní model $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ s $n = 4$ pozorováními můžeme zapsat pomocí maticového zápisu jako

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \end{pmatrix}.$$

Mnohonásobná regrese

Jedná se o zobecnění jednoduchého lineárního regresního modelu tak, aby zahrnoval více než jednu nezávislou proměnnou. Předpokládejme například, že y závisí na dvou proměnných a že jsme provedli $n = 6$ pozorování. Regresní model je pak $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i$ a maticově zapsáno

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ 1 & x_{31} & x_{32} \\ 1 & x_{41} & x_{42} \\ 1 & x_{51} & x_{52} \\ 1 & x_{61} & x_{62} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{pmatrix}.$$

[1, strana 8-10]

Zobecněné lineární modely

Linární modely jsou pro analýzu velmi důležité, nicméně jsou také limitované v mnoha směrech. Klasické aplikace lineárních modelů jsou založeny na předpokladech normality, linearity a homoskedasticity. Zobecnění lineárních modelů nám umožňuje modelovat data s použitím jiného než normálního rozdělení. Volba rozdělení ovlivňuje stanovené předpoklady ohledně rozptylu, neboť vztah mezi rozptylem a střední hodnotou je pro mnoho rozdělení znám.

V zobecněných lineárních modelech často předpokládáme, že vysvětlovaná proměnná Y je kvantitativní proměnnou a má normální rozdělení. Toto ovšem není jediný typ vysvětlované proměnné, na který můžeme v praxi narazit. [1, strana 33]

Vysvětlované a vystvětlující proměnné mohou být měřeny na některé z následujících škál:

1. nominální klasifikace – např. červená, zelená, modrá; ano, ne, nevím. Konkrétně pro binární, dichotomní nebo binomickou proměnnou existují vždy pouze dvě kategorie – např. muž, žena; živý, mrtvý. Pokud je kategorií více, pak takovou proměnnou nazýváme polytomní nebo multinomická.
2. ordinální klasifikace je taková, kdy mezi kategoriemi existuje nějaký přirozený řád nebo pořadí – např. mladý, ve středním věku, starý; diastolický krevní tlak seskupený následovně $\leq 70, 71-90, 91-110, 111-130, \geq 131$ mm Hg.
3. spojitá škála – kdy mohou pozorování teoreticky nabývat jakékoliv hodnoty, například váha, délka nebo čas. Tato škála zahrnuje jak intervalová, tak poměrová měření. Konkrétním příkladem pak může být doba než nastane nějaká specifická událost, například selhání elektronické součástky.

Nominální a ordinální data se také někdy nazývají kategoriálními nebo diskrétními proměnnými. Jako kvantitativní často označujeme proměnnou, která je měřena na spojité škále. Kvalitativní proměnnou měříme na nominální a někdy také na ordinální škále. Kvalitativní vysvětlující proměnná se nazývá faktor a její kategorie úrovně faktoru. [4, strana 9-10]

Vzhledem k pokrokům ve statistické teorii a počítačovém softwaru, můžeme aplikovat metody vyvinuté k použití na lineárních modelech v následujících zobecněných případech:

1. vysvětlovaná proměnná má jiné rozdělení než normální
2. vztah mezi vysvětlovanou a vysvětlujícími proměnnými nemusí být v jednoduché lineární formě.

Jedním z objevů bylo zjištění, že třída rodiny exponenciálních rozdělení (sem patří Poissonovo, normální a binomické rozdělení) sdílí mnoho užitečných vlastností normálního rozdělení. Dále pak rozšíření numerických metod odhadu parametrů β lineárního modelu tak, aby je bylo možné použít i v případě, kdy existuje nějaká nelineární funkce spojující $E(Y_i) = \mu_i$ s lineární komponentou $x_i^T \beta$:

$$g(\mu_i) = x_i^T \beta .$$

Funkce g se nazývá spojovací (link) funkce. V prvotní formulaci zobecněných lineárních modelů Nelderem a Wedderburnem (1972) je funkce g jednoduchá matematická funkce. [4, strana 50]

Poissonovo rozdělení se užívá k modelování dat ve formě součtů. Typicky jsou to počty výskytů určité události v předem definovaném časovém intervalu nebo prostoru, kdy je pravděpodobnost, že událost v krátkém časovém okamžiku nastane malá a události nastávají nezávisle na sobě. Příkladem může být počet tropických bouří během sezóny nebo počet gramatických chyb na jedné stránce novin.

Normální rozdělení užíváme pro modelování spojitých dat, které mají symetrické rozdělení. Hojně se používá ze tří důvodů:

1. mnoho přirozeně se vyskytujících jevů se dají velmi dobře popsat pomocí normálního rozdělení, například výška nebo krevní tlak,
2. i když data nemají normální rozdělení, průměr nebo součet náhodného výběru hodnot budou mít přibližně normální rozdělení
3. pro normální rozdělení existuje rozvinutá statistické teorie včetně aproximací k dalším rozdělením – v případě, že data nemají normální rozdělení, je tedy vhodné najít transformaci, v důsledku které budu mít data přibližně normální rozdělení.

Binomické rozdělení se obvykle používá k modelování první možnosti z pozorováních procesu s binárním výstupem. Příkladem je například úspěch uchazeče o zaměstnání nebo studenta v testu, diagnóza určitého onemocnění u pacienta. Poissonovo

rozdělení zase použijeme pro počet uchazečů, kteří úspěšně prošli testem nebo počet pacientů trpící nějakou nemocí, kteří jsou v určitém čase od diagnózy stále naživu. [4, strana 51-53]

Logistická regrese (LR)

Vzhledem k tomu, že v rámci praktické části této práce je použita při analýze dat logistická regrese, následuje její krátké představení.

LR byla navržena v 60 tých letech minulého století jako alternativní postup k metodě nejmenších čtverců pro případ, že vysvětlovaná proměnná je binární. V minulosti se týkala většina úloh aplikace LR oblasti medicíny a epidemiologie. Vysvětlovaná proměnná představuje přítomnost nebo nepřítomnost choroby. Příkladem mohou být výpočty rizika vzniku srdeční choroby jako funkce osobních a chování se týkajících charakteristik (věk, váha, krevní tlak, hladina cholesterolu a kouření.

LR je alternativní metodou klasifikace, když nejsou splněny předpoklady vícerozměrného normálního modelu. Může se aplikaovat na libovolnou kombinaci diskrétních nebo spojitých proměnných. Vyžaduje však znalost obou, jak závisle proměnné, tak i nezávisle proměnných analyzovaného výběru. Výsledný model pak může být využit k budoucímu klasifikování, když jsou uživateli dostupné pouze vysvětlující, nezávislé proměnné.

LR se liší od lineární v tom, že predikuje pravděpodobnost dané události, která se buď stala, nebo nestala. Vypočtená pravděpodobnost je tedy rovna buď 0 nebo 1. Aby se vytvořila tato vazební podmínka, užívá LR tzv. logitovou transformaci, která vede na sigmoidální vztah mezi závisle proměnnou y a vektorem nezávislých proměnných x. Rozdíl mezi logistickou a lineární regresí spočívá v tom, že LR používá kategorickou vysvětlovanou proměnnou zatímco lineární regrese užívá pouze spojitou vysvětlovanou proměnnou. Centrální roli zde hraje logitová transformace, která vychází z tzv. poměru šancí (odds ratio, OR). [4., strana 429 - 431]

LR porovnává pravděpodobnost události odehrané $L_{(1)}$ vůči pravděpodobnosti události neodehrané $L_{(0)} = 1 - L_{(1)}$. Využijeme pravděpodobnostní poměr $\frac{L_{(1)}}{L_{(0)}}$, ve kterém je pravděpodobnost $L_{(1)}$ vyjádřena logistickou funkcí

$$L_{(1)} = \frac{1}{1 + e^{C-Z}}.$$

Pravděpodobnostní poměr (poměr šancí, odds ratio – OR) je vyjádřen jako

$$\frac{L_{(1)}}{L_{(0)}} = e^{a_0 + a_1 x_1 + a_2 x_2 + ... + a_p x_p},$$

kde odhadované koeficienty $a_0, a_1, a_2, ..., a_p$ jsou míry změny poměru obou pravděpodobností $\frac{L_{(1)}}{L_{(0)}}$. Poměr je lineární funkcí diskriminační funkce o p nezávislých proměnných

$$Z = a_0 + a_1 x_1 + a_2 x_2 + ... + a_p x_p.$$

Po zlogaritmování a úpravě

$$C - Z = \ln\left(\frac{L_{(0)}}{L_{(1)}}\right),$$

kde C je absolutní člen a_0. [10., strana 82 - 83]

Analýza přežívání

Termín analýza přežívání se používá pro analýzu dat, které mají podobu časových okamžiků s přesně stanoveným počátkem a končí nastáním nějaké konkrétní události nebo dosažením tzv. end-pointu. V medicíně bývá často za počátek zvolen čas vstupu jedince do klinické studie, např. při testování účinnosti dvou nebo více druhů léčby, což také může zahrnovat diagnózu určitého stavu, zahájení léčby nebo nastání nějaké nepříznivé události. Jestliže je end-point chápán jako smrt pacienta, výslednými daty jsou doslova časy přežití. Pokud end-point neznamená smrt, můžeme nakonec dostat data podobného typu, například úleva od bolesti či znovu objevení symptomů (relaps). V tomto případě jsou pozorování často označována za data typu „čas do události“ (time to event). Metodologie analýzy přežívání může být použita i v jiných oblastech – čas přežití zvířat při experimentech, celkový čas jedince potřebný ke splnění určitého úkolu v psychologických studiích, skladovací doba semen rostlin v semenných bankách, životnost elektronických součástek, atd.

Speciální rysy dat v analýze přežívání

Proč pro analýzu těchto dat nemůžeme použít standardní statistické metody? Prvním důvodem je jejich rozdělení, které nemusí být obecně symetrické. Typicky histogram konstruovaný z časů přežití skupiny jedinců má tendenci mít pozitivní šikmost – histogram bude protažený napravo od intervalu, který obsahuje nejvíce pozorování. Tento problém můžeme vyřešit tak, že nejprve data transformujeme (například logaritmováním), abychom dosáhli více symetrického rozdělení. Slabinou takové transformace je ale ne příliš dobrá interpretace.

Hlavním důvodem proč jsou standardní metody pro tato data nevhodné je, že časy přežití bývají často cenzorovány. Časy přežití jsou cenzorovány, pokud end-point nebyl pro daného jedince pozorován, což může být způsobeno tím, že data ze studie jsou analýzovány ve chvíli, kdy jsou někteří účastníci studie stále naživu. Status přežití jedince taktéž nemusí být znám pokud se tento jedinec takříkajíc „ztratil“ (lost to follow-up). Například jedinec se po přijetí do klinické studie přestěhuje do jiné země a nemůže být nadále sledován. Jediná dostupná informace o něm je tedy datum, kdy byl stále živ nebo také datum poslední pravidelné prohlídky lékařem. Dále může být také jedinec ze studie vyřazen nebo sám dobrovolně skončí, také v tomto případě se jedná o cenzorovaná data. Čas přežití může být cenzorovaný i tehdy, pokud jedinec

zemře a příčina smrti nemá nic společného s probíhající léčbou. Příkladem může být účastník klinické studie zabývající se alternativními způsoby léčby rakoviny prostaty, který zahyne během dopravní nehody. Nehodu mohla například zapříčinit nevolnost jakožto vedlejší účinek léčby, kterou pacient podstupoval. V takovém případě pak smrt přímo souvisí s léčbou. Za takových okolností může být doba přežití do smrtelné události ze všech možných příčin nebo doba do smrti z jiných příčin než pro jaké je pacient léčen také předmětem analýzy přežívání.

V každé z těchto situací pacient, který vstoupil do klinické studie v čase t_0, zemře v čase $t_0 + t$. Čas t ale neznáme protože je jedinec buď stále naživu nebo se „ztratil“. Jestliže byl jedinec naposledy zaznamenán jako živý v čase $t_0 + c$, pak čas c nazýváme cenzorovaný čas přežití. Toto cenzorování se objevu po tom, co se jedinec připojil ke studii, což je napravo od naposledy známé doby přežití a tudíž je označováno jako cenzorování zprava.

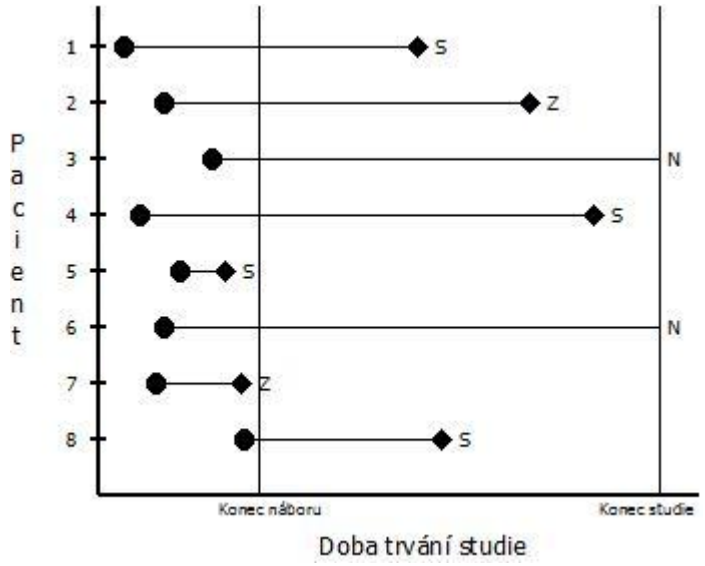
Další druh cenzorování je cenzorování zleva, které nastává v případě, kdy je skutečný čas přežití jedince menší než pozorovaný. Pro ilustraci tohoto typu cenzurování uvažujme studii, která se zaměřuje na dobu návratu určitého druhu rakoviny po chirurgickém odstranění primárního tumoru. Tři měsíce po operaci, jsou pacienti vyšetřeni, aby se zjistilo, zda se rakovina vrátila. V této chvíli může být u některých pacientů nemoc znovu objevena. Pro tyto pacienty je skutečná doba do znovuobjevení rakoviny menší než tři měsíce a tudíž je cenzorována zleva. Toto cenzorování se objevuje mnohem méně než cenzorování zprava.

Dalším typem je pak intervalové cenzorování. Zde je známa informace o události, která u jedince nastala a objevila se v určitém časovém intervalu. Uvažujme nyní stejnou studii jako v přechozím příkladu s cenzorováním zleva. Jestliže se při vyšetření pacienta v době tří měsíců od provedené operace neprokáže přítomnost rakoviny, ale následně je nalezena při vyšetření šest měsíců po operaci, skutečný čas znovuobjevení rakoviny je mezi třemi a šesti měsíci. Tato doba je pak intervalově cenzorována.

V typické studii nejsou všichni pacienti přijímání ve stejný čas, ale jejich počet se zvyšuje po dobu několika měsíců nebo dokonce let. Po přijetí jsou pacienti sledováni do doby než zemřou, nebo po dobu trvání studie, na jejímž konci se data analyzují. Nicméně skutečné časy přežití budou zaznamenány jen pro určitý počet pacientů – po

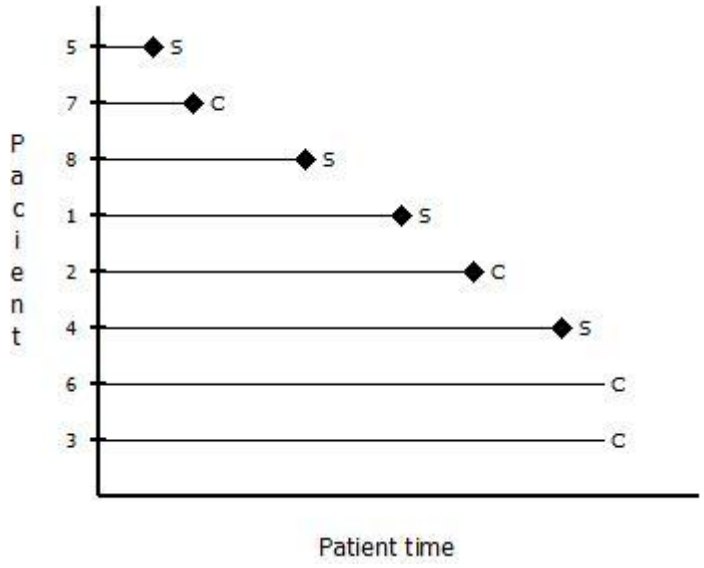
přijetí do studie se někteří mohou ztratit z dosahu, zatímco jiný budou na konci studie stále naživu. Časový úsek po který je pacient účastníkem studie se pak nazývá „doba studie“ (study time).

Obrázek č. (1) ilustruje dobu studie 8 jedinců, kteří jsou účastníky klinické studie. Čas vstupu do studie je označek tečkou. Pacienti 1, 4, 5 a 8 zemřeli (S) během průběhu studie, jedinci 2 a 7 se „ztratili“ (Z) a 3 a 6 byli na konci studie stále naživu (N).



Obrázek č. 1

Klinická studie začíná v časet_0, kdy jsou do studie přizván dostatečný počet pacientů. Na obrázku č. (2) jsou zobrazeny odpovídající časy přežití jednotlivých pacientů - časy přežití jsou uspořádany sestupně podle jejich délky. Doba, kterou pacient ve studii stráví měřená od „patient’s time origin“ je často označována jako „patient time“. Doba od počátku až do smrti pacienta (S) je potom dobou přežití a je zaznamenána pro pacienty 1, 4, 5 a 8. Doba přežití ostatních pacientů je pak cenzorována zprava (C).



Obrázek č. 2

V praxi budou sledovaná data následující – datum vstupu jedince do klinické studie a datum, kdy pacient zemře nebo kdyb byl naposledy shledán živým. Poté je možno spočítat doby přežití ve dnech, týdnech nebo měsících podle toho, co se jeví jako nejvhodnější. Spousta počítačových programů využívající balíčky pro analýzu přežívání má nástroje, které umí pracovat se vstupními daty ve formě datumů. [2, strana 1-12]

Funkce hazardu a přežití (Survivor function and hazard function)

V rámci analýzy přežívání nás zajímají především dvě funkce – funkce přežití a funkce hazardu.

Skutečný čas přežití osoby t, může být považován za hodnotu proměnné T, která může nabývat pouze kladných hodnot. Různé hodnoty, kterých proměnná T nabývá, mají rozdělení pravděpodobnosti a T nazýváme náhodnou proměnnou spojenou s časem přežití. Nyní předpokládejme, že náhodná proměnná T má rozdělení pravděpodobnosti s funkcí hustoty $f(t)$. Distribuční funkce proměnné T je dána jako

$$F(T) = P(T < t) = \int_0^t f(u)\,du,$$

a představuje pravděpodobnost, že doba přežití je menší než nějaká hodnota t.

Funkce přežití $S(t)$ je definovaná jako pravděpodobnost, že doba přežití je větší nebo rovna t, tedy

$$S(t) = P(T \geq t) = 1 - F(t).$$

Funkce přežití tedy může být použita k reprezentaci pravděpodobnosti, kdy osoba přežije od počátku až do nějakého času většího než t.

Funkce hazardu se používá k výjadření rizika (nebo hazardu) smrti v čase t a získá se z podmíněné pravděpodobnosti, že jedinec zemře v čase t za podmínky, že do tohoto času přežil. Abychom funkci hazardu formálně definovaly, uvažujme pravděpodobnost, že náhodná proměnná související s dobou přežití osoby, leží mezi t a $t+\delta t$, za podmínky, že T je větší nebo rovno t, psáno

$$P(t \leq T < t+\delta t \,|\, T \geq t).$$

Funkce hazardu $h(t)$ je limitou, jelikož se δt blíží nule, tedy

$$h(t) = \lim_{\delta t \to 0} \left\{ \frac{P(t \leq T < t+\delta t \,|\, T \geq t)}{\delta t} \right\}.$$

Funkci $h(t)$ také nazýváme mírou rizika (hazard rate), okamžitou mírou úmrtnosti (instantaneous death rate), mírou intenzity (intesity rate) nebo silou úmrtnosti (force of mortality).

$h(t)\delta t$ je přibližně pravděpodobnost, že jedinec zemře v intervalu $(t, t+\delta t)$ za předpokladu, že se dožil času t. Například, pokud je čas přežití měřen ve dnech, pak je $h(t)$ přibližně pravděpodobnost, že osoba, která je naživu v čase t zemře následující den. Z tohoto důvodu je často funkce hazardu jednoduše interpretována jako riziko smrti v čase t.

Z definice funkce hazardu můžeme získat užitečné vztahy mezi funkcí hazardu a funkcí přežití. Z teorie pravděpodobnosti víme, že pravděpodobnost jevu A za podmínky nastání jevu B je dána jako

$$P(A \mid B) = \frac{P(AB)}{P(B)},$$

kde $P(AB)$ je pravděpodobnost, že jevy A a B nastanou zároveň. S využitím tohoto můžeme psát podmíněnou pravděpodobnost v definici funkce hazardu jako

$$\frac{P(t \le T < t+\delta t)}{P(T \ge t)},$$

což je rovno

$$\frac{F(t+\delta t) - F(t)}{S(t)},$$

kde $F(t)$ je distribuční funkce T. Potom

$$h(t) = \lim_{\delta t \to 0} \left\{ \frac{F(t+\delta t) - F(t)}{S(t)} \right\} \frac{1}{S(t)}.$$

Nyní je limita

$$\lim_{\delta t \to 0} \left\{ \frac{F(t+\delta t) - F(t)}{S(t)} \right\}$$

Definicí derivace $F(t)$ podle t, což je $f(t)$, tedy

$$h(t) = \frac{f(t)}{S(t)}.$$

Potom

$$h(t) = -\frac{d}{dt}\{\log S(t)\},$$

a

$$S(t) = \exp\{-H(t)\},$$

kde

$$H(t) = \int_0^t h(u)\,du.$$

Funkce $H(t)$ je v analýze přežívání široce používána a je nazýváme ji sjednocený nebo kumulativní hazard a lze jej získat z funkce přežití jako

$$H(t) = -\log S(t).$$

V analýze přežívání se funkce přežití a funkce hazardu odhaduje z napozorovaných časů přežití.

Coxův regresní model (General proportional hazards model)

Základním modelem pro analýzu přežívání je proportional hazards model. Tento model byl navržen D. R. Coxem (1972) a je také znám jako Coxův regresní model. Navzdory tomu, že je tento model založen na přepokladu o úměrnosti hazardu (proportional hazards), nepředpokládáme žádné konkrétní pravděpodobnostní rozdělení časů přežití, model se pak často označuje jako semi-parametrický.

Předpokládejme, že jsou pacienti náhodně rozděleni do skupin přijímajíchc buďto standardní způsobe léčby nebo nový způsob léčby a nechť $h_S(t)$ a $h_N(t)$ představují rizika úmrtí v čase t pro pacienty se standardní, respektive novou léčbou. Podle jednoduchého modelu pro data přežití dvou skupin pacientů, je hazard v čase t

pacienta na nové léčbě je úměrný hazardu pacienta na standardní léčbě ve stejném čase. Tento model může být vyjádřen ve formě

$$h_N(t) = \psi h_S(t),$$

pro každou kladnou hodnotu t, kde ψ je konstanta. Implikací tohoto předpokladu je, že se odpovídající skutečné funkce přežití pro pacienty na nové a standardní léčbě neprotnou.

Hodnota ψ je poměr rizika smrti v jakémkoliv čase pro jedince podstupující novou léčbu a pro jedince na standardní léčbě a tak je ψ známé jako relativní hazard nebo poměr rizika (hazard ratio). Jestliže je $\psi < 1$, pak riziko smrti v čase t je menší pro osobu s novou léčbou vzhledem k riziku smrti osoby se standardní léčbou. Nový způsob léčby je pak vyplepšením standardní procedury. Na druhou stranu, pokud je $\psi > 1$, pak je standardní léčba lepší.

Alternativní způsob vyjádření modelu vede k vytvoření modelu, který může být mnohem snadněji zobecněn. Předpokládejme, že máme k dispozici data o n osobách a označme funkci hazardu pro i-té osoby jako $h_i(t), i = 1, 2, ..., n$. $h_0(t)$ značíme funkci hazardu pro osoby se standardní léčbou a $\psi h_0(t)$ pro osoby s novým způsobem léčby. Relativní hazard ψ nemůže být záporný, takže je výhodné psát $\psi = \exp(\beta)$. Parametr β je potom logaritmus poměru rizika, tedy $\beta = \log \psi$ a jakákoli hodnota β v intervalu $(-\infty, \infty)$ vyústí v kladnou hodnotu ψ. Poznamenejme, že kladné hodnoty β získáme v případě, kdy je poměr rizika ψ větší než jedna, tedy pokud je nová léčba horší než standardní.

Nechť je X indikační proměnná, která nabývá hodnoty nula, podstupuje-li osoba standardní léčbu a jedna, jestliže je na danou osobu aplikovaná nová léčba. Jestliže je x_i hodnotou X pro i-tou osobu ve studii, $i = 1, 2, ..., n$, pak pro tuto osobu píšeme funkci hazardu jako

$$h_i(t) = e^{\beta x_i} h_0(t).$$

kde $x_i = 1$ pokud osoba podstupuje novou léčbu a $x_i = 0$ jinak, což je proportional hazards model pro porovnání dvou skupin podstupující různé způsoby léčby.

Validace a Cross-Validace

Testovací chyba je průměrná chyba vznikající v důsledku použití statistické metody ke stanovení odhadů vysvětlované proměnné na nových pozorováních – je to tedy míra, která nebyla použitá při trénování metody. Použití konkrétní metody na daném data setu je odůvodněno tím, že ve výsledku bude testovací chyba malá. Testovací chybu lze snadno spočítat v případě, kdy je k dispozici testovací sada dat, což ale obvykle nebývá. Naproti tomu trénovací chybu lze snad spočítat tak, že metodu aplikujeme na data použitá při jejím trénování. Nicméně trénovací chyba je často velmi rozdílná od testovací chyby, ve výsledku pak trénovací chyba může velmi drasticky podurčit testovací chybu.

V případě, kdy nemáme k dispozici velkou množinu dat, na které bychom mohli přímo stanovit testovací chybu, jsou k dispozici metody užívající ke stanovení této chyby dostupná trénovací data. Některé metody užívají matematických úprav trénovací chyby k odhadu testovací chyby. Do takové kategorie lze zařadit metody popsané v této práci – subsetselection, penalizační (shrinkage)methods (lasso, ridgeregression) a dále pak například metody redukce dimenze. Nyní uvažujme třídu metod, které odhadují testovací chybu tím, že si „odloží“ podmnožinu trénovacích pozorování a pak na tato pozorování aplikuje danou metodu.

Pro jednoduchost předpokládejme, že chceme provést regresi s kvantitativní vysvětlovanou proměnnou. Hlavní myšlenky těchto metod však zůstávají stejně nehledě na povahu vysvětlované proměnné (kvantitativní x kvalitativní).

TheValidation Set Approach

Předpokládejme, že chceme odhadnout testovací chybu spojenou s aplikováním statistické metody na určitou množinu pozorování. Pro tento případ je použitíValidation Set Approach (VSA) velmi jednoduché. VSA spočívá v náhodném rozdělení množiny pozorování do dvou částí – na trénovací sadu a validační sadu (tzv. „hold-out set). Model aplikujeme na trénovací sadu a následně jej pak použijeme na pozorování ve validační sadě k určení odhadů předpovědí. Výsledná chyba na validační sadě (většinou používáme MSE – střední čtvercovou chybu, viz kapitola Míry vhodnosti modelu) je odhadem chyby testovací.

VSA je konceptě velmi jednoduchý a snadno implementovatelný přístup, nicméně má dvě potencionální nevýhody:

1. validační odhad testovací chyby může být velmi proměnlivý v závislosti na tom, jaká pozorování jsou v trénovací sadě a jaká ve validační sadě.
2. pro aplikaci modelu se používá pouze podmnožina všech pozorování (ta, která jsou zahrnuta do trénovací sady). Statistické metody vycházejí hůře při použití na menším počtu pozorování z čehož vyplývá, že chyba na validační sadě může mít tendenci přeurčit testovací chybu pro model se všemi pozorováními.

V následujících podkapitolách bude popsána cross-validace, vylepšení VSA, které řeší její výše zmíněné nevýhody.

Obrázek č. 3 - Validation Set Approach [3, strana 177]

Leave-one-out Cross-Validation (LOOCV)

LOOCV úzce souvisí s VSA popsané výše, nicméně tímto přístupem se snažíme odstranit nevýhody VSA.

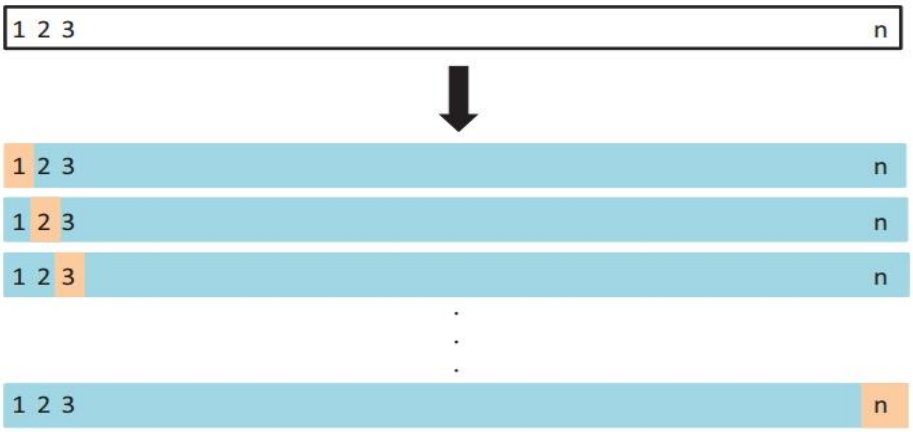
Stejně jako u VSA i v případě LOOCV rozdělíme množinu pozorování do dvou částí. Avšak v nyní nevytvoříme dvě podmnožiny o stejném počtu pozorování, nýbrž vyjmeme jedno pozorování (x_1, y_1), které bude představovat validační sadu, zatímco zbylá pozorování $\{(x_2, y_2), ..., (x_n, y_n)\}$ tvoří trénovací sadu. Statistická metoda se aplikuje na $n-1$trénovacích pozorování a předpověď $\hat{y}_1$ je vytvořena na základě vyloučeného pozorování, použitím hodnoty x_1. Jelikož bylo pozorování (x_1, y_1) odloženo, tak $MSE_1 = (y_1 - \hat{y}_1)^2$ je přibližný nevychýlený odhad testovací chyby.

Postup opakujeme vyjmutím pozorování (x_2, y_2), aplikací statistické metody na $n-1$ pozorování $\{(x_1, y_1), (x_3, y_3), ..., (x_n, y_n)\}$ a spočítáním $MSE_2 = (y_2 - \hat{y}_2)^2$. Opakováním

tohoto postupu $n-$krát získáme n čtvercových chyb $MSE_1,...,MSE_n$. LOOCV odhad testovací chyby MSE je průměrem n odhadů testovacích chyb

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i .$$

LOOCV má několik velkých výhod oproti VSA. Zaprvé má mnohem menší bias. Při LOOCV se statistická metoda opakovaně aplikuje na $n-1$ pozorování, což je pouze o jedno pozorování méně, než obsahuje množina všech pozorování. V tomto je rozdíl oproti VSA, kde je většinou velikost trénovací sady rovna polovině všech pozorování. Při LOOCV nedochází tak moc k přeurčení testovací chyby jako u VSA. Za druhé, v případě opakovaného použití VSA dostáváme různé výsledky v důsledku náhodnosti rozdělení množiny pozorování na trénovací a validační sadu. Naproti tomu opakovaným použitím LOOCV dostane vždy stejné výsledky, jelikož do rozdělení na trénovací a validační sadu nevstupuje náhoda.



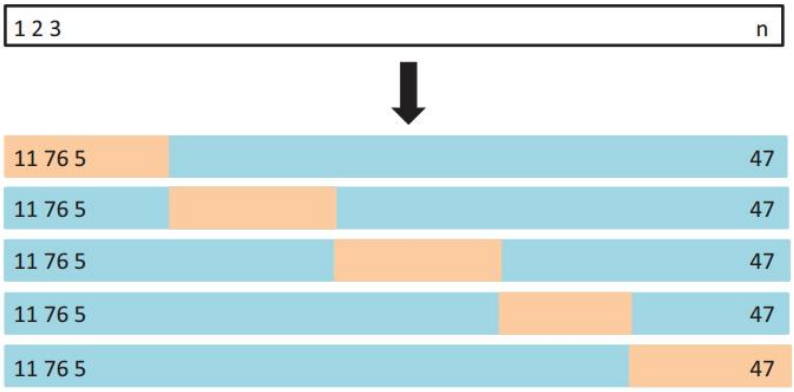
Obrázek č. 4 - LOOCV [3, strana 179]

K-Fold Cross-Validation

Alternativou k LOOCV je k-fold CV. Tento přístup spočívá v náhodném rozdělení všech pozorování do k skupin (folds) o přibližně stejné velikosti. První „fold“ je považován za validační sadu a statistická metoda je aplikována na zbývajících $k-1$ skupin. Střední čtvercová chyba MSE_1 se spočítá na validační sadě. Procedura se následně opakuje k-krát, v každém opakování se za validační sadu stanoví jiná množina pozorování. Výsledkem je k odhadů testovací chyby $MSE_1, MSE_2,..., MSE_k$, k-fold CV odhad se spočítá zprůměrováním těchto hodnot jako

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i .$$

LOOCV je speciálním případem k-fold CV, ve kterém je $k = n$. V praxi se často používá k-fold CV pro $k = 5$ nebo $k = 10$. V čem je použití $k = 5$ nebo $k = 10$ lepší než $k = n$? Nejočividnější výhoda je samozřejmě výpočetní – při LOOCV je statistická metoda použita n-krát, což může být výpočetně velmi náročné (kromě případu použití lineární modelu a metody nejmenších čtverců), nicméně cross-validace je velmi obecný přístup a lze jej použít na téměř každou statistickou metodu. Některé statistické metody jsou při aplikaci výpočetně náročné, takže použití LOOCV může způsobit výpočetní problémy zvláště v případech, kdy je n opravdu velké. Naproti tomu použití 10-fold CV vyžaduje aplikovat metodu pouze 10 krát, což je mnohem snáze proveditelné. [3, strana 176-182]



Obrázek č. 5 - K-fold CV [3, strana 181]

Míry vhodnosti modelu

Celkovou variabilitu v datech měříme pomocí celkového součtu čtverců

$$SS_T = \sum_i \left(y_i - \bar{y}\right)^2 .$$

Celkový součet čtverců SS_T můžeme rozložit na dvě části

$$SS_M = \sum_i \left(\hat{y}_i - \bar{y}\right)^2$$

a

$$SS_e = \sum_i \left(y_i - \hat{y}_i\right)^2 .$$

SS_e nazýváme reziduální součet čtverců. Čím je jeho hodnota menší, tím lepší je odhadnutý model.

Jako R^2 označujeme takzvaný index determinace.

$$R^2 = \frac{SS_M}{SS_T} = 1 - \frac{SS_e}{SS_T} .$$

Index determinace nabývá hodnot z intervalu $\langle 0,1 \rangle$. Čím blíže je jeho hodnota k 1, tím lepší je pak odhadnutý model. Není ovšem možné rozhodovat o vhodnosti modelu pouze na základě indexu determinace – například v ekonometrii mají modely často hodnotu blízkou k 1, zatímco v jiných oblastech mohou být modely vhodné i přestože jejich index determinace je relativně malý. Pokud aplikujeme různé modely na stejná data, může nám index determinace pomoci rozhodnout, který model zvolit. Jelikož se ale hodnota indexu determinace zvýší (nebo nezmění) v případě přidání další proměnné do modelu, častěji používám upravený index determinace

$$R^2_{adj} = 1 - \frac{n-1}{n-p}\left(1 - R^2\right).$$

[1,strana 4-6]

Nápad penalizovat věrohodnostní funkcni, takovým způsobem, aby lépe vycházely jednodušší modely mělo mnoho autorů. Realizací této myšlenky je porovnávat modely pomocí míry

$$D_C = D - \alpha q \phi ,$$

kde D je odchylka, q je počet parametrů v modelu a ϕ je disperzní parametr. Tuto míru nazýváme Akaikeho informačním kritériem (AIC) a je často používaná k výběru modelu – vybereme takový model, který má AIC nejmenší. [1, strana 46]

Schwarz (1978) odvodil Bayesovo informační kritérium (BIC) jako dva krát logaritmus Bayesova faktoru. Pro model M_j parametrizovaný m_j-rozměrným vektorem θ_j,

$$BIC = -2\left\{\ell_j\left(\hat{\theta}_j\right) - \ell_0\left(\hat{\theta}_0\right)\right\} + \left(m_j - m_0\right)\log\left(n\right),$$

kde $\ell_j\left(\hat{\theta}_j\right)$ a $\ell_0\left(\hat{\theta}_0\right)$ jsou věrohodnostní funkce (z metody maximální věrohodnosti) M_j a referenčního modelu M_0, jehož parametr má rozměr m_0 a n je velikost výběru. Jestliže je $BIC < 0$, pak preferujeme M_j před M_0, respektive čím víc je BIC záporný, tím více preferuje M_j. BIC bylo široce užíváno jako kritérium pro výběr vhodného statistického modelu. [6, strana 1]

Mallowsovo Cp (Mallows 1973) je velmi silná technika pro výběr regresního modelu. Cp statistiku definujeme jako

$$Cp = \frac{RSS_p}{\hat{\sigma}^2 - n + 2p},$$

kde RSS_p je reziduální součet čtverců pro podmodel P, p je dimenze tohoto podmodelu, n je počet pozorování a $\hat{\sigma}^2$ je odhad rozptylu chyb, který se obvykle počítá z plného modelu. Jestliže platí podmodel P, pak Cp bude blízké nebo menší než p. [7, strana 2]

V regresi často používáme míru zvanou střední čtvercová chyba (MSE), která je definovaná jako

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}\left(x_i\right)\right)^2 ,$$

kde $\hat{f}\left(x_i\right)$ je předpověď $\hat{f}$ pro i-té pozorování. MSE bude nízká v případě, že je předpověď blízko skutečné hodnotě a vysoká pokud se pro některá z pozorování bude skutečná hodnota a předpověď podstatně lišit. [3, strana 29]

SubsetSelection

Best Subset Selection

Při použití best subset selection aplikujeme metodu nejmenších čtverců na každou kombinaci p regresorů. To znamená, že bereme v úvahu všech p modelů obsahujících právě jeden regresor, všech $\binom{p}{2} = p(p-1)/2$ modelů, které obsahují právě dva regresory, a tak dále. Následně z těchto modelů vybereme ten „nejlepší“.

Problém výběru jednoho nejlepšího modelu z 2^p možností sestavených pomocí best subset selection není triviální – v následujícím algoritmu je postup rozložen do dvou částí:

Algoritmus – Best subsetselection [3, strana 205]

1. Nechť M_0 značí nulový model, který neobsahuje žádný regresor. Tento model jednoduše predikuje výběrový průměr pro každé pozorování
2. Pro $k = 1,2,\dots,p$
   a) vytvoříme všech $\binom{p}{k}$ modelů, které obsahují právě k regresorů
   b) vezmeme nejlepší z těchto $\binom{p}{k}$ modelů a označíme M_k. Za nejlepší model je zde považován ten s nejmenším RSS nebo ekvivalentně s největším R^2
3. Vybereme nejlepší model z $M_0,\dots,M_p$ na základě cross-validované chyby předpovědi, C_p (AIC), BIC nebo upraveného R^2.

V druhém kroku algoritmu nacházíme nejlepší model (na trénovací sadě dat) pro každou podmonižnu tak, abychom zredukovali problém výběru z 2^p možných modelů na výběr jednoho z $p+1$ modelů.

Nyní tedy vybíráme nejlepší model z pouhých $p+1$ možností. Výběr nejlepšího modelu ale musíme provést s obezřetností, neboť RSS $p+1$ modelů monotonně klesá a R^2 monotonně roste s každým dalším regresorem přidaným do modelu. Pokud bychom tedy použili tyto statistiky pro rozhodování o nejlepším modelu, vždy bychom vybrali model, který obsahuje všechny proměnné. Problémem je to, že malé RSS nebo naopak velké R^2 představuje model s malou trénovací chybou, zatímco my chceme vybrat model s malou testovací chybou. (příklad – trénovací chyba bývá obvykle menší než testovací, ale malá trénovací chyba v žádném případě negarantuje malou testovací chybu.

Ačkoliv zde byla zmíněna best subset selection pouze pro metodu nejmenších čtverců (lineární regresi), stejný postup lze aplikovat i na jiné typy modelů, například pro logistickou regresi. V případě logistické regrese namísto RSS pro porovnání jednotlivých modelů používáme devianci, míru, která hraje roli RSS v širší třídě modelů. Deviance je definována jako $-2 * max\ log\ L$ a čím je menší, tím je model lepší.

I když je best subset selection jednoduchý a koncepčně atraktivní přístup, trpí výpočetním omezením. Počet všech možných modelů, které v rámci volby nejlepšího modelu musíme uvažovat, strmě roste se zvětšujícím se p - obecně existuje 2^p možných modelů zahrnujících podmnožiny p regresorů. Čili pokud máme $p = 10$ dostáváme 1024 uvažovaných modelů, pro $p = 20$ je to 1 048 576 možností. Best subset selection se pak stává výpočetně nemožná pro hodnoty $p \geq 40$, dokonce i pro extrémně rychlé moderní počítače. Existují výpočetní „zkratky" – takzvané branch-and-bound techniky [11] – pro eliminaci některých možností, ale i ty mají své omezení pro rostoucí se p. Tyto techniky je také možné použít jen v lineární regresi.

Stepwise selection

Z výpočetních důvodů nemůže být best subset selection aplikována na případy s velmi velkým p, jelikož může trpět statistickými problémy. Čím větší je prohledávaný prostor, tím je vyšší pravděpodobnost, že najdeme model, který na trénovacích datech bude vypadat velmi dobře, nicméně nemusí mít žádnou předpovědní sílu budoucích dat. Navíc velký prostor může vést k overfittingu (přeurčení modelu) a velkému rozptylu odhadů koeficientů. Pro oba tyto důvody jsou stepwise metody, které zkoumají daleko více omezenou množinu modelů, lákavou alternativou k best subset selection.

Forward Stepwise Selection

Tato metoda je výpočetně efektivní alternativa k best subset selection. Zatímco procedura best subset selection uvažuje všech 2^p modelů zahrnujících podmnožiny p regresorů, forward stepwise selection uvažuje mnohem menší množinu modelů. Samotná procedura pak začíná s modelem neobsahující žádný regresor, postupně pak po jednom regresory přidáváme až do chvíle, kdy jsou v modelu všechny proměnné. V každém kroku je ale přidána pouze ta proměnná, která nejvíce přispěje k vylepšení modelu. Formálněji tento postup shrnuje následující algoritmus [3, strana 207]:

1. Nechť M_0 značí model značí nulový model, který neobsahuje žádný regresor.
2. Pro $k = 0, \dots, p-1$
   a) uvažujme všech $p-k$ modelů, které rozšíří počet regresorů v M_k o jeden další regresor
   b) vybereme nejlepší z těchto $p-k$ modelů a označíme jej M_{k+1}. Za nejlepší model je zde považován ten s nejmenším RSS nebo ekvivalentně s největším R^2.
3. Vybereme nejlepší model z $M_0, \dots, M_p$ na základě cross-validované chyby předpovědi, C_p (AIC), BIC nebo upraveného R^2.

Na rozdíl od best subset selection, která zahrnovala 2^p modelů, forward stepwise selection zahrnuje jeden nulový s $p-k$ modely v k-té iteraci pro $k = 0, \dots, p-1$. To činí dohromady $1 + \sum_{k=0}^{p-1}(p-k) = 1 + p(p+1)/2$ modelů. A v tom je podstatný rozdíl – pro $p = 25$, best subset selection vyžaduje vytvoření 33 554 432 modelů, zatímco forward stepwise vyžaduje pouhých 326 modelů. (pozn.: i když forward stepwise selection uvažuje $\frac{p(p+1)}{2} + 1$ modelů, provádí tzv. guided search přes prostor modelů a tedy uvažovaný efektivní prostor modelů obsahuje podstatně více než $\frac{p(p+1)}{2} + 1$ modelů.

V algoritmu v kroku 2b) musíme vybrat nejlepší model z $p-k$ modelů, které rozšíří M_k o jeden další regresor. Výběr můžeme jednoduše provést na základě nejmenšího RSS nebo největšího R^2. V kroku 3 ale musíme nalézt nejlepší model z množiny modelů, které mají odlišný počet regresů, což už je podstatně náročnější. Metody, na základě kterých volíme nejlepší model budou popsané dále.

Výpočetní výhody forward stepwise selection oproti best subset selection jsou jasné. Ačkoliv si forward stepwise selection vede v praxi velmi dobře, není garantováno nalezení nejlepšího možného modelu ze všech 2^p modelů obsahujících podmnožiny p regresorů. Pro příklad předpokládejme, že pro daný dataset s $p = 3$ regresory, nejlepší možný model s jednou proměnnou obsahuje X_1 , a nejlepší model se dvěma proměnnými zahrnuje X_2 a X_3. Pak forward stepwise selection selže při hledání nejlepšího modelu se dvěma proměnnými, jelikož M_1 bude obsahovat X_1 a tedy M_2 musí také obsahovat X_1 s jednou další proměnnou.

Forward stepwise selection může být aplikován i pro velké dimenze, kde je $n < p$, ale v tomto případě je možné pouze zkonstruovat podmodely $M_0, \dots, M_{n-1}$, jelikož se na každý podmodel použije metoda nejmenších čtverců a ta nebude dávat unikátní řešení pro $p \geq n$.

Backward Stepwise Selection

Stejně jako forward stepwise selection je tato metoda efektivní alternativou k best subset selection, ale narozdíl od forward stepwise selection začínáme s plným modelem obsahujícím všech p regresorů a následně je iteračně vyjímán z modelu po jednom vždy nejméně užitečný regresor. Detailně je postup popsán algoritmem [3, strana 209]:

1. Nechť M_p značí plný model, obsahující všechny regresory
2. Pro $k = p, p-1, \dots, 1$
   a) uvažujme všech k modelů, které obsahují všechny krom jednoho regresoru v M_k, z celkových $k-1$ regresorů
   b) vybereme nejlepší z těchto k a označíme M_{k-1}. Za nejlepší model je zde považován ten s nejmenším RSS nebo ekvivalentně s největším R^2
3. Vybereme nejlepší model z $M_0, \dots, M_p$ na základě cross-validované chyby předpovědi, C_p (AIC), BIC nebo upraveného R^2.

Stejně jako forward stepwise selection, tak i backward stepwise selection prohledává pouze $1 + p(p+1)/2$ modelů a může být aplikován v případě kdy je p příliš velké pro použití best subset selection. (pozn.: Stejně jako forward stepwise selection, backward stepwise selection provádí tzv. guided search přes prostor modelů a efektivně uvažuje podstatně více než $1 + p(p+1)/2$ modelů). A taktéž backward stepwise selection negarantuje nalezení nejlepšího modelu obsahujícího podmnožinu p regresorů.

Použití backward stepwise selection vyžaduje, aby byl počet pozorování n větší než počet proměnných p (aby mohl být sestaven plný model). Naproti tomu může být forward stepwise selection použito i v případě, kdy je $n < p$ a je tedy jedinou použitelnou subset metodou v případě, že je p velmi velké.

Hybridnípřístupy

Best subset selection, forward stepwise selection a backward stewise selection přístupy obecně dávají podobné ale ne identické modely. Jako další alternativu je

možný použít hybridní verze forward a backward stepwise selection, v kterých jsou proměnné postupně přidávány do modelu analogicky jako při forward stepwise selection. Na druhou stranu ale po přidání proměnné může dojít k vyloučení proměnné, která už nadále model nevylepšuje. Takový přístup se snaží blízce napodobnit best subset selection při zachování výpočetních výhod metod forward a backward stepwise selection.

Výběr optimálního modelu

Best subset selection, forward a backward stepwise selection ústí ve vytvoření sady modelů, kde každý z nich obsahuje podmnožinu p regresorů. Abychom tyto metody mohli použít, musíme najít způsob, jakým posoudíme, který z modelů je nejlepší. Jak už bylo zmíněno dříve, model obsahující všechny regresory bude mít vždy nejmenší RSS a největší R^2, jelikož jsou tyto statistiky spojeny s trénovací chybou. Naproti tomu chceme vybrat model, který bude mít nejmenší testovací chybu. Trénovací chyba může být velmi špatným odhadem testovací chyby, tudíž RSS a R^2 nejsou vhodné pro výběr nejlepšího modelu z množiny modelů s různými počty regresorů. Abychom mohli vybrat nejlepší model s ohledem na testovací chybu, musíme tuto chybu nejprve odhadnout. K tomu je možné použít jeden z následujících přístupů:

1. testovací chybu můžeme odhadnout nepřímo tím, že provedeme úpravu trénovací chyby vzhledem k Biasu kvůli „přeurčení modelu“,
2. testovací chybu můžeme odhadnout přímo pomocí validace nebo cross-validace.

Shrinkage Methods

Metody typu subset selection popsané v předchozí kapitole se snaží najít nejlepší model porovnáváním modelů, do kterých zahrnujeme jen některé (podmnožinu) ze všech možných vysvětlujících proměnných (v případě p>n je to přímo nutnost) . Alternativně můžeme brát v úvahu model obsahující všech p regresorů s použitím technik, které regularizují nebo ekvivalentně zmenšují odhady parametrů směrem k nule. Nemusí být hned očividné, proč by tato regularizace měla model vylepšit, ale později se ukáže, že zmenšování odhadů parametrů může velmi významně snížit jejich rozptyl. Dvě nejznámější metody pro zmenšování regresních koeficientů jsou hřebenová regrese (ridgeregression) a lasso regrese.

Hřebenová regrese

Připomeňme, že použitím metody nejmenších čtverců získáme odhady $\beta_0, \beta_1, \dots, \beta_p$ minimalizací výrazu

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2. \qquad (1)$$

Hřebenová regrese je metodě nejmenších čtverců velmi podobná až na to, že koeficienty jsou odhadovány pomocí minimalizace jiného výrazu. Konkrétně koeficienty hřebenové regrese jsou odhady β^R, které minimalizují

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = RSS + \lambda \sum_{j=1}^{p} \beta_j^2, \qquad (2)$$

kde $\lambda \geq 0$ je tzv. penalizační parametr. Tento parametr se volí zvlášť.Stejně tak jako u metody nejmenších čtverců, hřebenová regrese hledá takové odhady koeficientů, které nejlépe odpovídají datům, tím že zmenšuje RSS. Nicméně druhý výraz $\lambda \sum_{j=1}^{p} \beta_j^2$ se nazývá penalizace a je malý v případě, že jsou koeficienty $\beta_0, \beta_1, \dots, \beta_p$ blízké nule a tudíž má efekt „zmenšování“ odhadů β_j směrem k nule. Penalizační parametrλ slouží ke kontrole relativního dopadu těchto dvou výrazů na odhady regresních koeficientů. V případě, že $\lambda = 0$ nemá penalizační člen žádný efekt a z hřebenové regrese získáme stejně odhady jako metodou nejmenších čtverců. Jakmile jde $\lambda \to \infty$, dopad penalizačního členu roste a odhady parametrů hřebenové regrese se blíží nule. Narozdíl od metody nejmenších čtverců, kterou získáme jedinou kombinaci odhadů

koeficientů, nám hřebenová regrese dává různé odhady parametrů pro různé hodnoty λ. Správná volba hodnoty λ je velmi důležitá.

Povšimněme si, že penalizace je aplikována na koeficienty $\beta_1, \dots, \beta_p$, ale ne na absolutní člen β_0. Je to dáno tím, že chceme zmenšit odhadovanou spojitost vysvětlujících proměnných k vysvětlované proměnné, ale nechceme zmenšit absolutní člen, který je střední hodnotou v případě, že $x_{i1} = x_{i2} = \cdots = x_{ip} = 0$. Pokud vezmeme v úvahu, že vysvětlující proměnné (sloupce matice X) budou mít před aplikací hřebenové regrese nulovou střední hodnotu, pak odhadnutý absolutní člen bude ve tvaru $\beta_0 = \bar{y} = \sum_{i=1}^{n} \frac{y_i}{n}$.[3, strana 214-215]

Lasso regrese

Hřebenová regrese má jednu očividnou nevýhodu. Narozdíl od best subset, forward stepwise a backward stepwise selection, které obecně mají za výsledek modely obsahující pouze podmnožinu vysvětlujích proměnných, hřebenová regrese zahrne do výsledného modelu všech p regresorů. Penalizační člen $\lambda \sum \beta_j^2$ zmenšuje všechny koeficienty směrem k nule, ale žádný z nich nakonec nule roven nebude (koeficienty by byly rovny nule v případě, že $\lambda = \infty$). Tento problém se netýká přesnosti odhadů, nicméně pokud je počet proměnných p velký, může se stát výsledný model velmi špatně interpretovatelným. Zvyšováním hodnoty parametru λ dojde ke zmenšování velikosti jednotlivých koeficientů, ale nedojde k vyřazení žádného z nich.

Lasso [8., 9.] je relativně nová alternativa ke hřebenové regresi, která tuto nevýhodu překonává. Koeficienty lasso regrese minimalizují výraz

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| = RSS + \lambda \sum_{j=1}^{p} |\beta_j|. \qquad (3)$$

Pokud porovnáme předpisy lasso regrese a hřebenové regrese, zjistíme, že obě metody mají velmi podobný zápis. Jediným rozdílem je, že penalizační člen β_j^2 u hřebenové regrese je v lasso regresi nahrazen jiným penalizačním členem $|\beta_j|$. Ve statistické hantýrce používá lasso penalizaci ℓ_1 namísto ℓ_2. Norma ℓ_1 vektoru koeficientů β je dáno jako $\|\beta\|_1 = \sum |\beta_j|$.

Stejně jako použití hřebenové regrese, i lasso regrese zmenšuje odhady koeficientů směrem k nule. Nicméně v případě lasso regrese, má penalizace ℓ_1 za následek zmenšení některých koeficientů na hodnotu rovnu nule - za předpokladu dostatečně velkého parametru λ. Použitím lasso regrese se tedy stejně jako u subset selection provede výběr proměnných. Výsledný model je pak mnohem lépe interpretovatelný než je tomu u hřebenové regrese. Říkáme, že výsledkem lasso regrese jsou tzv. „sparse“ modely – tedy modely, které obsahují pouze podmnožinu všech regresorů. I v lasso regresi je velmi důležité vhodně zvolit hodnotu parametru λ, nejčastěji se pro volbu tohoto parametru používá cross-validace, ale někdy hledáme takovou hodnotu, abychom získali stanovený počet nenulových koeficientů.

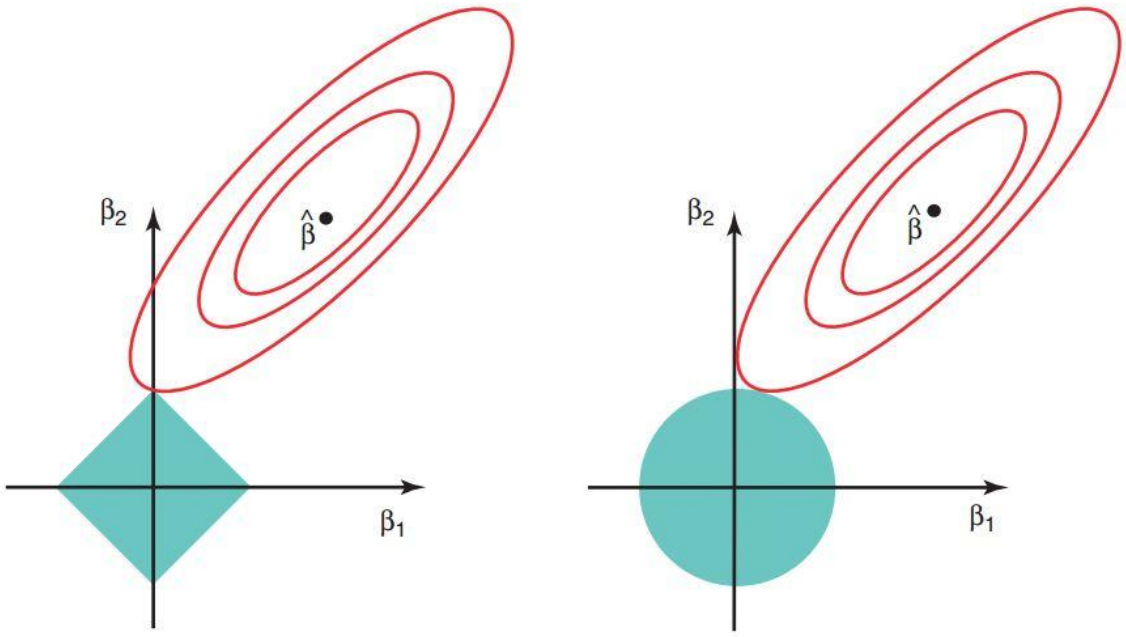
Odhady koeficientů lasso a hřebenové regrese jsou řešením

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \text{za podmínky} \sum_{j=1}^{p} |\beta_j| \leq s \qquad (4)$$

a

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \text{za podmínky} \sum_{j=1}^{p} \beta_j^2 \leq s. \qquad (5)$$

Jinými slovy pro každou hodnotu λ existuje takovés, pro které z rovnic (3) a (4) získáme stejné odhady parametrů lasso regrese. Podobně pak pro každou hodnotu λ existuje s, pro které z rovnic (2) a (5) získáme stejné odhady parametrů hřebenové regrese. V případě, kdy$p = 2$, je potom z rovnice (4) patrné, že odhady koefientů lasso regrese budou mít nejmenší RSS ze všech bodů, které leží uvnitř čtyřúhelníku definovaném $|\beta_1| + |\beta_2| \leq s$. Podobně pak odhady hřebenové regrese budou mít nejmenší RSS ze všech bodů, ležících uvnitř kruhu definovaného $\beta_1^2 + \beta_2^2 \leq s$.



Obrázek č. 6 - [3, strana 222]

O rovnici (4) můžeme uvažovat následovně – když používáme lasso regresi, tak se snažíme najít množinu odhadů koeficientů, které mají ve výsledku nejmenší RSS, za podmínky, že existuje budget s, na kterém zaleží, jak velké $\sum_{j=1}^{p} |\beta_j|$ může být. Pokud je s opravdu velké, potom není budget příliš omezující a tedy i odhady parametrů mohou být velmi velké. Jestliže je s dostatečně velké, aby budget pokryl odhady

metodou nejmenších čtverců, dostaneme z rovnice (4) přesně tyto odhady. Naproti tomu pokud je s malé, potom $\sum_{j=1}^{p}|\beta_j|$ musí být malé, aby neporušilo budget. Stejně tak rovnice (5) ukazuje, že použitím hřebenové regrese opět hledáme takovou množinu odhadů koeficientů, pro které bude RSS co nejmenší, za podmínky, že $\sum_{j=1}^{p}\beta_j^2$ nepřekročí budget s.

Rovnice (4) a (5) odhalují úzkou spojitost mezi lasso regresí, hřebenovou regresí a best subset selection. Uvažujme následující

$$\min_{\beta}\left\{\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \text{„subject to“ } \sum_{j=1}^{p} I(\beta_j \neq 0) \leq s. \qquad (6)$$

Zde je $\sum_{j=1}^{p} I(\beta_j \neq 0)$ indikační proměnnou – nabývá hodnoty 1 pro $\beta_j \neq 0$a 0 jinak. Potom rovnicí (6) najdeme množinu odhadů koeficientů, které budou mít nejmenší RSS za podmínky, že nejvýše s koeficientů může být nenulových, což je velmi podobné best subset selection. Naneštěstí je řešení rovnice (6) výpočetně neproveditelné v případě, kdy je s velké, jelikož je nutné uvažovat všech $\binom{p}{s}$ modelů, obsahující s prediktorů. Tedy lasso regresi a hřebenovou regresi můžeme interpretovat jako výpočetně proveditelné alternativy k best subset selection, které nahrazují neřešitelný tvar budgetu jako v rovnici (6) jinými, které jsou již snadněji řešitelné. Lasso má samozřejmě bližší spojitost k best subset selection, jelikož pouze lasso provádí „feature selection“ pro dostatečně malé s v rovnici (4).

Výběr proměnných pomocí lasso regrese

Čímto, že lasso narozdíl od hřebenové regrese má za ve výsledku odhady koeficientů, které jsou rovny nule? Rovnice (4) a (5) to pomohou osvětlit – obrázek (6) tuto situaci názorně ilustruje. Řešení metodou nejmenších čtverců je označeno $\hat{\beta}$, modrý čtyřúhelník a kruh pak představuje podmínky pro lasso a hřebenovou regresi v rovnicích (4) a (5). Pro dostatečně velké s budou modré oblasti podmínek obsahovat $\hat{\beta}$ a odhady koeficientů lasso i hřebenová regrese budou rovny odhadům metodou nejmenších čtverců (taková hodnota s odpovídá $\lambda = 0$ v rovnicích (1) a (3). Nicméně na obrázku (6) leží odhady metodou nejmenších čtverců mimo modré oblasti a tedy nejsou stejné jako odhady lasso a hřebenové regrese.

Elipsy, které mají střed v $\hat{\beta}$ reprezentují oblastí konstantního RSS. Jinými slovy, všechny body na dané elipse mají stejnou hodnotu RSS. Čím více se elipsy vzdalují od

odhadů koeficientů metodou nejmenších čtverců, tím větší je pak RSS. Rovnice (4) a (5) ukazují, že odhady koeficientů lasso a hřebenové regrese jsou dány prvním bodem, ve kterém se protne elipsa s oblastní podmínky. Jelikož má hřebenová regrese podmínku danou kruhovou oblastí, nebude obvykle tento průsečík ležet na ose, a tedy odhady koeficientů budou výhradně nenulové.Podmínka lasso regrese má tvar čtyřúhelníku a každý roh na jedné z os, tudíž elipsa často protne oblast podmínky právě zde. V takovém případě bude jeden z koeficientů roven nule. Ve vyšších dimenzích modelu může být takových koeficientů, které jsou rovny nule, mnohem více. Na obrázku (6) je průsečík v $\beta_1 = 0$ a tedy výsledný model pak obsahuje pouze β_2.

Na obrázku (6) uvažujeme jednoduchý případ, kdy $p = 2$. Pokud je $p = 3$, potom se oblast podmínky pro hřebenovou regresi stane koulí a pro lasso regresi mnohostěnem. Pro $p > 3$ se pak z koule stane hyperkoule a z mnohostěnu „nadstěn“ (mnohostěn ve vícerozměrném prostoru). Nicméně klíčové myšlenky z obrázku (6) stále platí a lasso regrese vede k feature selection pro $p > 2$ z důvodů oněch ostrých rohů mnohostěnu a „polytope“ (do češtiny se někdy překládá jako nadstěn).

Jednoduchý speciální příklad pro hřebenovou a lasso regresi

Abychom získali lepší představu o chování hřebenové a lasso regrese uvažujme jednoduchý případ, kdy $n = p$ a X je diagonální matice s hodnotami 1 na hlavní diagonále a 0 mimo hlavní diagonálu. Pro další zjednušení předpokládejme, že regrese je bez průniku. Za těchto předpokladů se řešení metodou nejmenších čtverců zjednodušuje na nalezení $\beta_1, \dots, \beta_p$, které minimalizují

$$\sum_{j=1}^{p} (y_j - \beta_j)^2. \tag{7}$$

V takovém případě je pak řešení dáno jako

$$\hat{\beta}_j = y_j .$$

Hřebenovou regresí pak hledáme takové $\beta_1, \dots, \beta_p$, které minimalizují

$$\sum_{j=1}^{p} (y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2, \tag{8}$$

a lasso regresí pak koeficienty minimalizující

$$\Sigma_{j=1}^{p}(y_j - \beta_j)^2 + \lambda \Sigma_{j=1}^{p}|\beta_j|. \quad (9)$$

Pro tento případ pak majíodhady hřebenové regrese tvar

$$\widehat{\beta_j^R} = y_j/(1+\lambda), \quad (10)$$

a odhady lasso regrese jsou ve tvaru

$$\widehat{\beta_j^L} = \begin{cases} y_j - \lambda/2 & pro\ y_j > \lambda/2 \\ y_j + \lambda/2 & pro\ y_j < -\lambda/2 \\ 0 & pro\ |y_j| \leq \lambda/2 \end{cases}.$$

Hřebenová regrese a lasso regrese provádí dva velmi odlišné typy zmenšování regresních koeficientů. V hřebenové regresi je každý koeficient zmenšován ve stejném poměru. Naproti tomu lasso zmenšuje koeficienty směrem k nule v konstantní míře $\lambda/2$. Koeficienty menší než $\lambda/2$ jsou pak zcela sníženy na hodnotu rovnou nule. Tento typ „shrinkage“, které lasso provádí je znám jako „soft-thresholding“. Fakt, že jsou některé odhady koeficientů lasso regrese rovny nule, odpovídá na otázku proč použití lasso regrese patří do metod výběru regresorů.

V případě obecnější datové matice X je pak celá situace ještě o něco komplikovanější, nicméně hlavní myšlenka zde stále platí. Hřebenová regrese snižuje koeficienty ve stejném poměru, zatímco lasso snižuje koeficienty směrem k nule a koeficienty s dostatečně malouhodnotou jsou pak přímo rovny nule.

Výběr penalizačního parametru

Stejně jako je třeba u subset selection zvolit vhodnou metodu výběru nejlepšího modelu, tak pro implementaci hřebenové a lasso regrese musíme zvolit metodu pro výběr optimální hodnoty penalizačního parametru λdo rovnic (1) a (3), respektive hodnotu s pro rovnice (4) a (5). Cross-validace nám tento problém pomůže velmi snadno vyřešit. Zvolíme si množství hodnot λ a vypočítáme pro ně cross-validační chybu. Následně pak volíme takovou hodnotu parametru λ, pro kterou je cross-validační chyba nejmenší. Nakonec vezmeme všechna dostupná pozorování, zvolenou hodnotu parametru λ a sestavíme výsledný model. [3, strana 219-228]

Praktická část

V následujících odstavcích budou feature selection metody popsané v předešlých kapitolách aplikovány na poskytnutá data, výsledky porovnány mezi sebou a nakonec popsány výhody a nevýhody jednotlivých přístupů. Taktéž zde budou popsány balíčky (packages) z R obsahující použité metody včetně syntaxe a popisu parametrů.

Data

Soubor pozorování, na který byly metody pro výběr proměnných aplikovány, představuje anonymizované údaje o pacientech s blíže nespecifikovaným nádorovým onemocněním, které jsou složeny z několika částí:

1) výsledek zpracování údajů z mikročipů (sloupce APn/APnCN – amplifikované úseky DNA, a DPn/DPnCN – deletované úseky DNA),

2) klinické údaje: věk (age), velikost nádoru (pT), postižení uzlin (pN), stádium onemocnění (stage), grading (G),

3) údaje o délce bezpříznakového období, tzv. desease-free survival (DFS) v měsících (DFSm), příslušné cenzorování (DFSevent), ,

4) údaj col25 související s typem léčby.

Základní úlohou je v datech nalézt faktory, které ovlivňují DFS přežívání, a to jednak se stanovenou hranicí 24 měsíců a poté celkově.

Vzhledem k povaze dat byla pro aplikaci a porovnání jednotlivých metod zvolena logistická regrese a Coxova regrese. Přestože Coxova regrese využívá více informací (čas místo prosté 0/1 proměnné), je vhodné představit i logistický regresní model, jenž umožňuje použití funkce `regsubsets()` z balíčku leaps pro metodu výběru faktorů best subsets. Funkci `regsubsets()` nelze na Coxův regresní model aplikovat.

Balíček LEAPS

Funkce regsubsets()

Funkci použijeme pro výběr modelu pomocí metod exhaustive search (bestsubset selection), forward/backward stepwise nebo sequential replacement (hybridní přístup).

Syntaxe:

```
regsubset(x = , data = , nbest = 1, nvmax = 8, force.in =
NULL, force.out = NULL, method = c("exhaustive", "backward",
"forward", "seqrep"), really.big = FALSE)
```

Parametry:

`x`	dizajnová matice nebo předpis modelu
`data`	data frame obsahující pozorování
`y`	vektor vysvětlované proměnné
`nbest`	počet nejlepších podmnožin modelu každé velikosti
`nvmax`	maximální velikost zkoumaných podmnožin
`force.in`	indexy sloupců dizajnové matice, které mají být vždy v modelu
`force.out`	indexy sloupců dizajnové matice, které nemají být v žádném modelu
`method`	metoda hledání nejlepších podmnožin („exhaustive“, „forward“, „backward“, „seqrep“)
`really.big`	musí být nastaveno na TRUE v případě velkého počtu proměnných

Balíček STATS

Funkce step()

Výběr modelu použitím stepwise algoritmu na základě AIC, resp. BIC (viz parametr k).

Syntaxe:

```
step(object, scope, direction = c("both", "backward",
"forward"), trace = 1, steps = 1000, k = 2)
```

Parametry:

`object`	objekt, představující model; výchozí model v algoritmu
`scope`	definuje rozsah zkoumaných modelů, jde o předpis modelu nebo `list` obsahující komponenty `upper` a `lower`
`direction`	typ algoritmu – `backward, forward, both.` Pokud není uvedeno, použije se `backward`
`trace`	pokud je kladné, pak se v každém kroku vypisují informace
`steps`	maximální počet kroků, defaultní hodnota je 1000
`k`	násobek počtu stupňů volnosti pro penalizaci, pro `k=2` dostáváme AIC, pro log(n) zase BIC.

Balíček SURVIVAL

Funkce coxph()

Funkce pro odhad Coxova regresního modelu proporcionálních rizik.

Syntaxe:

```
coxph(formula, data, ...)
```

Parametry:

`formula`	objekt ve tvaru předpisu modelu, s vysvětlovanou proměnnou nalevo od operátoru ~ a vysvětlujícími proměnnými napravo. Vysvětlovaná proměnná musí být ve tvaru survival objektu, vytvořeného pomocí funkce `Surv()`
`data`	data frame s pozorováními

Funkce Surv()

Funkce vytvoří survival objekt, který lze pak použít jako vysvětlovanou proměnnou ve funkci coxph

Syntaxe:

```
Surv(time, event, ...)
```

Parametry:

`time`	pro data zprava cenzorovaná jde o dobu sledování, pro intervalová data je to pak počátek časového intervalu
`event`	indikátor statusu, obvykle 0 = přežití, 1 = událost, např. smrt

Balíček PENALIZED

penalized()

Funkce na data použije zobecněný lineární model s L1 (lasso a fused lasso) a/nebo L2 (ridge) penalizací

Syntaxe:

```
penalized(response, penalized, unpenalized, lambda1 = 0,
lambda2 = 0, model = c("cox", "logistic", "linear", "poisson")
, steps = 1, trace = TRUE, standardize = FALSE...)
```

Parametry:

`response`	vysvětlovaná proměnná, číselný vektor pro lineární regresi, `Surv` objekt pro Coxovu regresi, vektor nul a jedniček pro logistickou regresi
`penalized`	penalizované proměnné
`unpenalized`	nepenalizované proměnné
`lambda`	penalizační parametr pro L1 a L2 penalizaci
`steps`	počet kroků
`trace`	pokud je `TRUE` vypisuje informace v každém kroku
`standardize`	před aplikováním penalizace provede standardizaci proměnných

Logistická regrese (LR)

Logistický regresní model je vhodné použít pro modelování DFS události, která nastane do 24 měsíců od diagnózy onemocnění. Jedná se tedy o dichotomickou proměnnou. Pro tyto účely je zapotřebí ze základního souboru se 133 pozorováními vyloučit ta, u nichž je délka sledování kratší než 24 měsíců a zároveň u nich nedošlo k DFS události, tj. relapsu onemocnění nebo úmrtí z libovolné příčiny. Celkově je pro logistickou regresi k dispozici 123 pozorování, z nichž 102 pacientů bylo bez DFS události a 21 pacientů mělo relaps onemocnění nebo zemřelo.

V případě několika metod výběru prvků (bestsubsets, forward, backward, stepwise, lasso) bude zapotřebí vyloučit i nekompletní pozorování vzhledem k vybrané podmnožině faktorů, takže se počty subjektů mohou v jednotlivých analýzách lišit.

Naivní přístup

V případě naivního přístupu k výběru proměnných do statistického modelu posuzujeme každou vysvětlující proměnnou zvlášť – zkoumáme její vliv na vysvětlovanou proměnnou a model je ve tvaru $y \sim x_i$. V R použijeme funkci `glm()` z balíčku `glm`, která na data aplikuje zobecněný lineární model. Ve funkci nastavujeme parametr `family` na „`binomial`".

Naivním přístupem výběru prvků, kdy do výsledného modelu zahrneme ty faktory, které mají v univariátních modelech p-hodnotu Waldova testu menší než 0,1 (10%-ní hladina významnosti), získáme následující výsledky:

```
Call:
glm(formula = as.formula(paste0("DFSevent~", paste(prom, collapse = "+"))),
    family = binomial, data = full.LR)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.76472  -0.39011  -0.17612  -0.04887   2.71366

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -7.1849     2.1290  -3.375 0.000739 ***
AP7           1.0786     1.1659   0.925 0.354932
AP19          0.6189     0.9135   0.677 0.498094
AP21          1.9984     0.9278   2.154 0.031244 *
AP23          1.0232     0.8757   1.168 0.242623
DP4           1.6625     0.7977   2.084 0.037161 *
DP9           0.7798     0.8084   0.965 0.334772
DP44          1.4561     0.7533   1.933 0.053244 .
DP59          0.4829     0.8590   0.562 0.574006
DP60          1.2040     0.8853   1.360 0.173831
DP61         -1.1281     0.8193  -1.377 0.168551
DP64          0.6470     1.0975   0.589 0.555531
DP65          0.4059     1.1332   0.358 0.720230
DP84          1.6073     0.8404   1.913 0.055806 .
pT            0.8357     0.8403   0.994 0.320011
pN            1.4327     0.7962   1.799 0.071963 .
stage        -0.9065     1.4126  -0.642 0.521041
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 104.855  on 119  degrees of freedom
Residual deviance:  58.716  on 103  degrees of freedom
  (3 observations deleted due to missingness)
AIC: 92.716

Number of Fisher Scoring iterations: 7
```

Obrázek č. 7

Do výsledného modelu jsmevybrali 16 proměnných z celkového počtu 121. Je vidět, že většina faktorů zahrnutých v modelu není statisticky významná. Jako významné negativní faktory DFS do 24 měsíců(na 5%-ní hladině významnosti) se jeví pouze genomické parametry DP4: OR=5.3, 95%CI= (1.1, 25.18) a AP21: OR = 7.4, 95% CI = (1.2, 45.46). Jako trend můžeme označit negativní vliv dalších genomických proměnných DP44, DP84 a postižení uzlin pN. Je zajímavé, že stádium onemocnění se v tomto případě neprojevuje jako významný faktor.

Stepwise selection

Pomocí funkce `step()` aplikujeme na data metody stepwise selection. Pokud pro stepwise selection v našem případě použijeme všechny vysvětlující proměnné (genomické proměnné AP a DP, klinické – age, stage, G, pT, pN a údaj související s typem léčby col25), tak v LR algoritmus bohužel zhavaruje (nedokonverguje). Řešením by pak mohlo být proměnné rozdělit na podskupiny a analyzovat je samostatně. Podobný postup může být také vyloučení pouze podskupiny genomických proměnných (delecí nebo amplifikací).

Pokud vyloučíme proměnné AP, které představují amplifikace, algoritmus opět nekonverguje. Algoritmus konverguje pouze pokud vyloučíme proměnné DP představující delece (datový soubor bez chybějících hodnot zahrnuje 120 pozorování 35 proměnných). Výstup funkce `summary()` pro výsledný model

DFSevent ~ AP7 + AP21 + AP23 + stage

při použití stepwise přístupu (parametr `direction` funkce `step()` nastaven na hodnotu „`both`“) je:

```
Call:
glm(formula = DFSevent ~ AP21 + stage + AP23 + AP7, family = binomial,
    data = na.omit(full.LR[, c(prom, "DFSevent")]))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0931  -0.6048  -0.3638  -0.2114   2.7613

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -4.8979     1.2313  -3.978 6.96e-05 ***
AP21          1.0759     0.5910   1.820   0.0687 .
stage         1.1080     0.5214   2.125   0.0336 *
AP23          1.4045     0.6662   2.108   0.0350 *
AP7           1.3939     0.7458   1.869   0.0616 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 104.855  on 119  degrees of freedom
Residual deviance:  82.041  on 115  degrees of freedom
AIC: 92.041

Number of Fisher Scoring iterations: 5
```

Obrázek č. 8

V modelus hodnotou AIC 92,041 jsou všechny proměnné významné alespoň na 10%-ní hladině, přičemž všechny ovlivňují DFS přežívání negativně, tj. Zvyšují šanci na DFS event do 24 měsíců:

Proměnná	OR	95%CI	p-value
AP7	4.03	(-0.068, 2.86)	0,069
AP21	2.93	(-0.082, 2.23)	0,034
AP23	4.07	(0.099, 2.71)	0,035
stage	3.03	(0.086, 2.13)	0,062

Mezi proměnné významné na 5%-ní hladině se již nyní dostalo stádium onemocnění, které vkládáme do modelu jako adjustační faktor, tj. jako numerickou proměnnou (OR se vztahuje k jednotkové změně). V našem případě tedy, jestliže se dva pacienti budou lišit ve stádiu o jednotku, bude mít pacient s vyšším stádiem přibližně 3x vyšší riziko relapsu onemocnění nebo úmrtí než pacient se stádiem o jednotku nižším.

Opět je zajímavé, že z dalších laboratorních ani klinických parametrů se žádný do výsledného modelu nedostal, ale figurují v něm 3 genomické parametry.

Použitím forward přístupu získáme totožný model jako pro stepwise přístup.

Backward stepwise selectiondokončí výběr proměnných (navíc velice rychle) a do výsledného modelu zahrne 11 proměnných, mezi nimiž nechybí ani velikost tumoru (pT), postižení uzlin (pN) a klinický parametr související s typem léčby (col25B). Tyto klinické proměnné jsou však v modelu nevýznamné (na 5%). Stádium onemocnění se tentokrát do modelu ani nedostalo. Proměnné významné alespoň na 10%-ní hladině spolehlivosti jsou uvedeny v tabulce.

Proměnná	OR	95%CI	p-value
AP7	40.36	(0.93, 6.47)	0.0089
AP8	0.087	(-5.31, 0.43)	0.095
AP19	87.92	(0.96, 7.99)	0.013
AP20	0.063	(-5.45, -0.07)	0.044
AP21	3.66	(-0.03, 2.63)	0.056
AP23	5.41	(0.18, 3.20)	0.028
pN	2.15	(-0.09, 1.63)	0.082

```
Call:
glm(formula = DFSevent ~ AP3 + AP7 + AP8 + AP9 + AP19 + AP20 +
    AP21 + AP23 + pT + pN + col25, family = binomial, data = na.omit(full.LR[,
    c(prom, "DFSevent")]))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4588  -0.5105  -0.2222  -0.0540   2.9612

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -5.0545     1.4030  -3.603 0.000315 ***
AP3           -1.1669     0.7825  -1.491 0.135875
AP7            3.6979     1.4127   2.618 0.008855 **
AP8           -2.4420     1.4617  -1.671 0.094800 .
AP9           -1.4428     0.9337  -1.545 0.122294
AP19           4.4764     1.7925   2.497 0.012516 *
AP20          -2.7593     1.3708  -2.013 0.044126 *
AP21           1.2964     0.6789   1.909 0.056200 .
AP23           1.6891     0.7692   2.196 0.028093 *
pT             0.6827     0.4809   1.420 0.155695
pN             0.7673     0.4416   1.738 0.082272 .
col25B       -16.7125  2000.7210  -0.008 0.993335
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 104.85  on 119  degrees of freedom
Residual deviance:  67.95  on 108  degrees of freedom
AIC: 91.95

Number of Fisher Scoring iterations: 17
```

Obrázek č. 9

Regsubsets

Funkce `regsubsets()` z balíčku `leaps` kromě klasického bestsubset přístupu (parametr `method="exhaustive"`) umožňuje i dopřednou (parametr `method="forward"`), zpětnou selekci (parametr `method="backward"`) či stepwise (parametr `method="seqrep"`).

V našem případě ovšem nemůžeme použít hodnotu parametru `method="exhaustive"`, tedy best subset selection, neboť ta je pro velký počet proměnných $(p \geq 40)$ na obyčejném počítači v rozumném čase nevypočitatelná (pro $p \geq 40$ algoritmus prohledává $2^{40} = 1099\,511\,627\,776$ modelů). Při použití metody `exhaustive` však můžeme omezit maximální velikost prohledávaných podmnožin proměnných nastavením hodnoty parametru `nvmax` na nějaké „rozumné" číslo. Na poskytnutých datech se 150 proměnnými jsem zkoušel různé nastavení parametru `nvmax` a sledoval rozdíly v časech:

`nvmax=1` – **0.17 s**

`nvmax=2` – **0.26 s**

`nvmax=3` – **5.11 s**

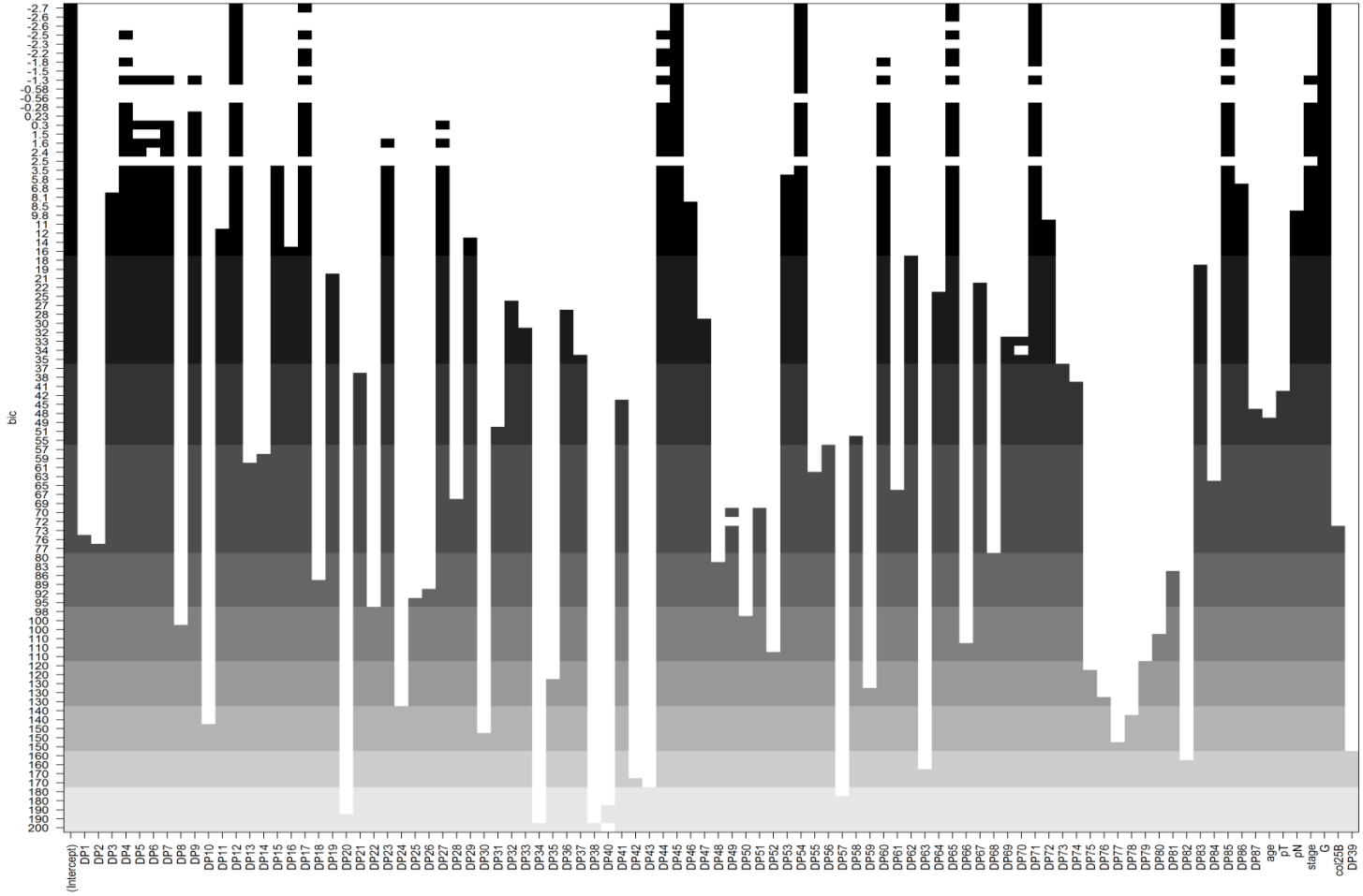
`nvmax=4` – **104.91 s**

`nvmax=5` – **1257.31 s**

Jak je vidět čas potřebný k nalezení modelu pomocí best subset selection velmi rychle stoupá. Pro `nvmax=10` algoritmus běžel cca 16 hodin (!) a stále ještě nenašel nejlepší model pro 10 proměnných.

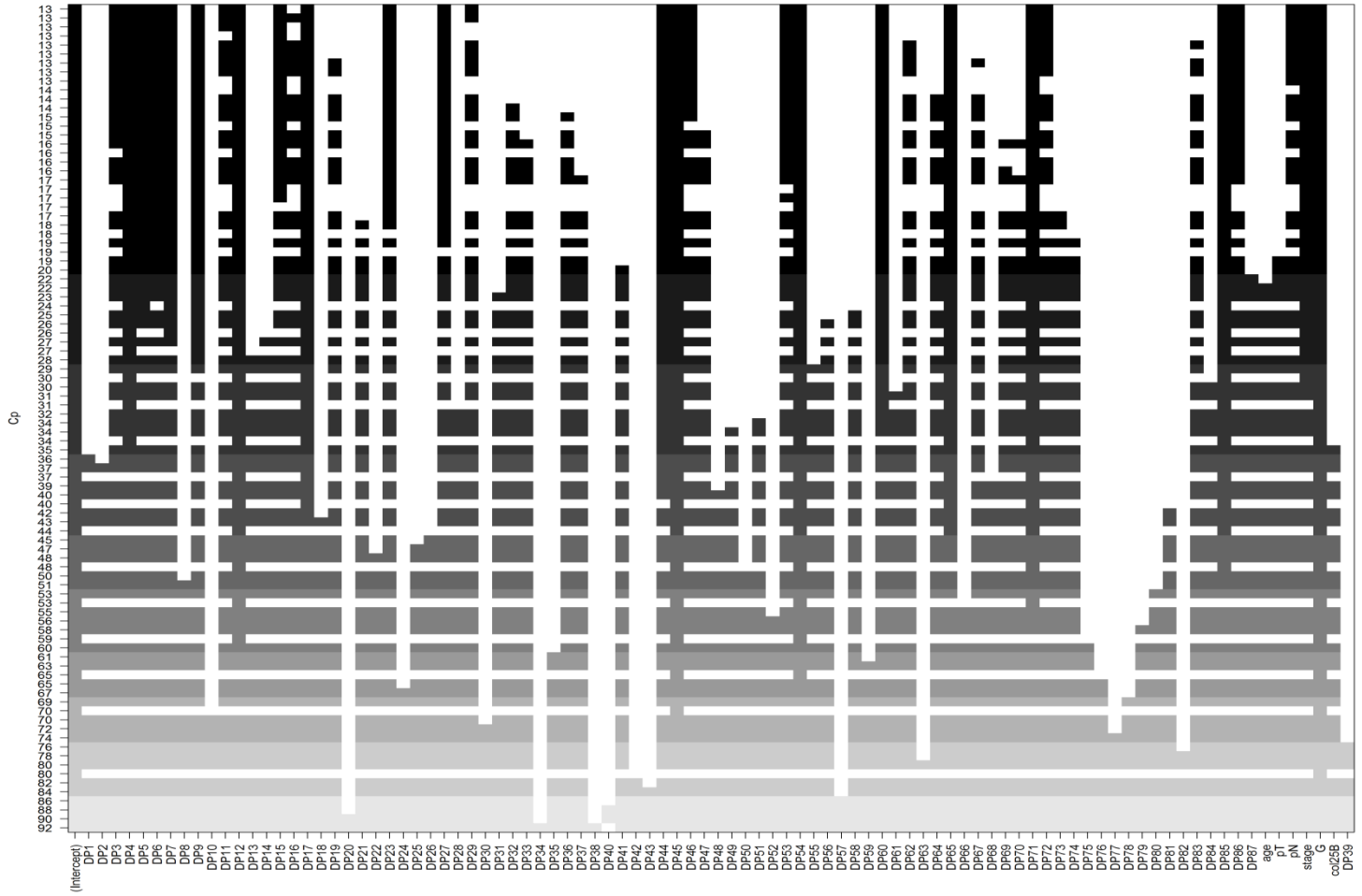
Z těchto důvodů jsem použil ve funkci `regsubsets()` metody `forward`, `backward` a `seqrep`. Výhodou funkce `regsubsets()` je možnost hledat podmnožinu faktorů nejen na základě BIC kritéria (jako `step()` pro `k=log(n)`), ale i jiných měr vhodnosti modelu. Pro porovnání s metodami postupného výběru jsem zkoumal výstupy pro dvě sady proměnných, které se liší jen typem vynechaných genomických proměnných – v jednom případě vynechávám všechny AP proměnné, ve druhém všechny DP proměnné. Výsledky pro míry BIC a Cp jsou přehledně znázorněny na obrázcích 10 – 21.

Forward, bez AP proměnných, dle BIC



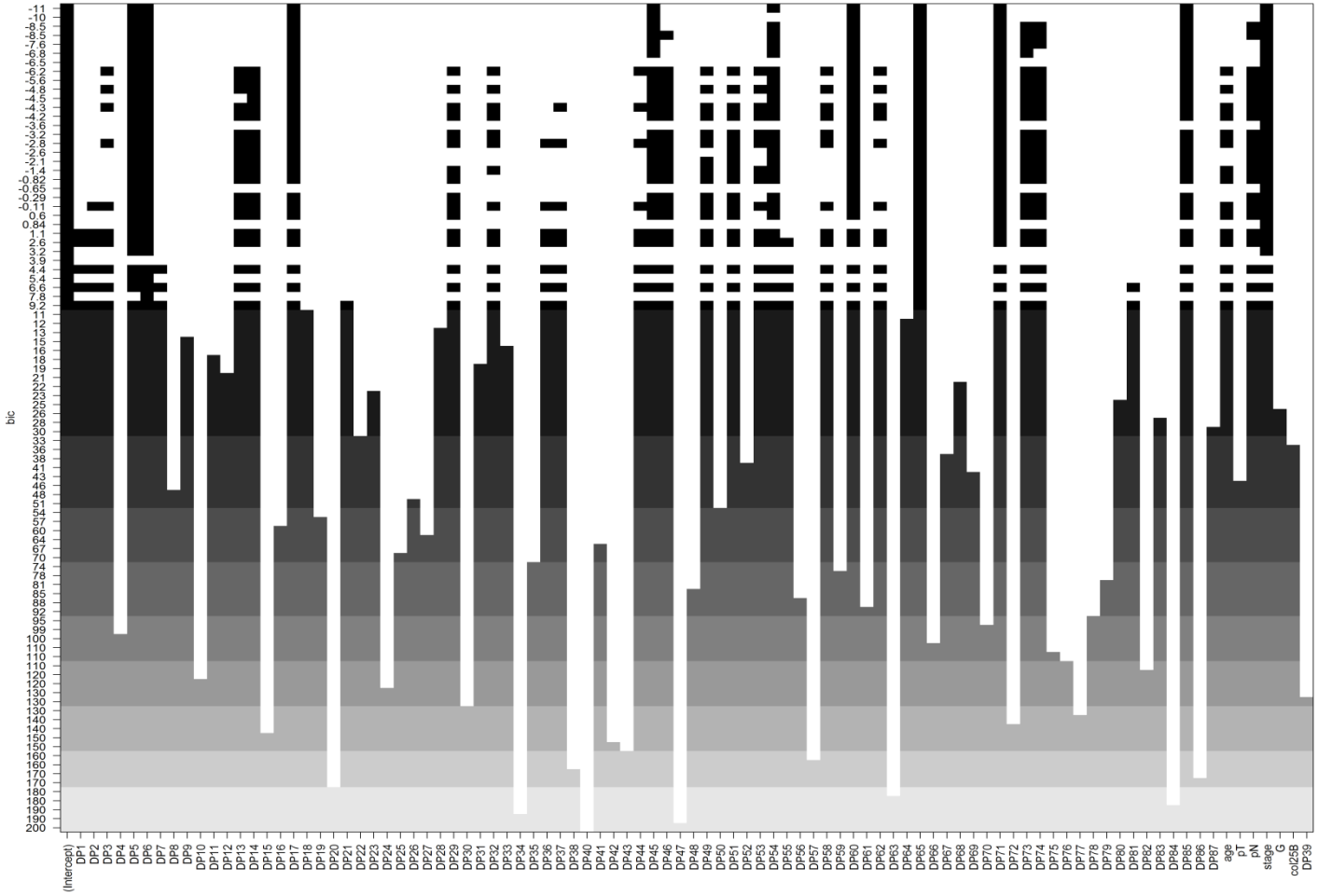
Obrázek č. 10

Forward, bez AP proměnných, dle Mallowsova Cp



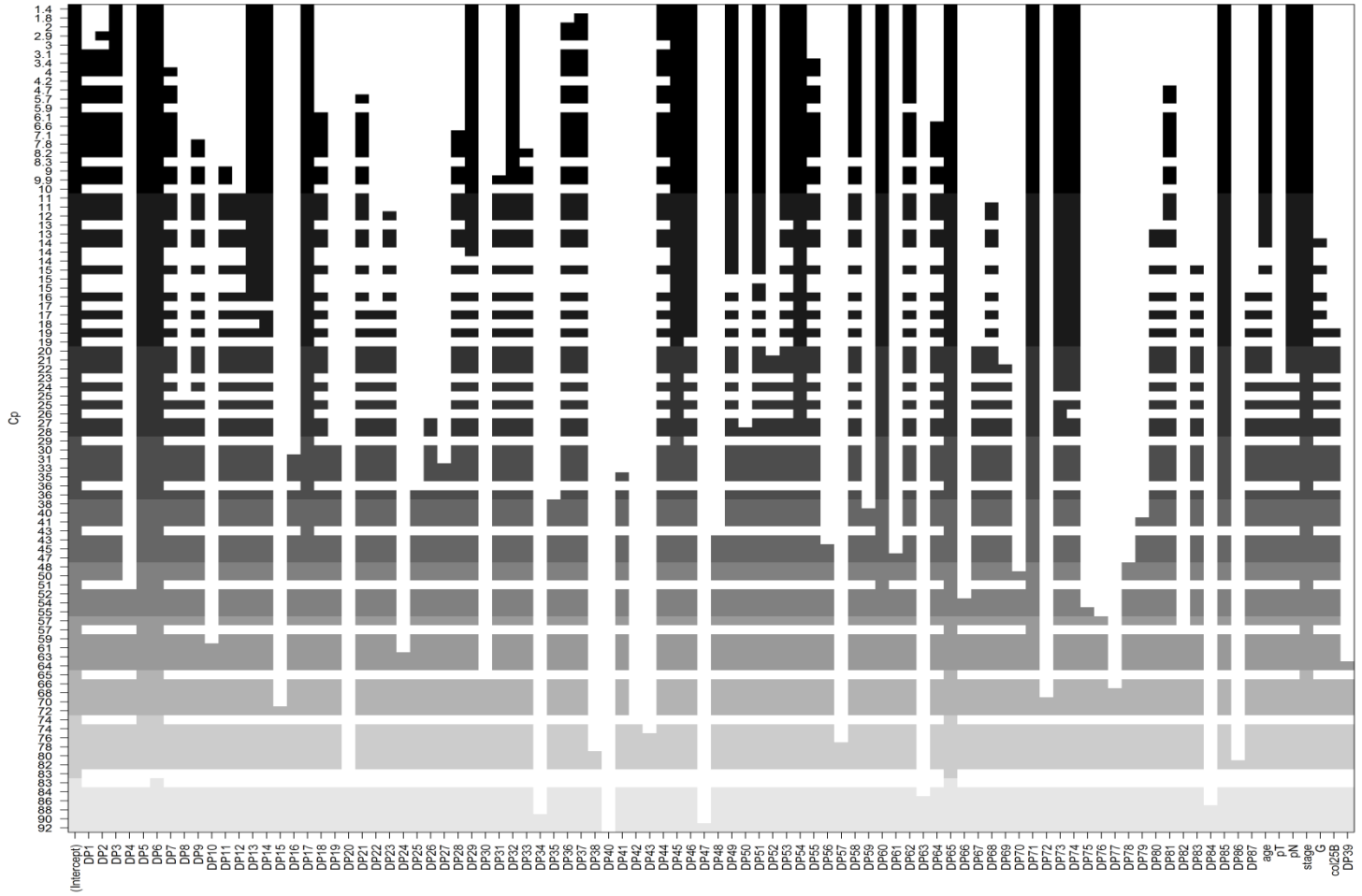
Obrázek č. 11

Backward, bez AP proměnných, dle BIC



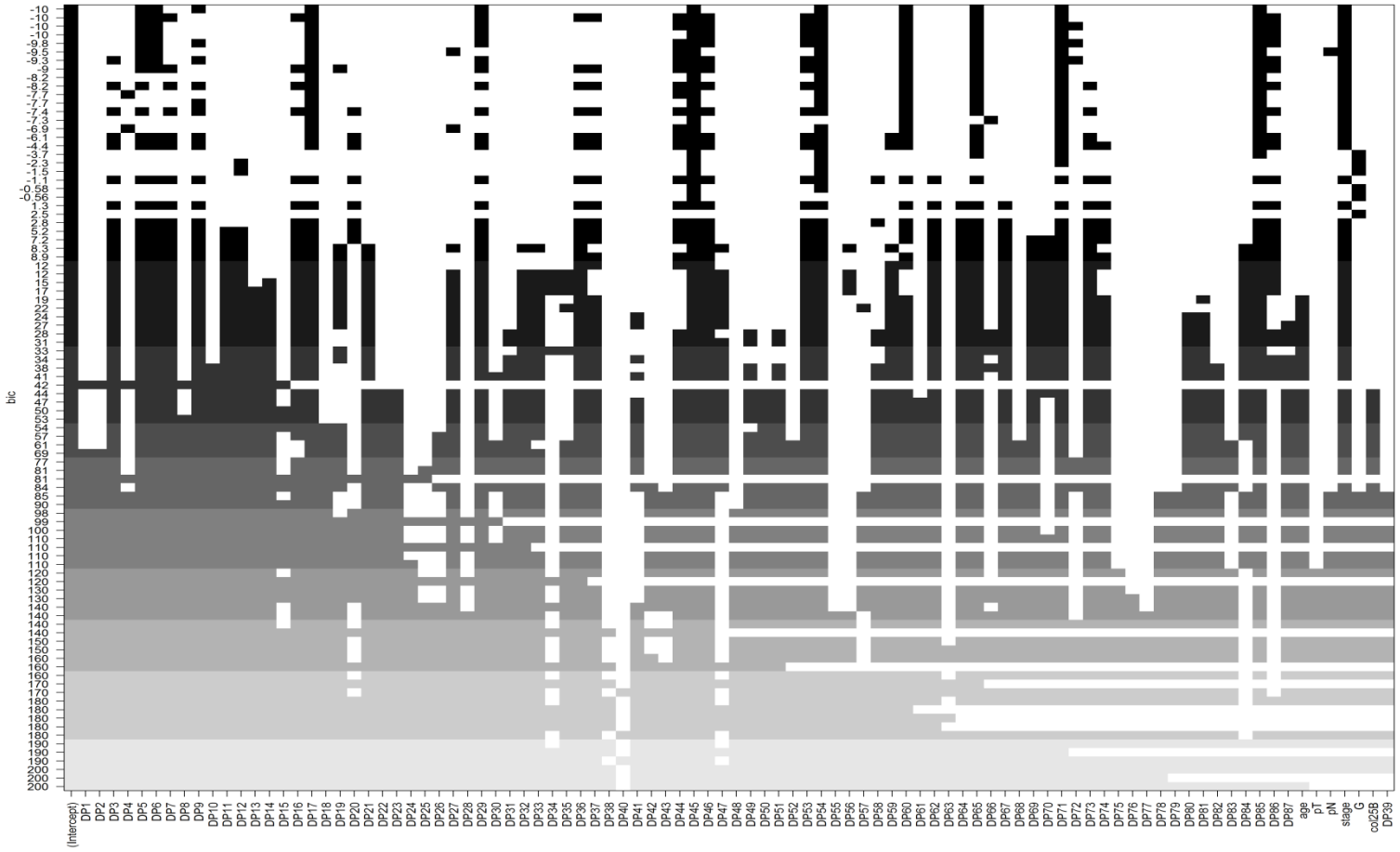
Obrázek č. 12

Backward, bez AP proměnných, dle Mallowsova Cp



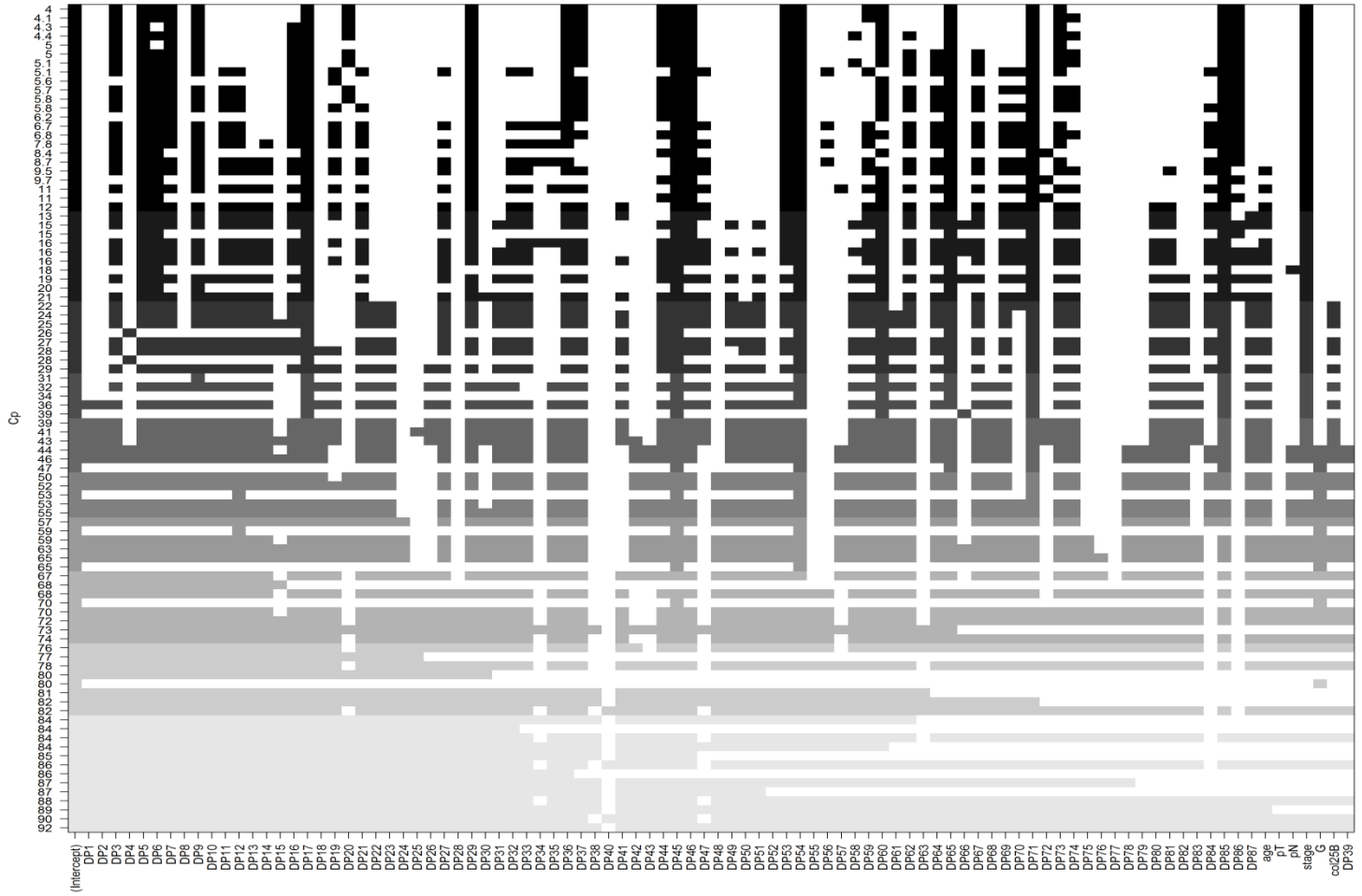
Obrázek č. 13

Seqrep, bez AP proměnných, dle BIC



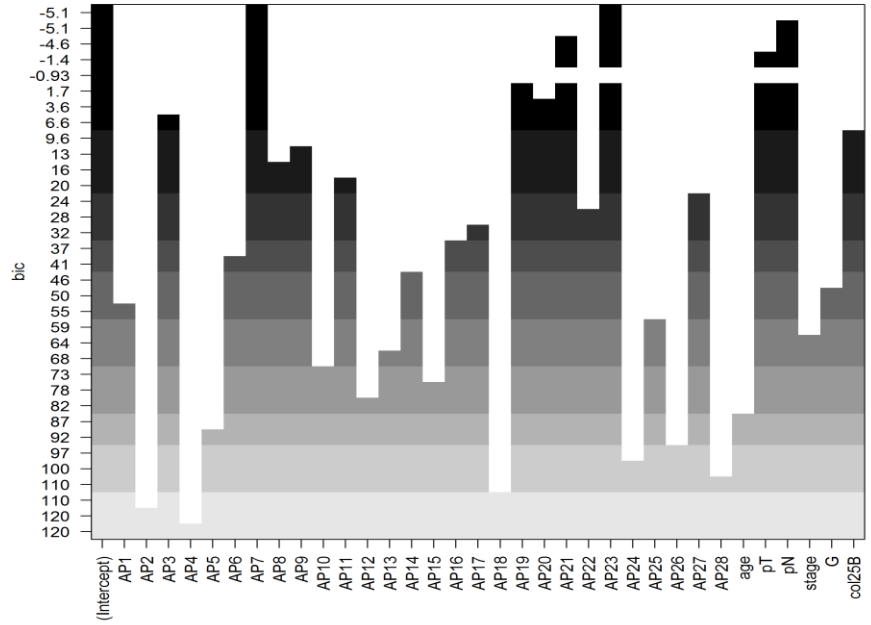
Obrázek č. 14

Seqrep, bez AP proměnných, dle Mallowsova Cp



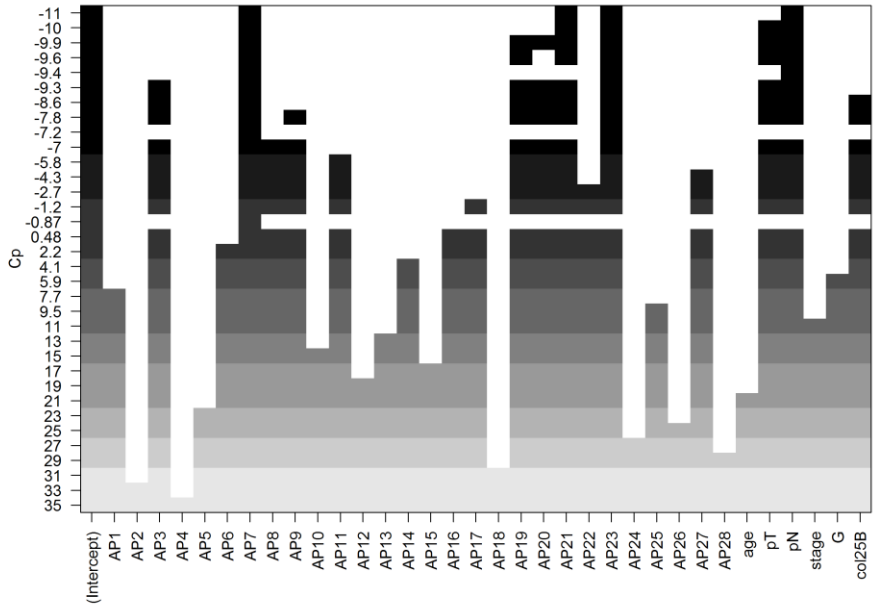
Obrázek č. 15

Forward, bez DP proměnných, dle BIC



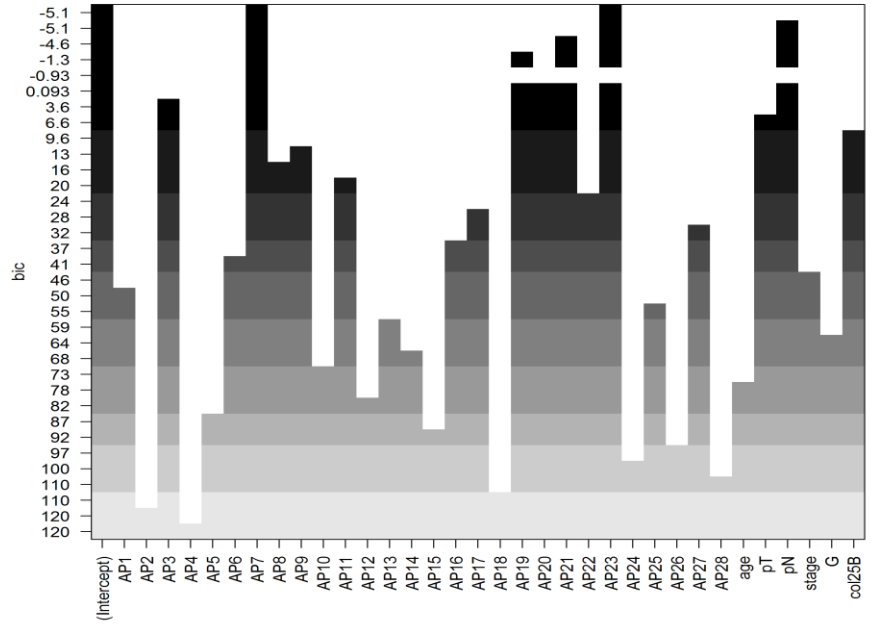
Obrázek č. 16

Forward, bez DP proměnných, dle Mallowsova Cp



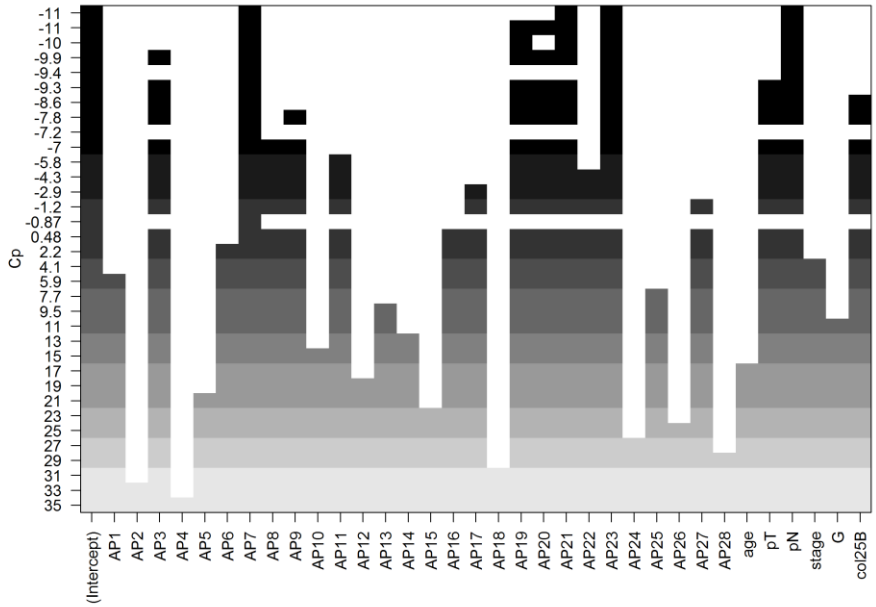
Obrázek č. 17

Backward, bez DP proměnných, dle BIC



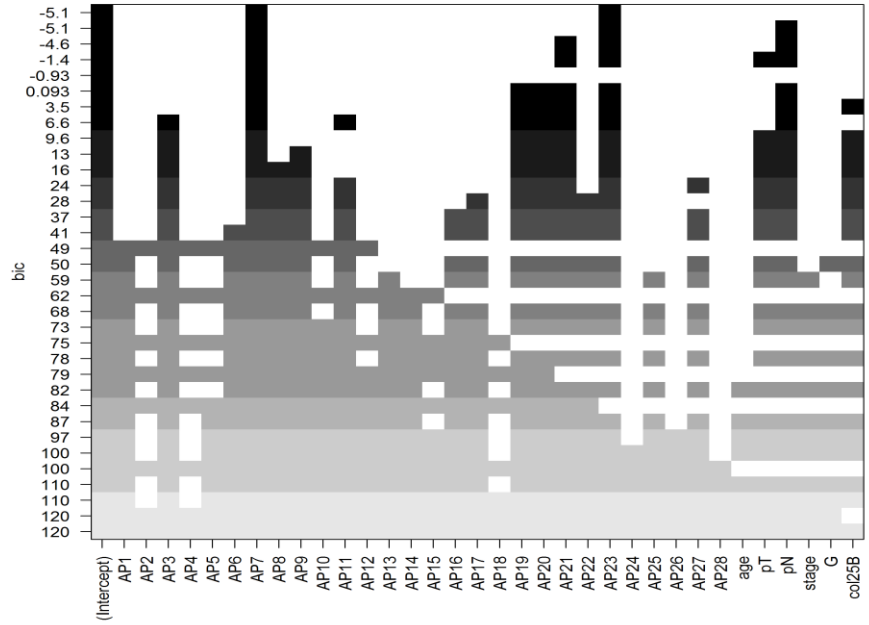
Obrázek č. 18

Backward, bez DP proměnných, dle Mallowsovo Cp



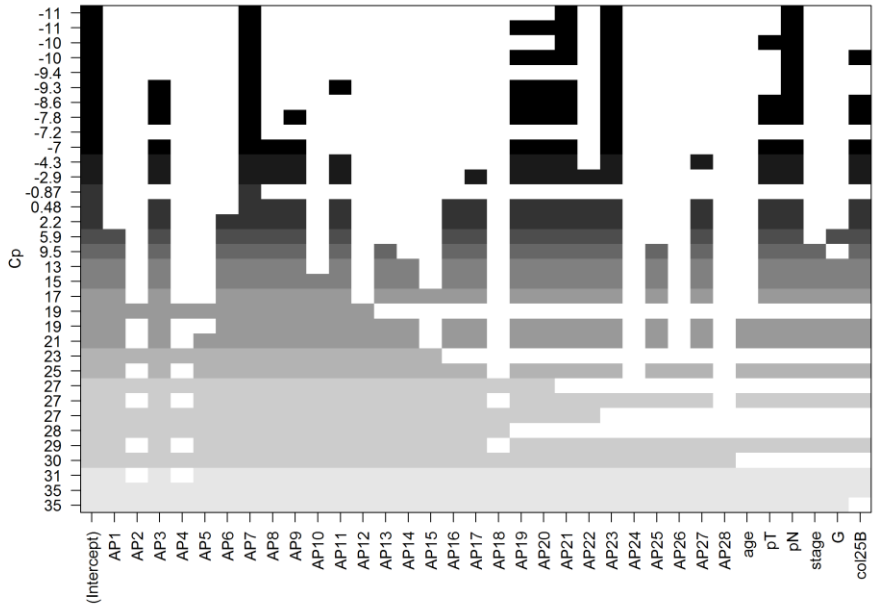
Obrázek č. 19

Seqrep, bez DP proměnných, dle BIC



Obrázek č. 20

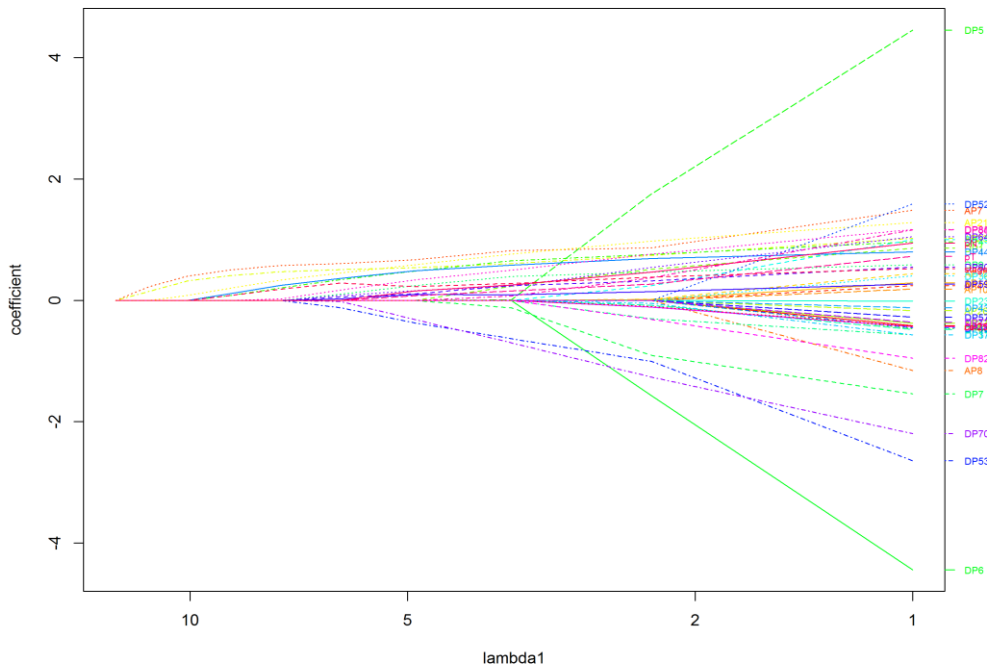
Seqrep, bez DP proměnných, dle Mallowsovo Cp



Obrázek č. 21

Lasso regrese

Při použití naivního přístupu jsme proměnné do výsledného modelu vybírali na základě výsledků univariátních modelů (které nezohledňují vliv ostatních proměnných) a na základě vlastního uvážení (10%-ní hladina významnosti). U forward, backward a stepwise selekce jsme nemohli použít kompletní sadu vysvětlujících proměnných a teprve po vyloučení všech úseků s delecemi (DP proměnné) byly algoritmy schopny poskytnout nějaké výsledky. Nadějí pro výběr z kompletní sady vysvětlujících proměnných jsou proto penalizační metody, zejména lasso regrese. V ní pro různou volbu penalizačního faktoru získáme různý počet nenulových koeficientů (na rozdíl od hřebenové regrese, kde se žádné koeficienty nevynulují), které reprezentují faktory, jež by se měly do výsledného modelu zahrnout. Čím bude vyšší hodnota penalizačního faktoru, tím menší počet nenulových parametrů obdržíme. Využijeme funkci `penalized()` ze stejnojmenného balíčku softwaru R, na jejíž výsledek následně aplikujeme funkci `plotpath`, která výsledek (hodnoty koeficientů) pro různé volby penalizačního faktoru lambda (parametr `lambda1`) vykreslí do grafu.



Obrázek č. 22

Pomocí tohoto grafu lze určit hodnotu lambda, kterou pak použijeme pro penalizaci proměnných.

Jak již bylo uvedeno, pro různé hodnoty lambda dostáváme jiný počet nenulových koeficientů (pozn.: do počtu je zahrnut i intercept), např. pro:

lambda = 5:

16 nenulových koeficientů – AP7, AP21, AP23, DP4, DP9, DP44, DP53, DP59, DP60, DP64, DP70, DP84, pT, pN, stage

lambda = 6:

15 nenulových koeficientů – AP7, AP21, AP23, DP4, DP9, DP44, DP53, DP59, DP60, DP64, DP70, DP84, pN, stage

lambda = 7:

11 nenulových koeficientů – AP7, AP21, AP23, DP4, DP9, DP44, DP60, DP64, DP84, stage

lambda = 8:

7 nenulových koeficientů – AP7, AP21, AP23, DP4, DP44, stage

lambda = 9:

7 nenulových koeficientů – AP7, AP21, AP23, DP4, DP44, stage

lambda = 10:

4 nenulové koeficienty – AP7, AP21, AP23

Výsledky finálního modelu

DFSevent ~ AP7 + AP21 + AP23 + DP4 + DP9 + DP44 + DP60 + DP64 + DP84 + stage

zahrnujícího proměnné, jejichž koeficienty byly nenulové pro volbu lambda = 7 , tj. genomické proměnné AP7, AP21, AP23, DP4, DP9, DP44, DP60, DP64, DP84a stádium onemocnění shrnuje funkce summary() takto:

```
Call:
glm(formula = model.lr, family = binomial, data = coxr)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9244  -0.5542  -0.3095  -0.1461   2.4962

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -6.7293     1.3783  -4.882 1.05e-06 ***
AP7           0.6672     0.8029   0.831  0.40600
AP21          0.7928     0.6246   1.269  0.20434
AP23          0.5607     0.6872   0.816  0.41452
DP4           1.2507     0.5730   2.183  0.02906 *
DP9           0.4814     0.6293   0.765  0.44426
DP44          0.1751     0.5637   0.311  0.75612
DP60          0.8239     0.5652   1.458  0.14494
DP64          0.7296     0.5967   1.223  0.22144
DP84          1.0677     0.5640   1.893  0.05834 .
stage         1.4648     0.4903   2.987  0.00281 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 131.426  on 132  degrees of freedom
Residual deviance:  93.143  on 122  degrees of freedom
AIC: 115.14

Number of Fisher Scoring iterations: 6
```

Obrázek č. 23

Ve finálním modelu se opět neobjevují klinické proměnné kromě stádia nemoci, které je významné dokonce na 1%-ní hladině významnosti a DFS ovlivňuje negativně. Tedy čím vyšší stádium nemoci pacient měl, tím je větší šance, že se u něj nemoc znovu objeví nebo tento člověk zemře. V modelu se také objevují genomické deleční proměnné DP4 (významná na 5%-ní hladině významnosti) a DP84 (s významností na 10%-ní hladině významnosti). Obě proměnné opět ovlivňují DFS negativně. Ostatní proměnné zahrnuté do modelu se jeví jako nevýznamné.

Proměnná	OR	95%CI	p-value
DP4	3.49	(0.13, 2.37)	0.029
DP84	2.91	(-0.38, 2.17)	0.058
stage	4.33	(0.50, 2.43)	0.003

Coxův regresní model

Před použitím coxova regresního modelu si musíme nejdříve upravit vysvětlovanou proměnnou pomocí funkce `Surv()`. Tato funkce vytvoří z časové (DFSm) a událostní (DFSevent) proměnné objekt, který je pak možné využít jako vysvětlovanou proměnnou ve funkci `coxph()`.

Naivní přístup

Stejně jako u naivního přístupu pro logistickou regresi, tak i pro Coxovu regresi zkoumáme postupně vliv jednotlivých vysvětlujících proměnných na vysvětlovanou proměnnou a funkcí `summary()` dostáváme následující výsledky:

```
Call:
coxph(formula = as.formula(paste0("Surv(DFSm,DFSevent)~", paste(prom,
    collapse = "+"))), data = coxr)

  n= 130, number of events= 24
   (3 observations deleted due to missingness)

         coef exp(coef) se(coef)      z Pr(>|z|)
AP7    0.8853    2.4238   0.6156  1.438 0.150363
AP21   1.0092    2.7434   0.5713  1.766 0.077315 .
AP22   0.3065    1.3586   0.4982  0.615 0.538419
AP23   1.2186    3.3823   0.6109  1.995 0.046087 *
AP28   1.2643    3.5406   0.5646  2.239 0.025143 *
DP4    1.8654    6.4588   0.5401  3.454 0.000553 ***
DP9    0.8123    2.2532   0.6222  1.306 0.191722
DP46   0.2513    1.2857   0.5144  0.489 0.625163
DP59   0.3074    1.3599   0.6353  0.484 0.628444
DP60   0.4537    1.5741   0.6638  0.683 0.494301
DP61  -0.3849    0.6805   0.5034 -0.765 0.444552
DP64   0.2304    1.2591   0.5621  0.410 0.681913
DP84   1.2292    3.4183   0.4756  2.584 0.009760 **
pT     0.8117    2.2517   0.3883  2.090 0.036609 *
pN     0.8301    2.2936   0.4443  1.868 0.061702 .
stage -0.8227    0.4392   0.8661 -0.950 0.342143
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Obrázek č. 24

Ve výsledném modelu se objevuje hned několik významných proměnných, všechny ovlivňují DFS negativně. Nejvýznamnější proměnnou je v tomto případě genomická deleční proměnná DP4, která je významná na 0,1%-ní hladině významnosti. Druhou nejvýznamnější je taktéž deleční proměnná a to DP84 s významností na 1%-ní hladině významnosti. Významnost na hladině 5% mají amplifikační proměnné AP23, AP28 a klinická proměnná pT. Posledními významnými proměnnými ve finálním modelu, a to na hladině významnosti 10%, jsou proměnné AP21 a pN. Můžeme si všimnou, že deleční proměnná DP61 ovlivňuje DFS pozitivně – tedy pokud se tato delece u pacienta objevuje, je vyšší šance, že se nemoc znovu neobjeví nebo pacient nezemře, nicméně je tato proměnná nevýznamná. V tomto případě je ještě zajímavé, že

stádium nemoci pozitivně ovlivňuje DFS, avšak také se jedná o nevýznamnou proměnnou.

Jelikož je výstupem funkce `coxph()` objekt typu coxph, lze velmi jednoduše získat tabulku s koeficienty a intervaly spolehlivosti pomocí funkce `summary()`:

```
      exp(coef) exp(-coef) lower .95 upper .95
AP7      2.4238     0.4126   0.72532     8.100
AP21     2.7434     0.3645   0.89535     8.406
AP22     1.3586     0.7360   0.51176     3.607
AP23     3.3823     0.2957   1.02139    11.201
AP28     3.5406     0.2824   1.17077    10.707
DP4      6.4588     0.1548   2.24075    18.617
DP9      2.2532     0.4438   0.66549     7.629
DP46     1.2857     0.7778   0.46914     3.523
DP59     1.3599     0.7353   0.39153     4.723
DP60     1.5741     0.6353   0.42856     5.782
DP61     0.6805     1.4694   0.25371     1.825
DP64     1.2591     0.7942   0.41839     3.789
DP84     3.4183     0.2925   1.34571     8.683
pT       2.2517     0.4441   1.05183     4.820
pN       2.2936     0.4360   0.96015     5.479
stage    0.4392     2.2767   0.08045     2.398

Concordance= 0.829  (se = 0.06 )
Rsquare= 0.327   (max possible= 0.825 )
Likelihood ratio test= 51.56  on 16 df,   p=1.292e-05
Wald test            = 39.44  on 16 df,   p=0.0009404
Score (logrank) test = 53.49  on 16 df,   p=6.278e-06
```

Obrázek č. 25

Stepwise selection

Pokud při stepwise selekci pro Coxův regresní model použijeme všechny proměnné (AP + DP + klinické + col25), pak algoritmus nekonverguje ani pro jednu z metod forward, backward, both. Po odejmutí genomických delečních proměnných DP a údaji o typu léčby col25, algoritmus pro metody backward a forward konverguje. Výsledné modely jsou zachyceny na obrázcích.

Výsledek pro metodu zpětného vyhledávání:

```
Call:
coxph(formula = Surv(DFSm, DFSevent) ~ AP2 + AP7 + AP13 + AP19 +
    AP20 + AP23 + AP28 + stage, data = na.omit(coxr))

        coef exp(coef) se(coef)     z       p
AP2   -0.777     0.460    0.497 -1.56 0.11797
AP7    1.566     4.788    0.526  2.98 0.00292
AP13   0.821     2.272    0.505  1.62 0.10422
AP19   2.579    13.183    0.876  2.94 0.00325
AP20  -2.113     0.121    0.786 -2.69 0.00716
AP23   2.017     7.515    0.537  3.75 0.00017
AP28   1.056     2.874    0.493  2.14 0.03224
stage  1.240     3.455    0.397  3.12 0.00180

Likelihood ratio test=34  on 8 df, p=4.01e-05
n= 104, number of events= 22
```

Obrázek č. 26

a model pro stepwise přístup, který se od předchozího liší pouze tím, že je zde navíc amplifikační proměnná AP24:

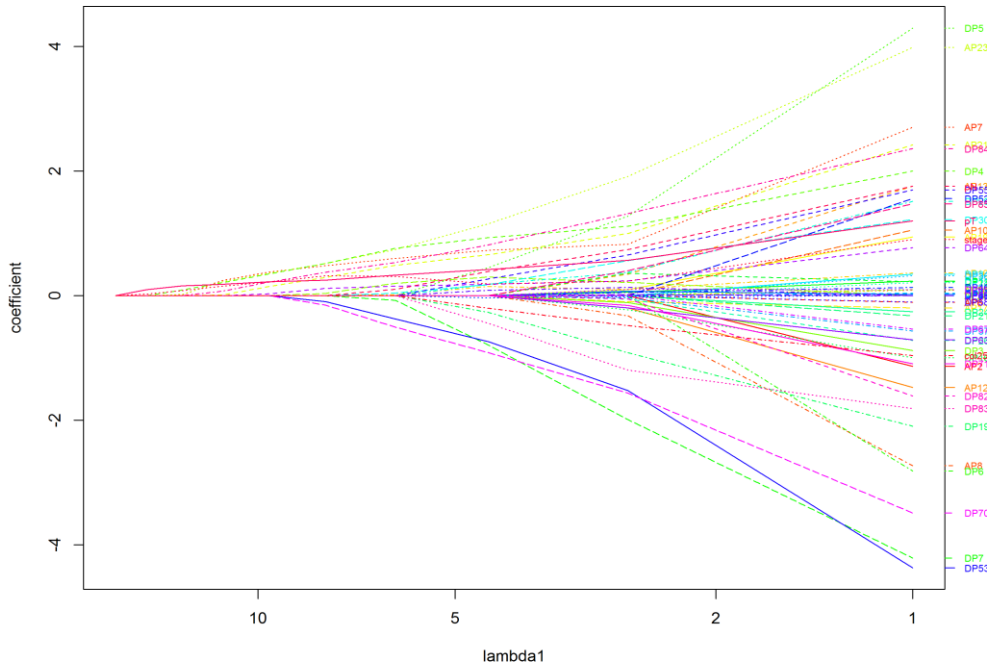
```
Call:
coxph(formula = Surv(DFSm, DFSevent) ~ AP2 + AP7 + AP13 + AP19 +
    AP20 + AP23 + AP28 + stage + AP24, data = na.omit(coxr))

         coef exp(coef) se(coef)     z       p
AP2   -1.0513    0.3495   0.5506 -1.91 0.05619
AP7    1.4559    4.2885   0.5420  2.69 0.00722
AP13   0.9576    2.6054   0.5136  1.86 0.06224
AP19   2.8417   17.1452   0.8907  3.19 0.00142
AP20  -2.3734    0.0932   0.7812 -3.04 0.00238
AP23   2.0113    7.4730   0.5504  3.65 0.00026
AP28   1.1831    3.2644   0.5046  2.34 0.01904
stage  1.2862    3.6191   0.3964  3.24 0.00118
AP24   0.8084    2.2443   0.5314  1.52 0.12818

Likelihood ratio test=36.3  on 9 df, p=3.56e-05
n= 104, number of events= 22
```

Obrázek č. 27

Lasso regrese

Jelikož všechny ostatní přístupy pro výběr proměnných v Coxově regresi v našem případě selhaly, nabízí se jako poslední možnost lasso regrese, ve které můžeme použít veškeré dostupné proměnné.

Nejdříve využijeme funkci `penalized()`, na jejíž výsledek opět aplikujeme funkci `plotpath`, která hodnoty koeficientů pro různé volby penalizačního faktoru lambda vykreslí do grafu.



Obrázek č. 28

Opět lze z grafu určit hodnotu lambda, kterou pak použijeme pro následnou penalizaci koeficientů.

Počty nenulových koeficientů pro různé hodnoty lambda dopadly následovně:

lambda = 5:

> 22 nenulových koeficientů – AP7, AP16, AP21, AP23, AP28, DP4, DP5, DP7, DP19, DP29, DP30, DP53, DP55, DP59, DP64, DP70, DP83, DP84, pT, pN, stage, col25

lambda = 6:

16 nenulových koeficientů – AP7, AP16, AP21, AP23, AP28, DP4, DP7, DP30, DP53, DP64, DP70, DP83, DP84, pT, pN, stage

lambda = 7:

12 nenulových koeficientů – AP7, AP21, AP23, AP28, DP4, DP53, DP64, DP70, DP84, pT, pN, stage

lambda = 8:

11 nenulových koeficientů – AP7, AP21, AP23, AP28, DP4, DP53, DP64, DP70, DP84, pT, stage

lambda = 9:

8 nenulových koeficientů – AP7, AP21, AP23, DP4, DP64, DP84, pT, stage

lambda = 10:

8 nenulových koeficientů – AP7, AP21, AP23, DP4, DP64, DP84, pT, stage

Výsledky finálního modelu

(DFSm, DFSevent) ~ AP7 + AP21 + AP23 + AP28+DP4 + DP53 + DP64 + DP70 + DP84 + pT + stage

zahrnujícího proměnné, jejichž koeficienty byly nenulové pro volbu lambda = 8 , tj. genomické proměnné AP7, AP21, AP23, AP28, DP4, DP53, DP64, DP70, DP84, velikost nádoru a stádium onemocnění shrnuje funkce `summary()` takto:

```
Call:
coxph(formula = model.lr, data = coxr)

  n= 133, number of events= 26

          coef exp(coef) se(coef)      z Pr(>|z|)
AP7    1.02362   2.78326  0.56958  1.797 0.072310 .
AP21   1.22749   3.41266  0.50071  2.452 0.014226 *
AP23   1.57275   4.81988  0.54497  2.886 0.003902 **
AP28   0.39373   1.48250  0.47270  0.833 0.404877
DP4    1.79686   6.03071  0.49038  3.664 0.000248 ***
DP53  -2.31555   0.09871  0.87321 -2.652 0.008007 **
DP64   0.25614   1.29193  0.50572  0.506 0.612512
DP70  -2.10702   0.12160  0.60914 -3.459 0.000542 ***
DP84   0.73164   2.07848  0.42782  1.710 0.087241 .
pT     0.62916   1.87602  0.28451  2.211 0.027008 *
stage  1.56136   4.76532  0.61134  2.554 0.010649 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Obrázek č. 29

Jak je z obrázku patrné, v modelu se nám objevují dvě velice významné delece – DP4 a DP70, obě s významností na 0,1% hladině. Můžeme si všimnout, že DP4 ovlivňuje DFS negativně, zatímco DP70 pozitivně, tedy u daného pacienta je pak vyšší šance, že se nemoc nevrátí nebo nezemře. Na 1%-ní hladině významnosti sjou významné amplifikace AP23, která DFS ovlivňuje negativně, a DP53 ovlivňující DFS naopak pozitivně. AP21, velikost nádoru a stádium onemocnění jsou významné proměnné na 5%-ní hladině. Všechny negativně ovlivňují DFS. Posledními významnými proměnnými na 10%-ní hladině jsou AP7 a DP84. Zbývající proměnné AP28 a DP64 se v modelu vyskytují jako nevýznamné.

Jelikož se jedná o objekt typu coxph, koeficienty a intervaly spolehlivosti jsou zobrazeny na následujícím obrázku:

```
       exp(coef) exp(-coef) lower .95 upper .95
AP7      2.78326     0.3593   0.91144    8.4992
AP21     3.41266     0.2930   1.27905    9.1054
AP23     4.81988     0.2075   1.65637   14.0254
AP28     1.48250     0.6745   0.58700    3.7441
DP4      6.03071     0.1658   2.30651   15.7682
DP53     0.09871    10.1305   0.01783    0.5466
DP64     1.29193     0.7740   0.47948    3.4810
DP70     0.12160     8.2237   0.03685    0.4013
DP84     2.07848     0.4811   0.89863    4.8074
pT       1.87602     0.5330   1.07415    3.2765
stage    4.76532     0.2098   1.43787   15.7930

Concordance= 0.866  (se = 0.058 )
Rsquare= 0.393   (max possible= 0.843 )
Likelihood ratio test= 66.31  on 11 df,   p=6.101e-10
Wald test            = 44.56  on 11 df,   p=5.803e-06
Score (logrank) test = 64.7  on 11 df,   p=1.227e-09
```

Obrázek č. 30

Shrnutí – výhody a nevýhody jednotlivých přístupů

Naivní přístup

+ lze zkoumat všechny proměnné jelikož se analyzují po jedné

+ můžeme zahrnout maximální počet pozorování

- „nebere ohled“ na ostatní proměnné (model ignoruje ostatní proměnné), lze však adjustovat vůči vybraným proměnným

Best subset selection

+ lze zkoumat všechny proměnné současně

+ modely lze analyzovat podle různých měr (AIC, BIC, Mallowsovo Cp,...)

- pro větší počet proměnných lze použít jen forward/backward přístup, nikoliv exhaustive

- v reálném čase výpočetně nemožné pro $p \geq 40$

- nutno vyloučit pozorování s chybějícími údaji

Stepwise selection

+ modely zahrnují více faktorů (na rozdíl od naivního přístupu)

+ lze použít hybridní přístup – kombinace forward a backward postupů

+ výpočetně velmi rychlé metody

- forward/backward přístup neumožňuje vyloučení/zahrnutí dříve zahrnutých/vyloučených proměnných

- nutno vyloučit pozorování s chybějícími údaji

Lasso regrese

+ lze zkoumat všechny proměnné naráz

+ použitelné i pro mnoho proměnných

+ různou volbou parametru lambda dostáváme různé množství nenulových koeficientů

- nutno vyloučit pozorování s chybějícími údaji

Na následujících dvou stranách jsou přehledně v tabulce uvedeny ty proměnné, které byly vybrány danou metodou alespoň do jednoho modelu. Jednotlivé znaky uvedené v tabulce vysvěluje legenda:

o	je v modelu, ale není významná
*	významná na 10%-ní hladině
x	významná na 5%-ní hladině
xx	významná na 1%-ní hladině
xxx	významná na 0,1%-ní hladině
(-)	negativně ovlivňuje DFS
(+)	pozitivně ovlivňuje DFS

LOGISTICKÁ REGRESE

		stepwise			lasso					
	naivní přístup	**both (bez DP)**	**F (bez DP)**	**B (bez DP)**	**lambda=5**	**lambda=6**	**lambda=7**	**lambda=8**	**lambda=9**	**lambda=10**
AP3				o (+)						
AP7	o (-)	* (-)	* (-)	xx (-)	o (-)	o (-)	o (-)	o (-)	o (-)	* (-)
AP8				* (+)						
AP9				o (+)						
AP19	o (-)			x (-)						
AP20				x (+)						
AP21	x (-)	* (-)	* (-)	* (-)	* (-)	o (-)	o (-)	* (-)	* (-)	o (-)
AP23	o (-)	x (-)	x (-)	x (-)	o (-)	o (-)	o (-)	o (-)	o (-)	* (-)
DP4	x (-)				x (-)	x (-)	x (-)	x (-)	x (-)	
DP9	o (-)				o (-)	o (-)	o (-)			
DP44	* (-)				o (-)	o (-)	o (-)	o (-)	o (-)	
DP53					* (+)	* (+)				
DP59	o (-)				o (-)	o (-)				
DP60	o (-)				o (+)	o (-)	o (-)			
DP61	o (-)									
DP64	o (-)				o (-)	o (-)	o (-)			
DP65	o (-)									
DP70					xx (+)	xx (+)				
DP84	* (-)				x (-)	* (-)	* (-)			
pT	o (-)			o (-)	* (-)					
pN	* (-)			* (-)	o (-)	o (-)				
stage	o (-)	x (-)	x (-)		o (-)	* (-)	xx (-)	xx (-)	xx (-)	
col25				o (+)						

COXŮV REGRESNÍ MODEL

		stepwise			lasso					
	naivní přístup	**both (bez DP)**	**F (bez DP)**	**B (bez DP)**	**lambda=5**	**lambda=6**	**lambda=7**	**lambda=8**	**lambda=9**	**lambda=10**
AP2		o (+)		o (+)						
AP7	o (-)	o (-)		o (-)	* (-)	o (-)	* (-)	* (-)	o (-)	o (-)
AP13		o (-)		o (-)						
AP16					o (+)	o (-)				
AP19		o (-)		o (-)						
AP20		o (+)		o (+)						
AP21	* (-)				xx (-)	x (-)	xx (-)	x (-)	* (-)	* (-)
AP22	o (-)									
AP23	x (-)	o (-)		o (-)	xxx (-)	xx (-)	xx (-)	xx (-)	* (-)	* (-)
AP24		o (-)								
AP28	x (-)	o (-)		o (-)	o (+)	o (-)	o (-)	o (-)		
DP4	xxx (-)				xx (-)	xx (-)	xxx (-)	xxx (-)	xx (-)	xx (-)
DP5					xx (-)					
DP7					o (+)	o (+)				
DP9	o (-)									
DP19					xxx (+)					
DP29					o (-)					
DP30					* (-)	* (-)				
DP46	o (-)									
DP53					xxx (+)	x (+)	x (+)	xx (+)		
DP55					x (-)					
DP59	o (-)				o (-)					
DP60	o (-)									
DP61	o (+)									
DP64	o (-)				o (-)	o (-)	o (-)	o (-)	o (-)	o (-)
DP70					xxx (+)	xxx (+)	xxx (+)	xxx (+)		
DP83					xxx (+)	* (+)				
DP84	xx (-)				xxx (-)	x (-)	* (-)	* (-)	x (-)	x (-)
pT	x (-)				* (-)	* (-)	* (-)	x (-)	x (-)	x (-)
pN	* (-)				xx (-)	x (-)	o (-)			
stage	o (+)	o (-)		o (-)	o (-)	o (-)	o (-)	x (-)	o (-)	o (-)
col25					o (+)					

Závěr

Cílem práce bylo zpracovat teorii týkající se Feature Selection metod, prozkoumat jejich použití ve statistickém softwaru R, aplikovat metody na poskytnutý datový soubor, popsat jejich výsledky a zhodnotit výhody a nevýhody jednotlivých přístupů selekce proměnných do statistického modelu. Myslím, že všechny tyto cíle byly v rámci diplomové práce splněny. V prvních kapitolách jsem popsal teorii k jednotlivým metodám, na začátku praktické části pak potřebné balíčky software R a důležité funkce z nich, následně jsem aplikoval metody z teoretické části na poskytnutá data a interpretoval výsledky. Na závěr jsem uvedl přehled jednotlivých metod, jejich výhody/nevýhody a vytvořil tabulku, ve které se objevily všechny proměnné, které byly alespoň jednou vybrány některým z přístupů selekce. V tabulkách je uvedna významnost jednotlivých proměnných a příznak, zda ovlivňují přežívání negativně či pozitivně.

Pro aplikaci metod z teoretické části jsem obdržel soubor anonymizovaných údajů o pacientech s jistým nádorovým onemocněním. Celkem v tomto souboru bylo 138 pozorování na 151 proměnných. Postupnou aplikací různých přístupů vyplynulo, že bohužel nebude možné použít celý soubor, ale analýza bude probíhat na podskupinách proměnných – většina algoritmů pro kompletní datový soubor nekonvergovala. Konkrétně byly v jednom případě vyloučeny genomické amplifikační proměnné AP v druhém případě deleční proměnné DP. Výsledky jednotlivých metod v rámci logistické a Coxovy regrese shrnují tabulky na posledních stránkách práce.

Na diplomovou práci by se mohlo ještě dále navázat a více ji rozvinout tímto způsobem – jelikož z modelů velmi rychle vypadnuly klinické proměnné (např. věk se v žádném z výsledných modelů vůbec neobjevuje) a lékaři často požadují, aby model tyto proměnné vždy obsahoval, bylo by možné jednotlivé metody modifikovat a nastavením určitých parametrů požadované proměnné do modelů vždy zahrnout, obdržené výsledky by pak mohly být velmi zajímavé (za předpokladu konvergence algoritmů).

Tato práce může čtenáři posloužit jako teoretický podklad k metodám typu Feature Selection a jejich využití ve statistickém softwaru R. Vypracování diplomové práce mě velmi bavilo a pokud bych se měl někdy v budoucnu statistikou zabývat, tak by jasnou volbou byla určitě statistická analýza v lékařství.

Literatura

[1] OLSSON, Ulf. *Generalized linear models an applied approach*. Lund: Studentlitteratur, 2006. ISBN 9144031416.

[2] COLLETT, D. *Modelling survival data in medical research*. 2nd ed. Boca Raton: Chapman & Hall/CRC, c2003. Text in statistical science. ISBN 1-58488-325-1.

[3] JAMES, Gareth R. *An introduction to statistical learning: with applications in R*. New York: Springer, c2013. Springer texts in statistics, 103. ISBN 978-1-4614-7137-0.

[4] MELOUN, Milan a Jiří MILITKÝ. *Statistická analýza experimentálních dat*. Vyd. 2., upr. a rozš. Praha: Academia, 2004. ISBN 80200-1254-0.

[5] DOBSON, Annette J. *An introduction to generalized linear models*. 2nd ed. Boca Raton: CRC Press, c2002. Texts in statistical science. ISBN 1-58488-165-8.

[6] VOLINSKY, Chris T. *Bayesian Information Criterion for Censored Survival Models*, Technical report no. 349, Department of Statistics, University of Washington, 1998

[7] RONCHETTI, Elvezio, *Robustnes Aspects of Model Choice*, Department of Econometrics, University of Geneva, CH-1211 Geneva, Switzerland, 1996

[8] *Journal of the Royal Statistical Society*. *Series B (Methodological)*,Volume 58, Issue 1 (1996), 267-288. (http://statweb.stanford.edu/~tibs/lasso/lasso.pdf, 10.4.2016)

[9] *Statistics in Medicine*, Volume 16 (1997), 385-395. (http://statweb.stanford.edu/~tibs/lasso/fulltext.pdf, 10.4.2016)

[10] MELOUN, Milan, *Základy logistické regrese*, 2007. ISBN 978-80-239-8998-4.

[11] CLAUSEN, Jens, *Branch and Bound Algorithms – Principles and Examples*, (http://www.imada.sdu.dk/Employees/jbj/heuristikker/TSPtext.pdf, 7. 5. 2016)