UNIVERZITA KOMENSKÉHO V BRATISLAVE

Fakulta matematiky, fyziky a informatiky

Sekvenčné metódy konštrukcie rozhodovacieho pravidla

2013 Bc. Veronika Kleinová

UNIVERZITA KOMENSKÉHO V BRATISLAVE

Fakulta matematiky, fyziky a informatiky

Sekvenčné metódy konštrukcie rozhodovacieho pravidla

Diplomová práca

Študijný program: Ekonomická a finančná matematika
Študijný odbor: 1114 Aplikovaná matematika
Školiace pracovisko: Katedra aplikovanej matematiky a štatistiky
Vedúci diplomovej práce: doc. Mgr. Radoslav Harman, PhD.

Bratislava 2013 **Bc. Veronika Kleinová**

17075452

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Veronika Kleinová
Študijný program: ekonomická a finančná matematika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.1.9. aplikovaná matematika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský

Názov: Sekvenčné metódy konštrukcie rozhodovacieho pravidla

Cieľ: Porovnať rôzne prístupy k sekvenčnej konštrukcii rozhodovacieho pravidla pre diskriminačnú analýzu založenú na logistickej regresii.

Vedúci: doc. Mgr. Radoslav Harman, PhD.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Daniel Ševčovič, CSc.

Dátum zadania: 25.01.2012

Dátum schválenia: 26.01.2012 prof. RNDr. Daniel Ševčovič, CSc.
garant študijného programu

.....
študent

.....
vedúci práce

Prehlásenie

Čestne prehlasujem, že túto prácu som vypracovala samostatne pod vedením vedúceho diplomovej práce a s použitím uvedenej literatúry.

V Bratislave, 23. apríla 2013 .....

podpis autora práce

Poďakovanie

Ďakujem vedúcemu mojej diplomovej práce, doc. Mgr. Radoslavovi Harmanovi, PhD. za pomoc a ochotu pri tvorbe práce. Zároveň ďakujem priateľovi a rodine za podporu.

Abstrakt

Kleinová, Veronika: Sekvenčné metódy konštrukcie rozhodovacieho pravidla [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: doc. Mgr. Radoslav Harman, PhD.

Hlavnou úlohou sekvenčných metód konštrukcie rozhodovacieho pravidla je dosiahnuť vyššiu presnosť klasifikačného pravidla s použitím menšieho počtu dát. Je to možné dosiahnuť vtedy, keď klasifikačnému modelu povolíme vyberať si dáta, ktorých kategóriu určíme a na základe ktorých je potom klasifikačné pravidlo vytvárané. Použitie sekvenčných metód klasifikácie dát je výhodné najmä v situáciách, keď získanie neoznačených (neklasifikovaných) dát je veľmi jednoduché, avšak zistenie ich skutočnej kategórie je náročné (finančne, časovo, atď.).

Kľúčové slová: logistická regresia, klasifikačné pravidlo, stratégie a scenáre výberu dát, informatívnosť dát.

Abstract

Kleinová, Veronika: Active learning [Diploma thesis], Comenius University Bratislava, Faculty of Mathematics, Physics, and Informatics, Department of Applied Mathematics and Statistics; Thesis Consultant: doc. Mgr. Radoslav Harman, PhD.

The key idea behind the active learning is to gain higher accuracy of a classification rule with fewer input data. It can be achieved by allowing the classification model to choose data that will be labeled. The classification model is then trained on these labeled instances. The use of active learning is especially beneficial in situations where unlabeled data are readily available but labels are difficult to obtain (i.e. expensive, time-consuming).

Key words: logistic regression, classification rule, query strategies and query scenarios, informativeness of data.

Obsah

Úvod **3**

1 Logistická regresia **5**

1.1 Binárna logistická regresia 5

1.2 Odhad parametrov 7

1.3 Fisherova miera informácie 9

1.3.1 Cramérova-Raova nerovnosť 10

1.4 Logistická diskriminačná analýza 11

2 Sekvenčné metódy konštrukcie rozhodovacieho pravidla **14**

2.1 Scenáre výberu dát na zatriedenie 15

2.1.1 Membership query synthesis 16

2.1.2 Stream-based selective sampling 16

2.1.3 Pool-based sampling 17

2.2 Stratégie výberu dát 19

2.2.1 Uncertainty sampling 19

2.2.2 Query by Committee 26

2.2.3 Expected model change 30

2.2.4 Expected error reduction 32

2.2.5 Variance reduction 33

2.2.6 Expected classification entropy 36

2.2.7 Porovnanie stratégií 38

Záver **50**

Príloha 1 **54**

Príloha 2 **59**

Zoznam obrázkov

1 Logistická funkcia 6
2 Cyklus scenára pool based sampling. 17
3 Scenáre výberu dát. 18
4 Iris dáta (druhy Iris setosa a Iris virginica) 22
5 Model natrénovaný na všetkých dátach 23
6 Inicializačný model 24
7 Informatívnosť podľa *least confidence* 25
8 Pretrénovanie modelu (*least confidence*) 26
9 Informatívnosť podľa *least confidence* 2 27
10 Konečný model (*least confidence*) 28
11 Informatívnosť podľa *vote entropy* 30
12 Informatívnosť podľa *expected gradient length* 33
13 Informatívnosť podľa A-optimality 35
14 Informatívnosť podľa *expected classification entropy* 38
15 Porovnanie stratégií (dáta Iris versicolor a Iris virginica) 40
16 Porovnanie stratégií (dáta Iris setosa a Iris versicolor) 43
17 Porovnanie stratégií (dáta Iris setosa a Iris virginica) 44
18 Porovnanie stratégií (dáta rakoviny prsníka) 47

Úvod

Dôležitou súčasťou moderných metód mnohorozmernej štatistiky a strojového učenia sú metódy automatickej klasifikácie dát do dvoch alebo viacerých skupín (kategórií). Kategóriou môže byť napríklad konkrétna diagnóza pacienta alebo zaradenie e-mailu medzi "spam". Klasifikátorom sa nazýva algoritmus, ktorým implementujeme klasifikáciu. Niekedy však klasifikátor môže znamenať aj matematickú funkciu získanú klasifikačným algoritmom, na základe ktorej je vytvorené klasifikačné pravidlo triedenia dát do kategórií. Na vytvorenie klasifikačného pravidla slúži "tréningová vzorka", čo je súbor dát, ktorý pre každý objekt obsahuje vektor vysvetľujúcich premenných a jeho známu, správnu klasifikáciu. Najčastejšie používanými metódami na vytvorenie klasifikačného pravidla sú lineárna diskriminačná analýza, metódy oporného bodu, neurónové siete, klasifikačné stromy a logistická regresia.

Je zrejmé, že čím máme k dispozícii väčší počet dát, tým presnejšie klasifikačné pravidlo môžeme vytvoriť. Dáta môžu byť síce ľahko získateľné, avšak na vytvorenie pravidla potrebujeme poznať aj ich kategórie, ktorých získanie je niekedy veľmi náročné. Vtedy sa snažíme označiť čo najmenej dát a na druhej strane, získať čo najlepšie klasifikačné pravidlo. Riešením v takýchto situáciách sú práve sekvenčné metódy klasifikácie dát, ktorým sa venujeme v tejto diplomovej práci.

Práca je rozdelená do dvoch hlavných kapitol. Prvá kapitola je zameraná na logistickú regresiu, konkrétne binárnu logistickú regresiu a na jej použitie v diskriminačnej analýze, ktorú aplikujeme na demonštrovanie a porovnanie sekvenčných metód konštrukcie rozhodovacieho pravidla na konkrétnych dátach. Okrem toho sa v prvej kapitole zameriavame aj na Fisherovu informačnú maticu, ktorá je okrem sekvenčných metód klasifikácie dát vo veľkej miere používaná aj v úzko prepojenej oblasti štatistiky, v optimálnom navrhovaní experimentov. Ako súvisí Fisherova informačná matica s optimálnym navrhovaním experimentov je opísané v prílohe 1.

V druhej kapitole sa nachádza úvod k sekvenčným metódam klasifikácie dát a motivácia ich použitia. Uvádzame niekoľko reálnych príkladov, kedy použitie sekvenčných metód je naozaj opodstatnené.

Menšia časť tejto kapitoly je venovaná scenárom výberu dát, ktoré sú potom použité na vytvorenie klasifikačného pravidla a v nosnej časti sme sa zamerali na stratégie výberu dát, ktoré určujú mieru informatívnosti dát. Každú stratégiu uvádzame všeobecne a aj pre binárnu logistickú regresiu. Porovnanie jednotlivých stratégií realizujeme na dvoch súboroch dát, kde okrem doteraz používaných stratégií porovnávame aj novú, nami navrhnutú stratégiu. Všetky výpočty boli realizované v programovacom jazyku "R" a uvedené sú v prílohe 2.

1 Logistická regresia

Regresné modely sa používajú na vyjadrenie vzťahu medzi závislou (vysvetľovanou) premennou Y a jednou alebo viacerými nezávislými (vysvetľujúcimi) premennými $\boldsymbol{x}$. V prípade diskrétnej vysvetľovanej premennej, čiže premennej nadobúdajúcej konečný počet hodnôt, sa najčastejšie používa model logistickej regresie. Pri opise logistickej regresie budeme vychádzať z [8]. Cieľom logistickej regresie, ako aj iných modelovacích štatistických techník, je identifikovať premenné, ktoré výrazne ovplyvňujú hodnotu vysvetľovanej premennej a na základe hodnôt týchto premenných predikovať hodnotu Y novozvoleného objektu.

Napríklad, ak chceme zistiť, či u pacienta možno očakávať pooperačné komplikácie ($Y = 1$) alebo nie ($Y = 0$), ak máme k dispozícii databázu testov a operačných výsledkov predchádzajúcich pacientov, môžeme použiť logistickú regresiu. Pomocou logistickej regresie a databázy zistíme, ktoré medicínske testy zo všetkých možných testov ovplyvňujú to, či nastali pooperačné komplikácie (identifikácia premenných $\boldsymbol{x}$) a takisto budeme vedieť predikovať pooperačné komplikácie (predikcia hodnoty Y) pre nového pacienta, ktorého výsledky týchto testov budeme poznať.

Hlavným dôvodom toho, že v prípade diskrétnej premennej Y nemôže byť použitá lineárna regresia je, že podmienená stredná hodnota $E(Y|\boldsymbol{x})$ v takomto prípade predstavuje pravdepodobnosť, čiže hodnotu z intervalu $\langle 0, 1\rangle$. Toto kritérium spĺňa napríklad logistická funkcia. Sú dva hlavné dôvody výberu práve tejto funkcie. Z matematického hľadiska je to výborne flexibilná a ľahko použiteľná funkcia a druhým dôvodom je dobrá interpretovateľnosť takto zvoleného modelu, resp. jeho parametrov.

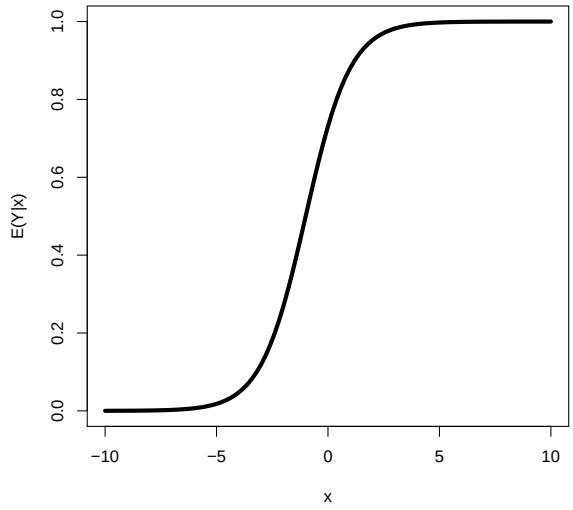
1.1 Binárna logistická regresia

V prípade binárnej logistickej regresie Y nadobúda dve možné hodnoty. Najčastejšie $Y = 1$ znamená nastatie javu a $Y = 0$ nenastatie javu. Pretože

$E(Y|\boldsymbol{x})$ v modeli logistickej regresie predstavuje pravdepodobnosť nastatia javu, budeme používať označenie $E(Y|\boldsymbol{x}) = \pi(\boldsymbol{x}, \boldsymbol{\beta})$. Model logistickej regresie potom vyzerá nasledovne:

$$\pi(\boldsymbol{x}, \boldsymbol{\beta}) = \frac{1}{1 + e^{-\boldsymbol{\beta}^T \boldsymbol{x}}}, \tag{1}$$

kde $\boldsymbol{x} = \begin{pmatrix} 1, & x_1, & x_2, & \cdots, & x_r \end{pmatrix}^T$ je vektor vysvetľujúcich premenných a $\boldsymbol{\beta} = \begin{pmatrix} \beta_0, & \beta_1, & \cdots, & \beta_r \end{pmatrix}^T$ je vektor parametrov modelu. V prípade jednej vysvetľujúcej premennej vidíme graf logistickej funkcie na obr. 1.



Obr. 1: Logistická funkcia s parametrom $\boldsymbol{\beta} = (1, 1)^T$

Keďže $\pi(\boldsymbol{x}, \boldsymbol{\beta})$ je stredná hodnota binárnej náhodnej premennej $Y|\boldsymbol{x}$, tak $\pi(\boldsymbol{x}, \boldsymbol{\beta})$ tiež predstavuje pravdepodobnosť nastatia javu, ktorý modeluje náhodná premenná Y pri hodnotách nezávislých premenných určených vektorom $\boldsymbol{x}$. Pravdepodobnosť, že jav nenastane je rovná $1 - \pi(\boldsymbol{x}, \boldsymbol{\beta})$.

Elementárnymi úpravami prevedieme logistický model na tvar

$$\ln \frac{\pi(\boldsymbol{x}, \boldsymbol{\beta})}{1 - \pi(\boldsymbol{x}, \boldsymbol{\beta})} = \boldsymbol{\beta}^T \boldsymbol{x}.$$

Výraz $\frac{\pi(\boldsymbol{x}, \boldsymbol{\beta})}{1 - \pi(\boldsymbol{x}, \boldsymbol{\beta})}$ sa nazýva **šanca** (angl. *odds*). Ak by napríklad výhra v lotérii predstavovala nastatie javu a jej pravdepodobnosť by bola 10%, tak šanca

na výhru je 1/9. Logaritmus šance sa nazýva **logit**. Význam tejto transformácie je, že logit má tvar lineárneho regresného modelu (lineárnosť v $\boldsymbol{x}$, nadobúda hodnoty z intervalu $(-\infty, \infty)$).

Ďalším podstatným rozdielom medzi lineárnou a logistickou regresiou je, že pri lineárnej regresii môžeme realizáciu Y zapísať nasledovne: $Y = E(Y|\boldsymbol{x}) + \varepsilon$, pričom ε obvykle pochádza z $N(0, \sigma^2)$, prípadne z iného spojitého rozdelenia nezávislého na $\boldsymbol{x}$. To však nie je prípad diskrétnej premennej Y. V takejto situácii síce realizáciu Y vyjadríme takisto ako $Y = E(Y|\boldsymbol{x}) + \varepsilon$, ale ε môže nadobudnúť len dve hodnoty. $Y = 1$ s pravdepodobnosťou $\pi(\boldsymbol{x}, \boldsymbol{\beta})$, teda $\varepsilon = 1 - \pi(\boldsymbol{x}, \boldsymbol{\beta})$ s pravdepodobnosťou $\pi(\boldsymbol{x}, \boldsymbol{\beta})$ a analogicky $\varepsilon = -\pi(\boldsymbol{x}, \boldsymbol{\beta})$ s pravdepodobnosťou $1 - \pi(\boldsymbol{x}, \boldsymbol{\beta})$. Čiže ϵ má rozdelenie so strednou hodnotou 0 a disperziou $\pi(\boldsymbol{x}, \boldsymbol{\beta})(1 - \pi(\boldsymbol{x}, \boldsymbol{\beta}))$.

1.2 Odhad parametrov

Pri teórii odhadu parametrov vychádzame z [10]. Predpokladajme, že máme k dispozícii n nezávislých pozorovaní $\{\boldsymbol{x}_i, y_i\}, i = 1, \ldots, n$. Na odhad parametrov $\boldsymbol{\beta}$ logistickej regresie sa používa metóda maximálnej vierohodnosti. Na aplikovanie tejto metódy potrebujeme najskôr zostrojiť vierohodnostnú funkciu, ktorá vyjadruje pravdepodobnosť nadobudnutia práve týchto n pozorovaní y_i pri daných hodnotách $\boldsymbol{x}_i$ ako funkciu $\boldsymbol{\beta}$. Pravdepodobnosť, že objekt, ktorého charakterizuje vektor premenných $\boldsymbol{x}_i$, nadobudne hodnotu $y_i \in \{0, 1\}$, sa rovná

$$p(y_i|\boldsymbol{x}_i, \boldsymbol{\beta}) = \pi(\boldsymbol{x}_i, \boldsymbol{\beta})^{y_i}(1 - \pi(\boldsymbol{x}_i, \boldsymbol{\beta}))^{1-y_i}.$$

Keďže jednotlivé pozorovania sú nezávislé, vierohodnostná funkcia vyzerá nasledovne

$$\begin{aligned} L(\boldsymbol{\beta}) &= \prod_{i=1}^{n} p(y_i|\boldsymbol{x}_i, \boldsymbol{\beta}) \\ &= \prod_{i=1}^{n} \pi(\boldsymbol{x}_i, \boldsymbol{\beta})^{y_i}(1 - \pi(\boldsymbol{x}_i, \boldsymbol{\beta}))^{1-y_i}. \end{aligned}$$

Princíp metódy maximálnej vierohodnosti spočíva v nájdení takého parametra $\boldsymbol{\beta}$, ktorý maximalizuje hodnotu vierohodnostnej funkcie, čo je ekvivalentné

nájdeniu parametra maximalizujúceho logaritmus tejto funkcie, s ktorou sa z matematického hľadiska ľahšie pracuje:

$$\begin{aligned}
\ell(\boldsymbol{\beta}) = \ln(L(\boldsymbol{\beta})) &= \sum_{i=1}^{n} \{y_i \ln \pi(\boldsymbol{x}_i, \boldsymbol{\beta}) + (1 - y_i) \ln(1 - \pi(\boldsymbol{x}_i, \boldsymbol{\beta}))\} \\
&= \sum_{i=1}^{n} \left\{ y_i \ln \frac{e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}}{1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}} + (1 - y_i) \ln \frac{1}{1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}} \right\} \\
&= \sum_{i=1}^{n} \left\{ y_i (\boldsymbol{\beta}^T \boldsymbol{x}_i - \ln(1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i})) + (1 - y_i)(-\ln(1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i})) \right\} \\
&= \sum_{i=1}^{n} \left\{ y_i \boldsymbol{\beta}^T \boldsymbol{x}_i - \ln(1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}) \right\}.
\end{aligned}$$

Parameter $\boldsymbol{\beta}$, ktorý maximalizuje $\ell(\boldsymbol{\beta})$ označíme $\hat{\boldsymbol{\beta}}$. Tento parameter $\hat{\boldsymbol{\beta}}$ potom musí spĺňať

$$\begin{aligned}
\dot{\ell}(\hat{\boldsymbol{\beta}}) = \left. \frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \right|_{\hat{\boldsymbol{\beta}}} &= \sum_{i=1}^{n} \left\{ y_i \boldsymbol{x}_i - \frac{e^{\hat{\boldsymbol{\beta}}^T \boldsymbol{x}_i}}{1 + e^{\hat{\boldsymbol{\beta}}^T \boldsymbol{x}_i}} \boldsymbol{x}_i \right\} \\
&= \sum_{i=1}^{n} \{y_i - \pi(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}})\} \boldsymbol{x}_i = 0,
\end{aligned}$$

čo vedie k vyriešeniu $r+1$ nelineárnych rovníc v $r+1$ logistických parametroch $\boldsymbol{\beta}$. Na vyriešenie sa dá použiť iteratívna verzia váženej metódy najmenších štvorcov (angl. *iteratively reweighted least-squares* (IRLS)), kde $(k + 1)$ -vý krok algoritmu vyzerá nasledovne:

$$\tilde{\boldsymbol{\beta}}^{(k+1)} = \tilde{\boldsymbol{\beta}}^{(k)} - \left(\ddot{\ell} \left(\tilde{\boldsymbol{\beta}}^{(k)} \right) \right)^{-1} \dot{\ell} \left(\tilde{\boldsymbol{\beta}}^{(k)} \right),$$

pričom ako štartovaciu hodnotu sa odporúča zvoliť (podľa [10]) $\tilde{\boldsymbol{\beta}}^{(0)} = 0$ a druhú deriváciu logaritmu vierohodnostnej funkcie vypočítame nasledovne:

$$\ddot{\ell}(\boldsymbol{\beta}) = \frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} = -\sum_{i=1}^{n} \boldsymbol{x}_i \frac{e^{\boldsymbol{\beta}^T \boldsymbol{x}_i} \boldsymbol{x}_i^T (1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}) - e^{\boldsymbol{\beta}^T \boldsymbol{x}_i} e^{\boldsymbol{\beta}^T \boldsymbol{x}_i} \boldsymbol{x}_i^T}{(1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i})^2}$$

$$= -\sum_{i=1}^{n} \frac{e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}}{1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}} \left(1 - \frac{e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}}{1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i^T \boldsymbol{x}_i}}\right) \boldsymbol{x}_i \boldsymbol{x}_i^T$$

$$= -\sum_{i=1}^{n} \pi(\boldsymbol{x}_i, \boldsymbol{\beta})(1 - \pi(\boldsymbol{x}_i, \boldsymbol{\beta})) \boldsymbol{x}_i \boldsymbol{x}_i^T$$

$$= -\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X}, \tag{2}$$

kde

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1, & \boldsymbol{x}_2, & \ldots, & \boldsymbol{x}_n \end{pmatrix}^T,$$

$$\boldsymbol{W} = \begin{pmatrix} \pi(\boldsymbol{x}_1, \boldsymbol{\beta})(1 - \pi(\boldsymbol{x}_1, \boldsymbol{\beta})) & 0 & \cdots & 0 \\ 0 & \pi(\boldsymbol{x}_2, \boldsymbol{\beta})(1 - \pi(\boldsymbol{x}_2, \boldsymbol{\beta})) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \pi(\boldsymbol{x}_n, \boldsymbol{\beta})(1 - \pi(\boldsymbol{x}_n, \boldsymbol{\beta})) \end{pmatrix}.$$

Maticu $-\ddot{\ell}(\boldsymbol{\beta})$ označme $I(\boldsymbol{\beta})$. Táto matica sa nazýva Fisherova informačná matica a je jej venovaná nasledujúca podkapitola.

1.3 Fisherova miera informácie

Pri odvodení Fisherovej informačnej matice a Cramérovej-Raovej nerovnosti sme vychádzali z [14]. Fisherova miera informácie je definovaná ako variancia *skóre*, ktoré sa vypočíta ako derivácia logaritmu vierohodnostnej funkcie $\ell(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{\beta})$, kde $\boldsymbol{Y}$ je vektor príslušných kategórií dát s vektormi "čŕt" $\boldsymbol{X}$. Môžeme ju chápať ako množstvo obsiahnutej informácie, ktoré nesie náhodná premenná $\boldsymbol{Y}|\boldsymbol{X}$ o neznámom parametri $\boldsymbol{\beta}$, od ktorého závisí pravdepodobnosť tejto náhodnej premennej. Dá sa ukázať, že stredná hodnota skóre je rovná 0. Preto Fisherovu mieru informácie môžeme vyjadriť takto:

$$(I(\boldsymbol{\beta}))_{ij} = E\left[\frac{\partial \ell(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}_i} \frac{\partial \ell(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j} \middle| \boldsymbol{\beta}\right].$$

Ak je logaritmus vierohodnostnej funkcie dvakrát diferencovateľný podľa $\boldsymbol{\beta}$, čo v prípade logistickej regresie je, Fisherova informácia môže byť vyjadrená ako záporne vzatá stredná hodnota druhej derivácie logaritmu vierohodnostnej funkcie vzhľadom na parameter $\boldsymbol{\beta}$:

$$(I(\boldsymbol{\beta}))_{ij} = -E\left[\frac{\partial^2 \ell(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}_i \partial \boldsymbol{\beta}_j}\middle| \boldsymbol{\beta}\right].$$

Využijúc vzťah (2), dostávame

$$\begin{aligned}(I(\boldsymbol{\beta}))_{ij} &= -E\left[-(\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})_{ij}\right] \\ &= (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})_{ij}.\end{aligned}$$

1.3.1 Cramérova-Raova nerovnosť

Cramérova-Raova hranica nám udáva hranicu toho, koľko informácie je obsiahnutej v dátach o neznámom parametri $\boldsymbol{\beta}$. Konkrétne, Cramérova-Raova nerovnosť určuje minimálnu varianciu nevychýleného odhadu neznámeho parametra $\boldsymbol{\beta}$.

V jednorozmernom prípade (jeden parameter β) využitím *Cramérovej-Raovej nerovnosti*, podľa ktorej pre ľubovoľný nevychýlený odhad platí, že jeho variancia je väčšia alebo rovná ako inverzia Fisherovej informačnej matice, dostaneme:

$$Var(\hat{\beta}) \geq \frac{1}{I(\beta)}.$$

Ak variancia nejakého nevychýleného odhadu dosahuje túto dolnú hranicu, tento odhad sa nazýva efektívny. Podľa [7] je odhad metódou maximálnej vierohodnosti asymptoticky efektívny, čiže $Var(\hat{\beta}) \doteq \frac{1}{I(\beta)}$.

Vo viacrozmernom prípade nemožno použiť predchádzajúce vzťahy, pretože Fisherova informačná matica v tamto prípade nie je skalárom, ale štvorcovou maticou rozmeru, ktorý závisí od počtu neznámych parametrov. V našom prípade $(r+1) \times (r+1)$. Nech $\boldsymbol{Y}$ je vektor príslušných kategórií dát s vektormi "čŕt" $\boldsymbol{X}$ a nech $\boldsymbol{T}(\mathbb{X}) = (T_1(\mathbb{X}), T_2(\mathbb{X}), \ldots, T_N(\mathbb{X}))$, kde $\mathbb{X} = \mathbb{X}(\boldsymbol{X}, \boldsymbol{Y})$

je odhad ľubovoľného vektora funkcií parametrov. Očakávanú hodnotu tohto vektora $E(\boldsymbol{T}(\mathbb{X}))$ označme $\boldsymbol{\psi}(\boldsymbol{\beta})$. Po splnení podmienok regularity (pozri [14]) Cramérova-Raova nerovnosť udáva ohraničenie pre kovariančnú maticu $cov(\boldsymbol{T}(\mathbb{X}))$:

$$cov(\boldsymbol{T}(\mathbb{X})) \succeq \frac{\partial \boldsymbol{\psi}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} I^{-1}(\boldsymbol{\beta}) \left(\frac{\partial \boldsymbol{\psi}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \right)^T,$$

kde maticové porovnanie $A \succeq B$ znamená, že $A - B$ je pozitívne semidefinitná matica. Všimnime si, že ak $\boldsymbol{T}(\mathbb{X})$ je nevychýlený odhad vektora parametrov $\boldsymbol{\beta}$ (čiže $E(\boldsymbol{T}(\mathbb{X})) = \psi(\boldsymbol{\beta}) = \boldsymbol{\beta}$), tak Cramérova-Raova nerovnosť sa redukuje na tvar

$$cov(\boldsymbol{T}(\mathbb{X})) \succeq I^{-1}(\boldsymbol{\beta}).$$

Keďže odhad metódou maximálnej vierohodnosti je asymptoticky nevychýlený, dá sa očakávať približná platnosť tvrdenia

$$cov(\hat{\boldsymbol{\beta}}) \succeq I^{-1}(\boldsymbol{\beta}) = (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1}.$$

Využijúc asymptotickú efektívnosť odhadu metódou maximálnej vierohodnosti

$$cov(\hat{\boldsymbol{\beta}}) \doteq (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1}.$$

Presnejšia analýza vzťahu kovariančnej matice odhadu metódou maximálnej vierohodnosti a informačnej matice presahuje rámec tejto práce.

1.4 Logistická diskriminačná analýza

Logistickú regresiu je možné použiť aj ako metódu diskriminačnej analýzy. Vychádzajúc z [10] si ukážeme spôsob jej aplikovania v binárnom prípade, čiže v prípade dvoch kategórií. Pomocou premennej Y je identifikovaná kategória (Π_1, resp. Π_2), do ktorej je zaradený údaj s vektorom "čŕt" $\boldsymbol{x}$ nasledovne:

$$Y = \begin{cases} 0 & \text{ak } \boldsymbol{x} \in \Pi_1 \\ 1 & \text{ak } \boldsymbol{x} \in \Pi_2. \end{cases}$$

Budeme požívať nasledujúce značenie: $p_i(\boldsymbol{x}, \boldsymbol{\beta})$ - pravdepodobnosť, že údaj s vektorom "čŕt" $\boldsymbol{x}$ patrí do kategórie Π_i na základe modelu s parametrom $\boldsymbol{\beta}$. [1]

Pomocou parametra $\hat{\boldsymbol{\beta}}$ získaného metódou maximálnej vierohodnosti (pozri kapitolu 1.2) je vytvorená lineárna diskriminačná funkcia

$$\hat{L}(\boldsymbol{x}) = \hat{\boldsymbol{\beta}}^T \boldsymbol{x}.$$

Klasifikačné pravidlo logistickej diskriminačnej analýzy definujeme nasledovne:

$$\text{ak } \hat{L}(\boldsymbol{x}) > 0, \boldsymbol{x} \text{ je zaradené do kategórie } \Pi_2,$$

inak je zaradené do kategórie Π_1.

Ekvivalentnou klasifikačnou procedúrou je použitie funkcie $\hat{L}(\boldsymbol{x})$ na odhad pravdepodobnosti $p_2(\boldsymbol{x}, \boldsymbol{\beta})$ vo vzťahu (1):

$$\hat{p_2}(\boldsymbol{x}, \hat{\boldsymbol{\beta}}) = \frac{1}{1 + e^{-\hat{L}(\boldsymbol{x})}}.$$

Potom, ak

$$\hat{p_2}(\boldsymbol{x}, \hat{\boldsymbol{\beta}}) > \text{ stanovená hranica}, \boldsymbol{x} \text{ je zaradené do kategórie } \Pi_2,$$

inak je zaradené do kategórie Π_1. Väčšinou je hranicou zvolená hodnota 0.5, ktorú budeme používať aj my.

Ešte pred samotnou aplikáciou sekvenčných metód klasifikácie dát pomocou logistickej diskriminačnej analýzy vysvetlíme niektoré v tomto texte často používané pojmy:

- trénovanie modelu - pod pojmom trénovanie modelu chápeme získanie odhadu parametrov $\hat{\boldsymbol{\beta}}$ na základe trénovacej vzorky,

- trénovacia vzorka - súbor dát, pre ktoré poznáme ich skutočnú kategóriu a sú použité na natrénovanie modelu,

[1] V prípade binárnej logistickej regresie nastatím javu chápeme zaradanie objektu do kategórie Π_2, čiže $\pi(\boldsymbol{x}, \boldsymbol{\beta}) = p_2(\boldsymbol{x}, \boldsymbol{\beta})$.

- vektor "čŕt" - vektor vysvetľujúcich premenných $\boldsymbol{x}$ pre konkrétny údaj,

- pretrénovanie modelu (angl. *retraining*) - pod slovenským výrazom pretrénovanie modelu sa niekedy myslí vytvorenie štatistického modelu s veľkým počtom parametrov vzhľadom na počet dát, ktorý namiesto vzťahu premenných popisuje náhodnú chybu modelu (angl. *overfitting*). My však pod pojmom pretrénovanie modelu budeme chápať opätovné vytvorenie modelu (trénovanie modelu) na základe novej trénovacej vzorky.

2 Sekvenčné metódy konštrukcie rozhodovacieho pravidla

Sekvenčná klasifikácia dát (angl. *active learning*) je súčasťou teórie strojového učenia (angl. *machine learning*), ktoré je zase súčasťou vednej disciplíny umelá inteligencia. Sekvenčnú klasifikáciu dát je však taktiež možné považovať za súčasť štatistického navrhovania experimentov. Hlavnou myšlienkou sekvenčných metód klasifikácie dát je povoliť učiacemu sa algoritmu vyberať si dáta, na ktorých bude model trénovaný s cieľom dosiahnuť lepší výsledok s použitím menšieho počtu dát. V [17] sa nachádza všeobecný prehľad sekvenčných metód klasifikácie dát a takisto poskytuje prehľad literatúry zaoberajúcej sa touto problematikou.

V niektorých prípadoch obdržanie dát aj s ich zatriedením nie je náročné, napríklad pri vytváraní klasifikačného algoritmu, pomocou ktorého chceme dokázať roztriediť e-mailové správy na "spam" a ostatné. V tomto prípade môžeme trénovacie dáta, na základe ktorých budeme vytvárať klasifikačné pravidlo, získať veľmi jednoducho a to od používateľov, ktorí nechcenú správu jednoducho označia ako "spam". V spomenutom prípade sú dáta a ich kategórie ľahko dostupné. Avšak existuje veľa úloh, pri ktorých označenie (zatriedenie) dát je časovo, finančne alebo inak náročné. Práve v takýchto prípadoch, kde nezatriedené dáta sú síce ľahko dosiahnuteľné, ale ich označenie do príslušnej kategórie je nákladné, sa používajú sekvenčné metódy klasifikácie dát. Príkladom použitia sekvenčných metód klasifikácie dát je vytvorenie algoritmu na rozpoznávanie reči. Získanie takejto trénovacej vzorky je extrémne časovo náročné, aj keď samotné nahrávky reči sú väčšinou ľahko dostupné, ale rozpoznanie (označenie) toho, čo sa na jednotlivých nahrávkach nachádza (aké slová, resp. slabiky) si vyžaduje veľa času a niekedy aj pomoc jazykovedca. Rozpoznávanie na úrovni slov môže trvať aj 10-krát dlhšie ako samotná audio nahrávka [21]. Spomedzi mnoho ďalších môžeme spomenúť ešte algoritmus na klasifikáciu dokumentov alebo webových stránok. Nezatriedené dokumenty môžeme získať

veľmi jednoducho, ale určenie, či daný dokument (napr. článok) alebo webová stránka patrí alebo nepatrí do danej klasifikačnej triedy, si vyžaduje preštudovanie daného dokumentu, čo môže byť veľmi časovo náročné. S finančnou náročnosťou označovania dát sa často môžeme stretnúť v medicínskej oblasti. Na určenie kategórie trénovacích dát totiž neraz potrebujeme vykonať nákladný experiment.

Sekvenčné metódy konštrukcie rozhodovacieho pravidla sa snažia predchádzať zbytočnému označovaniu dát, o ktorých sa predpokladá, že nepomôžu výrazne zlepšiť klasifikačné pravidlo. Naopak, žiada označenie len tých dát, ktoré sú pre daný klasifikačný algoritmus kritické. Takýmto spôsobom je do experimentu zahrnutá len časť zo všetkých dostupných dát a pomocou nich je vytvorené efektívne klasifikačné pravidlo. Sekvenčné metódy klasifikácie dát teda znižujú náklady na daný experiment.

Pri výbere najinformatívnejších dát pre doposiaľ natrénovaný model rozlišujeme viacero scenárov a viacero stratégií výberu.

2.1 Scenáre výberu dát na zatriedenie

Po každom vytvorení modelu je vybratý údaj, ktorý model považuje za najinformatívnejší. Prvý vytvorený model budeme nazývať inicializačný a vytvorenie každého ďalšieho nového modelu zase pretrénovanie modelu. Existuje viacero spôsobov nazývaných scenáre, ktorými sa daný údaj vyberie. Článok [17] rozlišuje tri základné scenáre

- *membership query synthesis*,
- *stream-based selective sampling*,
- *pool-based sampling*.

2.1.1 Membership query synthesis

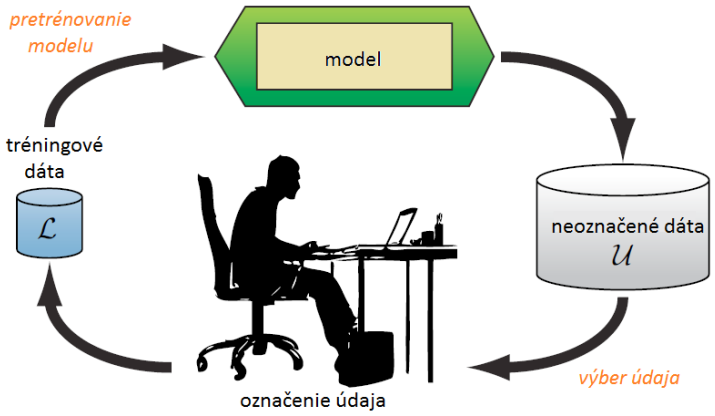
Jedným z prvých scenárov sekvenčných metod, ktorý bol skúmaný, je *membership query synthesis* ([2]). Pri tomto druhu scenára model požaduje označenie údaja, ktorý je modelom nanovo vygenerovaný. V niektorých prípadoch je takýto scenár veľmi užitočný, avšak v niektorých je jeho použitie nemožné. Napríklad, ak by sme chceli natrénovať model na rozpoznávanie písmen, model môže požiadať o určenie symbolu, ktorý nebude predstavovať žiadne písmeno abecedy. Na tento problém upozornili aj autori [11]. Podobný problém nastáva aj v modeloch rozpoznávania reči.

2.1.2 Stream-based selective sampling

Scenár *stream-based selective* sampling bol uvedený v [4]. Hlavným predpokladom použitia tohto scenára je, že získanie neoznačeného údaja nestojí nič, resp. je veľmi lacné. Po pretrénovaní modelu sa náhodne vyberie údaj, ktorý doposiaľ nie je označený a model rozhodne, či vyžiada jeho označenie alebo nie. Toto rozhodnutie môže byť vykonané na základe rôznych kritérií. Napríklad, ak ku každému údaju je určená jeho informatívnosť pomocou funkcie informatívnosti, ktorá predstavuje akúsi mieru zaujímavosti poznania kategórie daného údaja pre doterajší model, tak sa určí minimálna hranica hodnoty tejto funkcie, ktorá bude kritériom rozhodnutia vyžiadania si označenia. Ak hodnota funkcie informatívnosti bude nad touto hranicou, bude vyžiadané určenie kategórie takéhoto údaja a naopak, ak bude pod hranicou, neurčí sa zatriedenie údaja, pretože pre súčasný model nie je až také podstatné poznať toto zatriedenie a náhodne sa vyberie iný údaj. Ďalšou možnosťou rozhodnutia pre zatriedenie alebo nezatriedenie vybratého údaja je určenie oblasti, ktorá je stále nejednoznačná z hľadiska zatriedenia pre doterajší model. Ak náhodne vybratý údaj bude pochádzať z tejto oblasti nejednoznačnosti, vyžiada sa jeho označenie, v opačnom prípade sa vyberie nový údaj.

2.1.3 Pool-based sampling

Pre veľa klasifikačných problémov môžeme získať neoznačené dáta naraz, čiže nezískavame ich postupne s časovým odstupom. V takom prípade spôsobom, ktorým vyberieme údaj na označenie, môže byť scenár *pool-based sampling*, ktorým sa ako prví zaoberali autori v [12]. Pri tomto scenári sa na určenie kategórie vyberá vždy údaj, ktorý je pre súčasný model najinformatívnejší. Čiže pre všetky dáta z doteraz nezatriedených dát U, sa určí ich informatívnosť a vyberie sa ten, ktorého hodnota informatívnosti je najvyššia. Po pretrénovaní modelu musí byť opäť určená informatívnosť pre každý údaj z U.



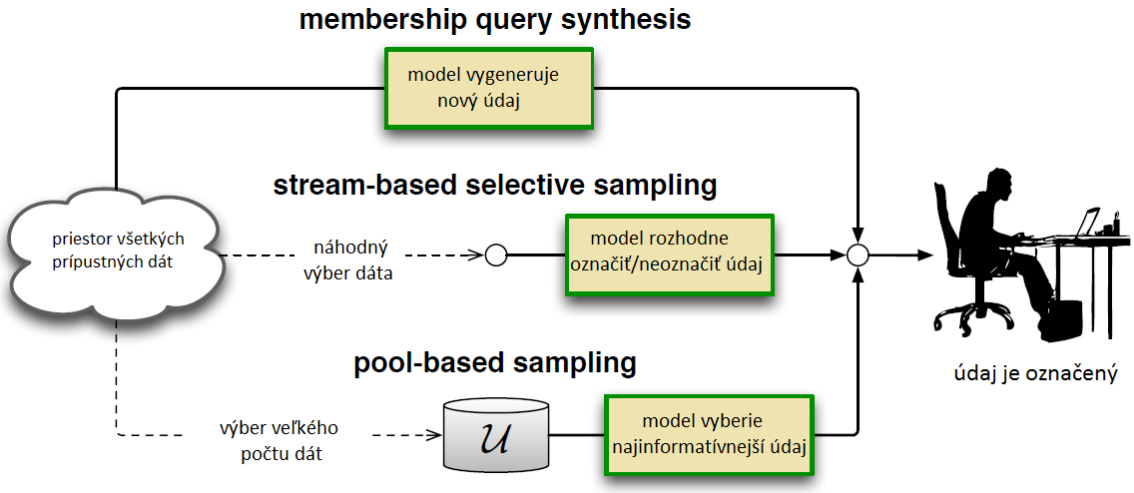
Obr. 2: Cyklus scenára pool based sampling. Obrázok pochádza z článku [17].

Na nasledujúcej schéme (obr. 2) vidíme všeobecný priebeh trénovania klasifikačného algoritmu s použitím sekvenčných metód a scenára *pool-based sampling*. Na začiatku máme k dispozícii súbor neoznačených dát U (angl. *unlabeled pool*). V prvom kroku náhodne vyberieme niekoľko z nich na vytvorenie inicializačného modelu. Podľa stavu tohto modelu sa vyberie na označenie ten údaj (alebo viacero dát), ktorý bude model požadovať, pretože ho bude považovať za najinformatívnejší. Tento údaj sa zaradí medzi označené dáta L (angl. *labeled pool*). Následne je model pretrénovaný a celý proces od vybratia údaja požadovaného modelom sa opakuje, až kým nedosiahneme maximálny počet dát, ktoré môžu byť označené, cenové ohraničenie experimentu alebo iné

kritérium ukončenia experimentu.

Hlavný rozdiel scenárov *stream-based sampling* a *pool-based sampling* je, že pri *stream-based sampling* sa určuje informatívnosť individuálne, zatiaľ čo pri *pool-based sampling* sa vždy vyhodnotí informatívnosť všetkých neoznačených dát pred tým, ako sa určí požiadavka na zatriedenie. Je zrejmé, že *pool-based sampling* vyberá vždy údaj, ktorý je informatívnejší ako *stream-based sampling* (alebo rovnako informatívny), avšak v niektorých prípadoch je výhodnejšie použiť práve *stream-based sampling*. Konkrétne, ak napríklad je obmedzená pamäť alebo výpočtový výkon, napríklad pri mobilných telefónoch.

Princíp všetkých troch scenárov vidíme zhrnutý na obr. 3.



Obr. 3: Scenáre výberu dát. Obrázok je z článku [17].

2.2 Stratégie výberu dát

Ako sme už spomenuli, pri trénovaní klasifikačného algoritmu s použitím sekvenčných metód, sa podľa stavu doteraz vytvoreného modelu vyberie najinformatívnejší údaj, ktorého skutočnú kategóriu následne zistíme. Čo však určuje mieru informácie, ktorú nám poskytne označenie údaja? V tejto kapitole sa pozrieme na základné stratégie výberu dát, ktoré nám určujú práve spomenutú informatívnosť. Budeme používať nasledovné značenie:
c - počet kategórií,
U - množina neoznačených dát $\boldsymbol{x}$,
L - množina označených dát $\boldsymbol{x}$,
x^* - najinformatívnejší údaj.

Ku každej stratégii uvedieme výpočet vo všeobecnom modeli a zároveň v modeli binárnej logistickej regresie.

2.2.1 Uncertainty sampling

Asi najjednoduchšou a najpoužívanejšou stratégiou je metóda maximálnej neurčitosti (angl. *uncertainty sampling*), ktorá bola prvýkrát uvedená v [12]. Podstatou tejto stratégie je vybratie na označenie takého údaja, o ktorého kategórii si je doposiaľ zostavený model najmenej istý. V rámci stratégie *uncertainty sampling* uvedieme konkrétne tri metódy a to *least confidence*, *margin* a *token entropy*, ktoré sú prezentované taktiež v [17].

- Least confidence (LC)
  Pri metóde *least confidence* sa vyžiada označenie toho údaja z doposiaľ nezatriedených dát, ktorého pravdepodobnosť najpravdepodobnejšej kategórie je najnižšia. Napríklad, ak by sme v modeli mali len dve kategórie a niektorý z údajov by s pravdepodobnosťou 50% patril do jednej a teda aj druhej kategórie, bude vybratý práve tento údaj na zistenie skutočnej

kategórie. Matematický zápis metódy *least confidence* vyzerá nasledovne:

$$\boldsymbol{x}^*_{LC} = \arg\min_{\boldsymbol{x}\in U} p_{\hat{y}}(\boldsymbol{x}, \boldsymbol{\beta}),$$

kde $\hat{y} = \arg\max_y p_y(\boldsymbol{x}, \boldsymbol{\beta})$, čiže kategória, do ktorej bude $\boldsymbol{x}$ zaradené s najväčšou pravdepodobnosťou v modeli s daným parametrom $\boldsymbol{\beta}$.
v prípade binárnej logistickej regresie:

$$\boldsymbol{x}^*_{LC} = \arg\min_{\boldsymbol{x}\in U} \left(\max \left\{ \frac{1}{1+e^{-\boldsymbol{\beta}^T\boldsymbol{x}}}, 1 - \frac{1}{1+e^{-\boldsymbol{\beta}^T\boldsymbol{x}}} \right\} \right).$$

- Margin (M)

  Ako sme si mohli všimnúť, metóda *least confidence* berie do úvahy len jednu (najviac pravdepodobnú) kategóriu a pravdepodobnosti, že údaj patrí do iných kategórií, neberie do úvahy. Práve preto metóda *margin* je upravením metódy *least confidence*. *Margin* vyberie na označenie ten údaj, ktorého rozdiel pravdepodobností dvoch najpravdepodobnejších kategórií je minimálny.

$$\boldsymbol{x}^*_{M} = \arg\min_{\boldsymbol{x}\in U} (p_{\hat{y}_1}(\boldsymbol{x}, \boldsymbol{\beta}) - p_{\hat{y}_2}(\boldsymbol{x}, \boldsymbol{\beta})),$$

  - $\hat{y}_1(\boldsymbol{x}, \boldsymbol{\beta})$ - kategória, do ktorej bude $\boldsymbol{x}$ zaradené s najväčšou pravdepodobnosťou v modeli s daným parametrom $\boldsymbol{\beta}$,

  - $\hat{y}_2(\boldsymbol{x}, \boldsymbol{\beta})$ - kategória, do ktorej bude $\boldsymbol{x}$ zaradené s druhou najväčšou pravdepodobnosťou v modeli s daným parametrom $\boldsymbol{\beta}$.

Rozdielnosť *least confidence* a *margin* si demonštrujeme na jednoduchom príklade. Predpokladajme, že dáta triedime do troch kategórií. Ďalej predpokladajme, že vyberáme informatívnejší údaj len z dvoch doteraz nezatriedených údajov. Prvý údaj má na základe doteraz vytvoreného modelu pravdepodobnosti, že bude zaradený do jednotlivých kategórií nasledovné: $p_1(\boldsymbol{x}, \hat{\boldsymbol{\beta}}) = 48\%$, $p_2(\boldsymbol{x}, \hat{\boldsymbol{\beta}}) = p_3(\boldsymbol{x}, \hat{\boldsymbol{\beta}}) = 26\%$ a druhý údaj: $p_1(\boldsymbol{x}, \hat{\boldsymbol{\beta}}) = 50\%$, $p_2(\boldsymbol{x}, \hat{\boldsymbol{\beta}}) = 49\%$ a $p_3(\boldsymbol{x}, \hat{\boldsymbol{\beta}}) = 1\%$. Metóda *least confidence* by vybrala na označenie prvý údaj, avšak jeho kategóriu by sme

mohli pokladať za celkom istú, pretože pravdepodobnosť, že patrí do prvej kategórie je relatívne vysoká oproti ostatným dvom pravdepodobnostiam. Zatiaľ čo pri druhom údaji to tvrdiť nemôžeme, pretože prvá aj druhá kategória tohto údaja majú porovnateľné pravdepodobnosti. Metódou *margin* by bol vybratý práve tento druhý údaj na určenie jeho skutočnej kategórie, lebo pri prvom údaji je rozdiel medzi dvomi najpravdepodobnejšími kategóriami 22% a pri druhom len 1%.

V prípade binárnej (logistickej) regresie sa metóda *margin* redukuje na metódu *least confidence*:

$$\begin{aligned}
\boldsymbol{x}_M^* &= \arg\min_{\boldsymbol{x}\in U}(p_{\hat{y}_1}(\boldsymbol{x},\boldsymbol{\beta}) - p_{\hat{y}_2}(\boldsymbol{x},\boldsymbol{\beta})) \\
&= \arg\min_{\boldsymbol{x}\in U}(p_{\hat{y}_1}(\boldsymbol{x},\boldsymbol{\beta}) - (1 - p_{\hat{y}_1}(\boldsymbol{x},\boldsymbol{\beta}))) \\
&= \arg\min_{\boldsymbol{x}\in U} p_{\hat{y}_1}(\boldsymbol{x},\boldsymbol{\beta}) \\
&= \arg\min_{\boldsymbol{x}\in U}\left(\max\left\{\frac{1}{1+e^{-\boldsymbol{\beta}^T\boldsymbol{x}}}, 1 - \frac{1}{1+e^{-\boldsymbol{\beta}^T\boldsymbol{x}}}\right\}\right).
\end{aligned}$$

- Token entropy (TE)

  Metóda *token entropy* neberie do úvahy iba dve najpravdepodobnejšie kategórie ako metóda *margin*, ale zohľadňuje pravdepodobnosti všetkých kategórií daného údaja pomocou entropie. Za najinformatívnejší volí údaj, ktorého entropia (neistota) príslušnosti do kategórie

  $$-\sum_{i=1}^{c} p_i(\boldsymbol{x},\boldsymbol{\beta}) \ln p_i(\boldsymbol{x},\boldsymbol{\beta}),$$

  je najvyššia:

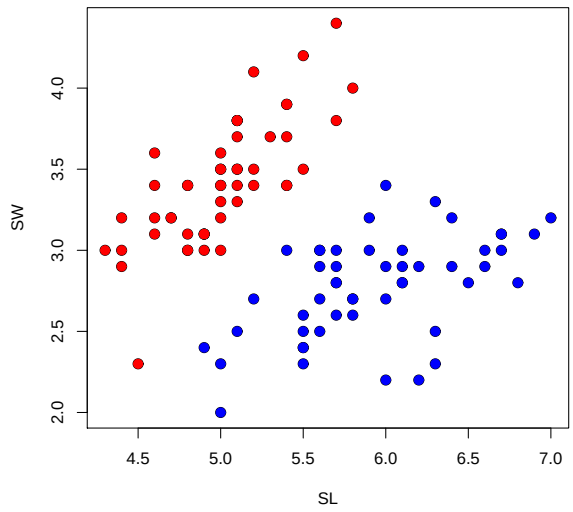
  $$\boldsymbol{x}_{TE}^* = \arg\max_{\boldsymbol{x}\in U}\left(-\sum_{i=1}^{c} p_i(\boldsymbol{x},\boldsymbol{\beta}) \ln p_i(\boldsymbol{x},\boldsymbol{\beta})\right).$$

  V prípade binárnej logistickej regresie:

  $$\boldsymbol{x}_{TE}^* = \arg\max_{\boldsymbol{x}\in U}\left(-\sum_{i=1}^{2} p_i(\boldsymbol{x},\boldsymbol{\beta}) \ln p_i(\boldsymbol{x},\boldsymbol{\beta})\right).$$

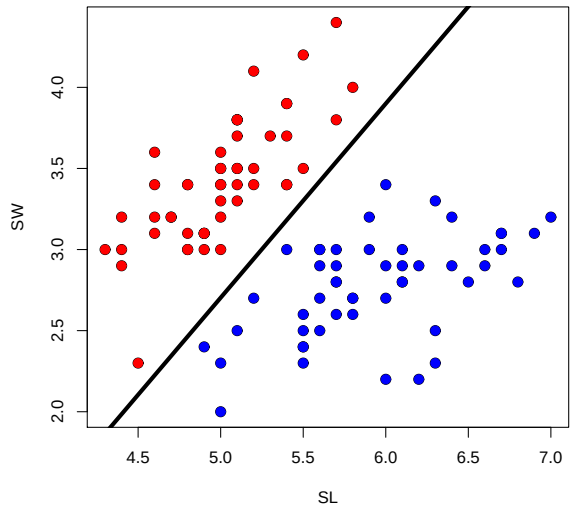
  Takisto ako metóda *margin* je v prípade dvoch kategórií podľa [9] ekvivalentná metóde *least confidence*.

Výber dát metódou *least confidence* si ukážeme aj graficky na reálnom príklade. Dáta (dostupné v [6]), ktoré použijeme, popisujú kvety kosatcov (angl. *Iris*), konkrétne ku každému kvetu kosatca je zaznamenaná dĺžka a šírka kališného lístka, dĺžka a šírka okvetného lístka (všetko v centimetroch) a zaradenie daného kvetu do jedného z troch druhov kosatcov - Iris setosa, Iris virginica, Iris versicolor. Z každého druhu sa v danom súbore dát nachádza 50 kvetov, čiže spolu máme 150 údajov. Pretože budeme používať binárnu logistickú regresiu, zvolíme si len dva druhy kosatcov - Iris setosa a Iris versicolor. Kvôli grafickej vizualizácii jednotlivých stratégií budeme tento model binárnej logistickej regresie zostavovať len na základe dvoch premenných a to dĺžky a šírky kališného lístka (angl. *sepal*). Tieto testovacie dáta sú zobrazené na obr. 4. Červenou farbou sú zaznačené kvety druhu Iris setosa a modrou farbou Iris virginica. Na x-vej osi je nanesená dĺžka kališného lístka (SL) a na y-vej šírka kališného lístka (SW).



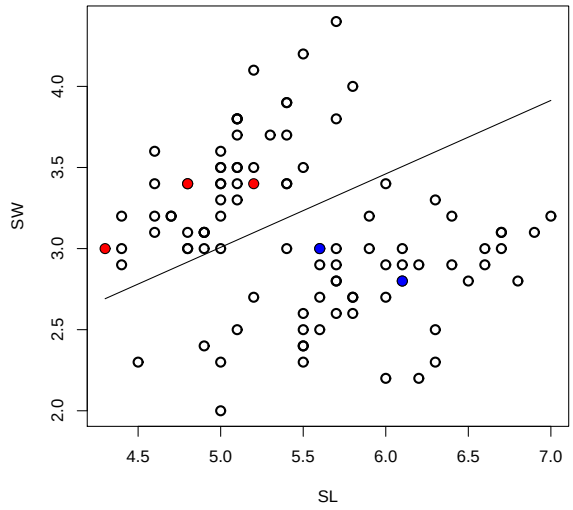
Obr. 4: Dáta dvoch druhov kosatcov (Iris setosa červenou, Iris virginica modrou) znázornené v závislosti od dĺžky (SL) a šírky (SW) kališného lístka.

Z grafu vidíme, že na základe hodnôt týchto premenných sú zobrazené dva druhy kosatcov ľahko odlíšiteľné. Dáta rozdelíme aplikovaním logistickej diskriminačnej analýzy a to tak, že do grafu zobrazíme deliacu priamku, ktorá predstavuje také kombinácie premenných (SL a SW), že pravdepodobnosť, že kvet s takýmito hodnotami patrí do druhu Iris setosa, resp. Iris virginica je presne 50%. Túto priamku nazveme diskriminačná hranica. Diskriminačná hranica predstavuje vlastne také hodnoty $\boldsymbol{x}$, pre ktoré je lineárna diskriminačná funkcia $\hat{L}(\boldsymbol{x})$ rovná 0 (obr. 5).



Obr. 5: Rozdelenie druhov Iris setosa a Iris virginica pomocou lineárnej diskriminačnej funkcie modelu, pre ktorý je známa kategória každého z dát.

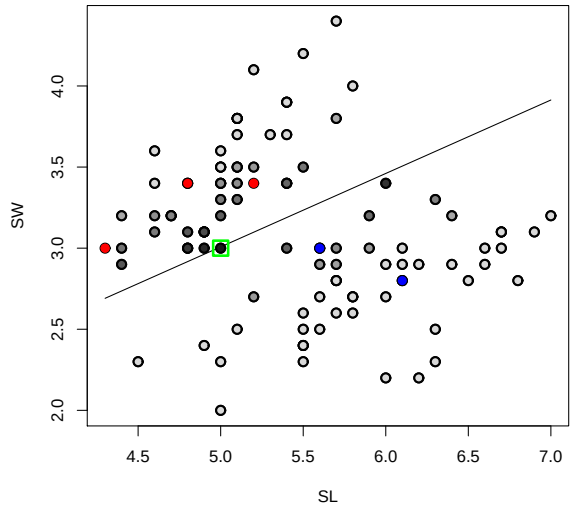
Aby sme mohli aplikovať stratégie výberu dát, budeme predpokladať, že na začiatku nepoznáme skutočné kategórie dát. V prvom kroku náhodne vyberieme malú časť kvetov, ktorých druh zistíme a na základe týchto dát vytvoríme incializačný model logistickej regresie (obr. 6). Dáta, ktorých skutočné kategórie poznáme, sú vyznačené farebne (červenou, resp. modrou farbou), ostatné sú na grafe znázornené prázdnym krúžkom.



Obr. 6: Lineárna diskriminačná hranica inicializačného modelu, ktorý vznikol na základe piatich náhodne vybratých dát, ktoré sú vyznačené farebne. Kategórie ostatných dát reprezentovaných prázdnym krúžkom sú pre model neznáme.

Ďalším krokom je aplikovanie metódy *least confidence*, pomocou ktorej zistíme, ktorý údaj vyberieme na zistenie jeho druhu, pretože je pre inicializačný model dôležité poznať jeho kategóriu. Pri výbere najinformatívnejšieho údaja budeme používať scenár *pool-based sampling*, čiže zistíme informatívnosť každého z doposiaľ neoznačených údajov. Za najinformatívnejší zvolíme ten, ktorého hodnota informatívnosti podľa danej stratégie bude najvyššia. Na nasledujúcom obrázku (obr. 7) je znázornený práve tento údaj v zelenom rámčeku. Okrem tohto najinformatívnejšieho údaja sú v grafe vyznačené všetky doposiaľ nezatriedené dáta podľa hodnoty ich informatívnosti pomocou rôznych úrovní sivej. Čím je údaj znázornený tmavšou farbou, tým je jeho informatívnosť pre súčasný model vyššia. Ako môžeme vidieť, najinformatívnejší údaj leží na deliacej priamke a bol vybratý preto, lebo jeho kategória je najmenej istá, pretože je približne 50% šanca, že bude zaradený medzi Iris setosa a takisto 50% šanca, že bude patriť medzi Iris virginica. Ďalej si môžeme všimnúť,

že informatívnosť dát sa so zväčšujúcou vzdialenosťou od deliacej priamky zmenšuje.

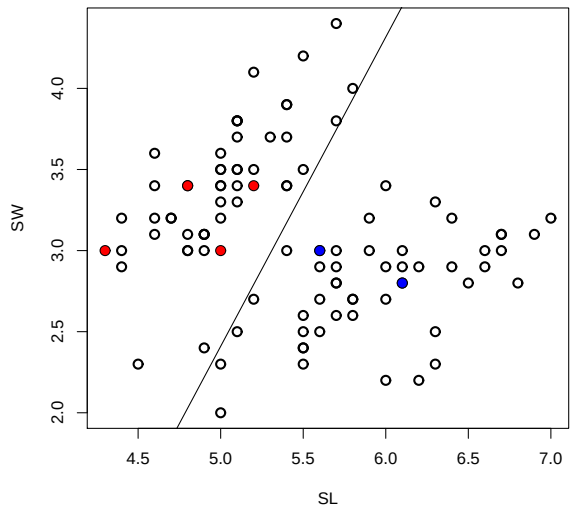


Obr. 7: Informatívnosť dát podľa stratégie *least confidence* pre inicializačný model. Miera informatívnosti jednotlivých údajov je vyjadrená intenzitou farby - čím tmavšia farba, tým väčšia informatívnosť. Najinformatívnejší údaj je zobrazený v zelenom rámčeku.

Nasleduje určenie druhu kvetu, ktorý je najinformatívnejší a pretrénovanie celého modelu na dátach, ktoré zahŕňajú doposiaľ označené dáta a tento najinformatívnejší údaj. Bolo zistené, že údaj patrí v skutočnosti do druhu Iris setosa (červený). Výsledok pretrénovania modelu môžeme vidieť na obr. 8.

Celý proces sa následne opakuje - zistí sa informatívnosť jednotlivých doposiaľ nezatriedených údajov a určí sa najinformatívnejší (obr. 9), pozrieme sa na jeho skutočnú kategóriu a model pretrénujeme.

Po tom, ako je model 5-krát pretrénovaný (obr. 10) (do trénovacej vzorky je postupne pridaných 5 ďalších bodov), sa už deliaca priamka prakticky nelíši od deliacej priamky modelu, ktorý bol natrénovaný na všetkých dátach. Stačilo teda, aby sme celkovo označili 10 dát (10 % z celkového počtu dát) a získali sme presne také rozdelenie kvetov na jednotlivé druhy, ako by sme získali pomocou modelu natrénovaného na všetkých dátach.

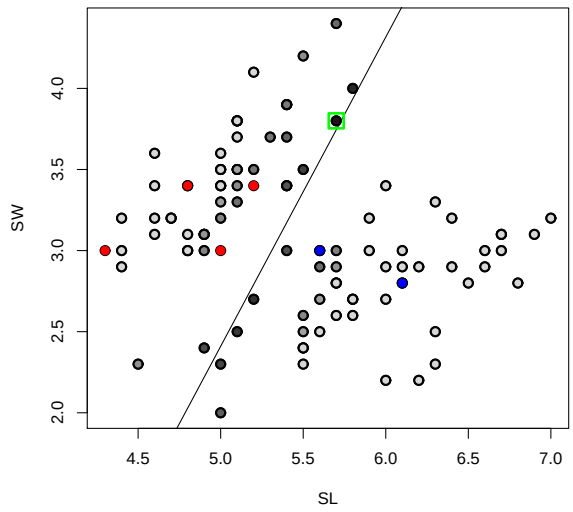


Obr. 8: Pretrénovanie modelu po zistení kategórie najinformatívnejšieho údaja pre predchádzajúci (inicializačný) model podľa stratégie *least confidence* a jeho zahrnutí do trénovacej vzorky. Zobrazená diskriminačná hranica teda vznikla na základe šiestich doposiaľ označených dát.

2.2.2 Query by Committee

Táto metóda výberu údaja, ktorou sa ako prví zaoberali autori článku [19], spočíva v tom, že zostavíme súbor modelov $K = \{\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)}\}$ konzistentných so vzorkou doposiaľ označených objektov $\boldsymbol{x}$. Cieľom tejto metódy je minimalizovať oblasť, ktorá je nejednoznačná vzhľadom na zaradenie do kategórie pre daný súbor modelov. Napríklad, ak by sa dva modely (rovnakého typu líšiace sa len parametrami) zhodovali vo všetkých dátach a iba jeden údaj by zaradili do rozdielnych kategórií, tak tento údaj by ležal v oblasti nejednoznačnosti. Čiže údajom, ktorého označenie budeme požadovať, bude práve ten, v ktorého označení sa budú modely zo súboru modelov K najviac nezhodovať. Na aplikovanie *query by committee* potrebujeme

1. skonštruovať súbor modelov konzistentných s trénovacou vzorkou (s označenými objektami)

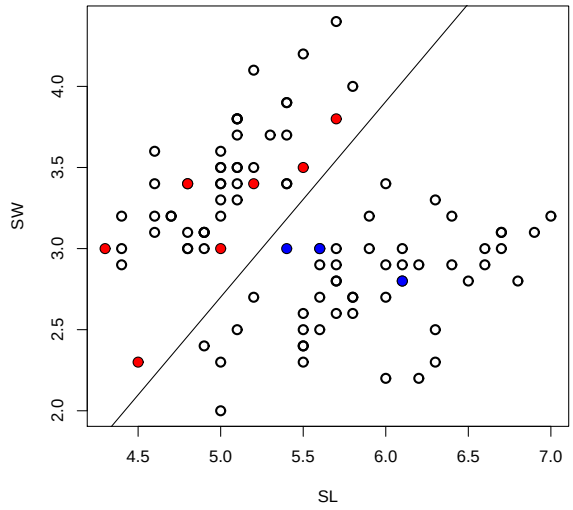


Obr. 9: Informatívnosť všetkých doposiaľ neoznačených dát vyjadrená intenzitou farby (čím tmavšia farba, tým väčšia informatívnosť) podľa *least confidence* pre súčasný model logistickej regresie reprezentovaný lineárnou diskriminačnou hranicou. Najinformatívnejší údaj je zobrazený v zelenom rámčeku.

2. definovať "mieru nezhôd" v rámci súboru modelov.

Vychádzajúc z [17] súbor modelov možno získať pomocou metódy *query by bagging* alebo *query by boosting*. Konkrétne pri *query by bagging* metóde súbor modelov zostrojíme tak, že K-krát zopakujeme náhodný výber s opakovaním z L na vytvorenie trénovacej vzorky $L^{(k)}$. Pri metóde *query by boosting*, podľa [1], každý z K náhodných výberov nie je rovnomerný, ale záleží od predchádzajúceho výberu, konkrétne od odlišnosti zatriedenia dát predošlého modelu zostrojeného na vzorke z predošlého výberu a skutočných kategórií. Podrobnejšie sa metódou *query by boosting* nebudeme zaoberať. Každý model $\theta^{(k)} \in K$ je potom zostrojený na základe svojej vlastnej trénovacej vzorky $L^{(k)}$.

Na určenie "miery nezhody" sa požívajú rôzne metódy, napr.:



Obr. 10: Diskriminačná hranica získaná sekvenčným výberom piatich dát podľa stratégie *least confidence*. Tento model bol teda vytvorený postupným zisťovaním kategórií piatich dát a ich pridaním k prvotným piatim náhodne vybratým dátam. Diskriminačná hranica sa nelíši od hranice získanej natrénovaním modelu, ktorý poznal kategórie všetkých dát.

- *vote entropy* (VE)

$$\boldsymbol{x}^*_{VE} = \arg\max_{\boldsymbol{x}\in U} \left(-\sum_{i=1}^{c} \frac{V_i(\boldsymbol{x})}{K} \ln \frac{V_i(\boldsymbol{x})}{K} \right),$$

  kde $V_i(\boldsymbol{x})$ je počet takých $p_i(\boldsymbol{x}, \boldsymbol{\beta}^{(k)}), \quad k = 1, \ldots, K,$ že $p_i(\boldsymbol{x}, \boldsymbol{\beta}^{(k)}) = \max_{j\in 1,\ldots,c} p_j(\boldsymbol{x}, \boldsymbol{\beta}^{(k)})$. Čiže $V_i(\boldsymbol{x})$ je počet hlasov, ktoré dostane i-ta kategória pre údaj $\boldsymbol{x}$. Teda, ak by sme zostavovali 10 modelov ($K = 10$) na doteraz označených dátach L a 7 z týchto 10 modelov by údaj, ktorého informatívnosť skúmame, zaradili do prvej kategórie, dva modely do druhej kategórie a jeden model do tretej kategórie, tak počty hlasov by boli takéto: $V_1(\boldsymbol{x}) = 7$, $V_2(\boldsymbol{x}) = 2$ a $V_3(\boldsymbol{x}) = 1$. Stratégiou *vote entropy* je teda na označenie vybratý taký údaj, ktorého entropia počtu hlasov jednotlivých kategórií (v pomere k celkovému počtu vytvorených modelov K) je najvyššia. Pri odvádzaní sme vychádzali z [5].

- *average Kullback-Leibler divergence* (KL)

  Druhým spôsobom určenia "miery nezhody", ktorý uvedieme, je *average Kullback-Leibler divergence* [13]. Pri tejto stratégii nie sú dôležité len konečné kategórie (hlasy), do ktorých radia jednotlivé modely sledovaný údaj, ale aj pravdepodobnosti, že dáta patria do týchto kategórií:

  $$\boldsymbol{x}^*_{KL} = \arg\max_{\boldsymbol{x}\in U} \frac{1}{K} \sum_{k=1}^{K} D(P_{\boldsymbol{\beta}^{(k)}} \| P_K),$$

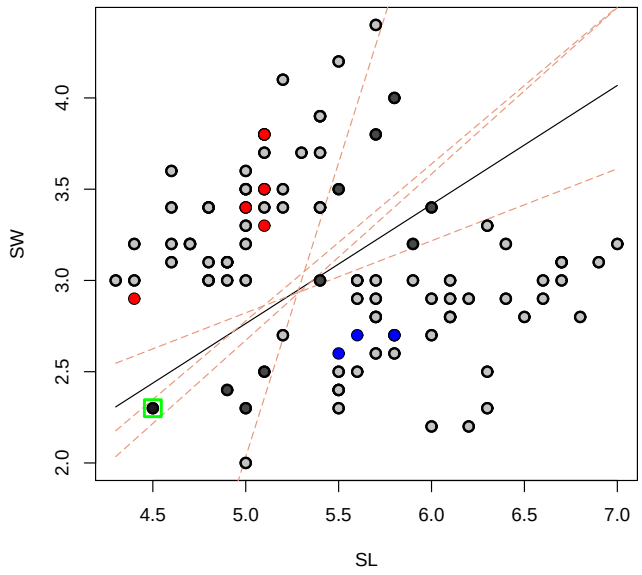
  kde Kullbackova-Leiblerova divergencia sa vypočíta ako

  $$D(P_{\boldsymbol{\beta}^{(k)}} \| P_K) = \sum_{i=1}^{c} p_i(\boldsymbol{x}, \boldsymbol{\beta}^{(k)}) \log \frac{p_i(\boldsymbol{x}, \boldsymbol{\beta}^{(k)})}{p_i(\boldsymbol{x}, K)},$$

  kde K predstavuje vytvorený súbor modelov ako celok, preto $p_i(\boldsymbol{x}, K) = \frac{1}{K} \sum_{k=1}^{K} p_i(\boldsymbol{x}, \boldsymbol{\beta}^{(k)})$. Teda priemerná pravdepodobnosť, že údaj s vektorom "čŕt" $\boldsymbol{x}$ patrí do kategórie i. Nazýva sa aj "konsenzus" pravdepodobnosti, že i je správnou kategóriou. Kullbackova-Leiblerova divergencia je mierou rozdielu medzi dvoma pravdepodobnostnými rozdeleniami. Čiže za najinformatívnejší je v tomto prípade vybratý údaj, ktorého priemerná vzdialenosť medzi rozdeleniami pravdepodobností kategórií danými modelmi od "konsenzu" je najväčšia.

Princíp stratégie *vote entropy* si znázorníme na jednoduchom príklade dát kvetov kosatcov, ktoré sme použili aj pri metóde *least confidence*. Na obr. 11 máme okrem deliacej priamky súčasného modelu logistickej regresie znázornené ďalšie štyri deliace priamky (čiarkované, svetlou farbou), ktoré sme vytvorili natrénovaním modelu na náhodnom výbere z doteraz označených dát. Vidíme, že informatívnejšie dáta sa nachádzajú v oblasti vymedzenej týmito priamkami, pretože práve to je oblasť, v ktorej sú dáta, ktoré nie sú zaradené vždy do tej istej kategórie na základe štyroch vytvorených modelov. Dáta nachádzajúce sa mimo tejto oblasti nejednoznačnosti majú pre súčasný model rovnako nízko informatívnosť. Za najinformatívnejší bol zvolený údaj znázornený v zelenom rámčeku, pretože ako jediný bol dvoma modelmi zaradený medzi Iris setosa

a zvyšnými dvoma medzi Iris virginica. Ostatné dáta v oblasti neurčitosti mali entropiu kategórií nižšiu, pretože až troma modelmi boli zaradené do jednej z kategórií a iba jedným do opačnej kategórie.



Obr. 11: Informatívnosť bodov podľa *vote entropy*. Plnou čiarou je znázornená diskriminačná hranica inicializačného modelu natrénovaného na vzorke ôsmich náhodne vybratých dát. Kategórie ostatných dát sú neznáme. Čím je neznámy údaj znázornený tmavšou farbou, tým je miera jeho informácie pre inicializačný model vyššia. Svetlými prerušovanými čiarami sú znázornené diskriminačné hranice modelov zostavených na niektorých zo vzorky doposiaľ označených dát. Najinformatívnejší údaj je ohraničený zeleným rámčekom.

2.2.3 Expected model change

Ďalšou zo známych stratégií výberu dát na označenie ich kategórie je *expected model change*, ktorá spočíva vo výbere takého údaja, ktorý by spôsobil najväčšiu zmenu doteraz vytvoreného modelu, ak by sme poznali jeho kategóriu. Vychádzajúc z [18] príkladom takejto stratégie je *expected gradient length*, ak model, ktorý používame, je založený na pravdepodobnosti, čo je aj v prípade logistickej regresie. Pri odhade parametrov logistickej regresie

sa využíva vierohodnostná funkcia ℓ, ktorej maximalizáciou získame logistické koeficienty. Čím je hodnota vierohodnostnej funkcie vyššia, tým model lepšie "fituje" podkladové dáta, čo využíva práve stratégia *expected gradient length*. Zmena modelu je pri tejto stratégii meraná dĺžkou gradientu logaritmu vierohodnostnej funkcie, čiže zlepšením "fitovania" dát. Inými slovami, požadovať sa bude označenie takého údaja, ktorý ak označíme a pridáme do trénovacej vzorky L, spôsobí najväčšiu zmenu hodnoty logaritmu vierohodnostnej funkcie. Označme $\nabla \ell_{\boldsymbol{\beta}}(L)$ gradient logaritmu vierohodnostnej funkcie ℓ vzhľadom na parameter $\boldsymbol{\beta}$. $\nabla \ell_{\boldsymbol{\beta}}(L \cup \langle \boldsymbol{x}, y \rangle)$ je teda gradient logaritmu vierohodnostnej funkcie, ktorý získame pridaním trénovacieho údaja $\langle \boldsymbol{x}, y \rangle$ do trénovacej vzorky L. Pretože skutočnú kategóriu y dopredu nepoznáme, počítame len s očakávanou hodnotou dĺžky gradientu váženou pravdepodobnosťami možných kategórií.

Všeobecne:

$$\boldsymbol{x}^*_{EGL} = \arg\max_{\boldsymbol{x} \in U} \sum_{i=1}^{c} p_i(\boldsymbol{x}, \boldsymbol{\beta}) \|\nabla \ell_{\boldsymbol{\beta}}(L \cup \langle \boldsymbol{x}, y_i \rangle)\|, \tag{3}$$

kde $\|.\|$ je Euklidovská norma.

Pretože $\|\nabla \ell_{\boldsymbol{\beta}}(L)\|$ by malo byť približne rovné 0 (pretože vektor koeficientov $\boldsymbol{\beta}$ sme hľadali metódou maximálnej vierohodnosti, čiže práve tak, aby spĺňal danú podmienku), tak platí $\nabla \ell_{\boldsymbol{\beta}}(L \cup \langle \boldsymbol{x}, y_i \rangle) \approx \nabla \ell_{\boldsymbol{\beta}}(\langle \boldsymbol{x}, y_i \rangle)$. Čiže vzťah (3) môžeme upraviť nasledovne:

$$\boldsymbol{x}^*_{EGL} \approx \arg\max_{\boldsymbol{x} \in U} \sum_{i=1}^{c} p_i(\boldsymbol{x}, \boldsymbol{\beta}) \|\nabla \ell_{\boldsymbol{\beta}}(\langle \boldsymbol{x}, y_i \rangle)\|.$$

V prípade binárnej logistickej regresie:

$$
\begin{aligned}
\boldsymbol{x}^*_{EGL} &= \arg\max_{\boldsymbol{x}\in U} \sum_{i=1}^{2} p_i(\boldsymbol{x},\boldsymbol{\beta}) \|\nabla_{\boldsymbol{\beta}}\{y_i \ln p_1(\boldsymbol{x},\boldsymbol{\beta}) + (1-y_i)\ln(1-p_1(\boldsymbol{x},\boldsymbol{\beta}))\}\| \\
&= \arg\max_{\boldsymbol{x}\in U} \sum_{i=1}^{2} p_i(\boldsymbol{x},\boldsymbol{\beta}) \|\nabla_{\boldsymbol{\beta}}\{y_i \boldsymbol{\beta}^T\boldsymbol{x} - \ln(1+e^{\boldsymbol{\beta}^T\boldsymbol{x}})\}\| \\
&= \arg\max_{\boldsymbol{x}\in U} \sum_{i=1}^{2} p_i(\boldsymbol{x},\boldsymbol{\beta}) |y_i - p_1(\boldsymbol{x},\boldsymbol{\beta})| \|\boldsymbol{x}\| \\
&= \arg\max_{\boldsymbol{x}\in U} \{p_1(\boldsymbol{x},\boldsymbol{\beta})(1-p_1(\boldsymbol{x},\boldsymbol{\beta}))\|\boldsymbol{x}\| + p_2(\boldsymbol{x},\boldsymbol{\beta})p_1(\boldsymbol{x},\boldsymbol{\beta})\|\boldsymbol{x}\|\} \\
&= \arg\max_{\boldsymbol{x}\in U} p_1(\boldsymbol{x},\boldsymbol{\beta})p_2(\boldsymbol{x},\boldsymbol{\beta})\|\boldsymbol{x}\|.
\end{aligned}
$$

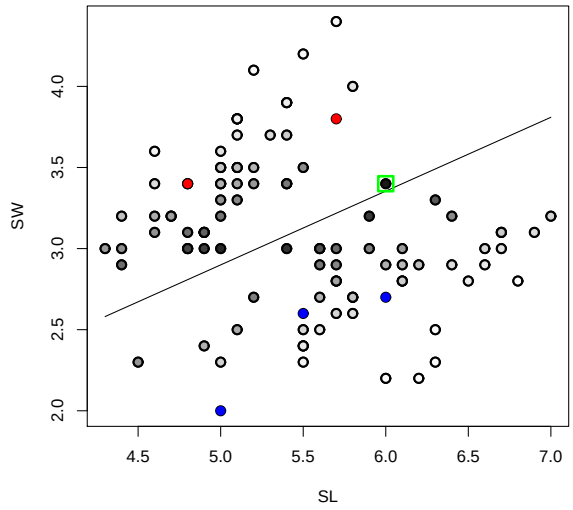
Opäť si danú stratégiu demonštrujeme na jednoduchom príklade dát kvetov Iris. Na obr. 12 vidíme prvých päť náhodne vybratých dát, na základe ktorých sme vytvorili inicializačný model, na obrázku reprezentovaný priamkou. Všetky ostatné dáta, ktorých kategórie sú zatiaľ neznáme, sú znázornené farbou v závislosti od ich informatívnosti. Údaj, ktorý je najinformatívnejší, sa nachádza v zelenom rámčeku.

2.2.4 Expected error reduction

Pri stratégii *expected model change* sme sa zameriavali na to, o koľko sa zmení model pridaním ďalšieho objektu do trénovacej vzorky. Teraz však rozhodovacím kritériom bude to, o koľko sa zredukuje očakávaná chyba modelu. Myšlienka tejto metódy uvedenej v [16] spočíva v odhade očakávanej chyby na neoznačených dátach z množiny U, pričom model je trénovaný na vzorke $L \cup \langle \boldsymbol{x}, y\rangle$. Jednou z možností je vybrať taký objekt $\boldsymbol{x}$, ktorý bude minimalizovať očakávanú "0/1" stratovú funkciu, čiže očakávaný počet nesprávnych predikcií:

$$
\boldsymbol{x}^*_{0/1} = \arg\min_{\boldsymbol{x}\in U} \sum_{i=1}^{c} p_i(\boldsymbol{x},\boldsymbol{\beta}) \left(\sum_{u=1}^{U} 1 - p_{\hat{y}}(\boldsymbol{x}^{(u)}, \boldsymbol{\beta}^{+\langle \boldsymbol{x}, y_i\rangle}) \right),
$$

kde $\boldsymbol{x}^{(u)}, u \in \{1,\ldots,U\}$ sú postupne vektory "čŕt" pre všetky neoznačené dáta a $\boldsymbol{\beta}^{+\langle \boldsymbol{x}, y_i\rangle}$ je vektor koeficientov získaný trénovaním modelu na vzorke



Obr. 12: Informatívnosť bodov podľa stratégie *expected gradient length*. Pomocou trénovacej vzorky získanej označením piatich náhodne vybratých dát (farebne) bol vytvorený inicializačný model. Informatívnosť dát pre tento model sa zvyšuje intenzitou farby a je vyznačený aj najinformatívnejší údaj.

$L \cup \langle \boldsymbol{x}, y_i \rangle$. Iným prístupom je minimalizovať "log-loss" funkciu:

$$\boldsymbol{x}^*_{log} = \arg\min_{\boldsymbol{x} \in U} \sum_{i=1}^{c} p_i(\boldsymbol{x}, \boldsymbol{\beta}) \left(- \sum_{u=1}^{U} \sum_{j=1}^{c} p_j(\boldsymbol{x}^{(u)}, \boldsymbol{\beta}^{+\langle \boldsymbol{x}, y_i \rangle}) \ln p_j(\boldsymbol{x}^{(u)}, \boldsymbol{\beta}^{+\langle \boldsymbol{x}, y_i \rangle}) \right),$$

ktorá minimalizuje očakávanú entropiu. Nevýhodou metódy *expected error reduction* je, že pre každý objekt z množiny neoznačených objektov a pre každé z jeho možných označení musí byť model pretrénovaný, čo je v niektorých prípadoch, ako aj v prípade logistickej regresie, výpočtovo náročné.

Stratégiu *expected error reduction* neuvádzame na príklade ako predošlé stratégie, pretože po jej aplikácii na dáta Iris sme usúdili, že použitie tejto stratégie nie je veľmi vhodné na tento súbor dát.

2.2.5 Variance reduction

Nasledujúce stratégie redukcie variancie sú odvodené zo štatistickej teórie optimálneho návrhu experimentov. Súvis medzi nimi bližšie uvádzame v prílohe.

Veľmi podstatná je v tomto prístupe práca s Fisherovou informačnou maticou, ktorej sme sa už venovali v kapitole 1. Využijúc Cramérovu-Raovu nerovnosť, inverzia Fisherovej informačnej matice určuje spodnú hranicu variancie odhadov parametrov modelu. Teda, ak chceme minimalizovať variancie odhadov parametrov, mali by sme maximalizovať Fisherovu informačnú maticu (alebo minimalizovať jej inverziu). Ak by model pozostával len z jedného parametra, maximalizácia je samozrejmá. Avšak, ak je v modeli viacero parametrov, Fisherova informačná matica je štvorcovou maticou, ktorej rozmer je daný práve počtom parametrov. Čo v takomto prípade znamená maximalizácia Fisherovej informačnej matice? Podľa [17] existuje viacero prístupov maximalizácie Fisherovej informačnej matice vo viacrozmernom prípade.

- *A-optimalita*

  A-optimalita minimalizuje stopu inverznej informačnej matice. Takáto minimalizácia zodpovedá minimalizácii priemernej variancie odhadu parametrov. A-optimalita je najviac populárnou možnosťou minimalizácie inverzie Fisherovej informačnej matice v rámci sekvenčných metód klasifikácie.

- *D-optimalita*

  D-optimalita maximalizuje determinant informačnej matice. Keďže determinant je mierou objemu, takáto stratégia môže byť chápaná ako minimalizácia oblasti neurčitosti, čiže princíp je podobný ako pri stratégii *query by committee*.

- *E-optimalita*

  E-optimalita maximalizuje najväčšie vlastné číslo informačnej matice. Nie je veľmi často používanou stratégiou pri sekvenčných metódach a ani my sa jej nebudeme hlbšie venovať.

A-optimalitu sme si zvolili za stratégiu, ktorú porovnáme so všetkými ostatnými opísanými stratégiami. Matematický zápis výberu najinformatívnej-

šieho údaja je nasledovný:

$$\boldsymbol{x}_A^* = \arg\min_{\boldsymbol{x}\in U} tr\left[\left(I(\boldsymbol{\beta})^{+\boldsymbol{x}}\right)^{-1}\right],$$

kde

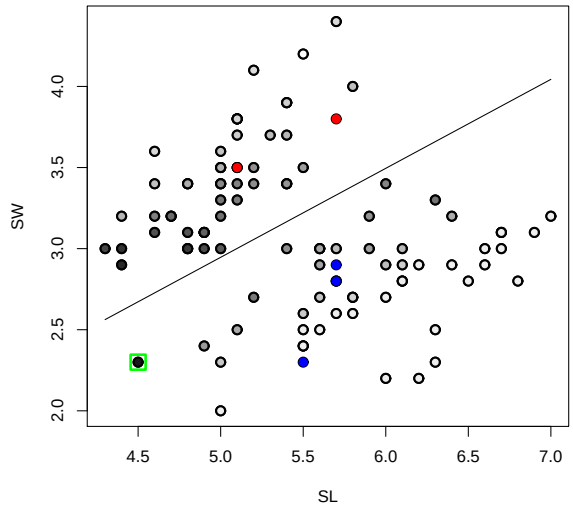
$tr[.]$ je stopa matice, čiže súčet diagonálnych prvkov matice,

$I(\boldsymbol{\beta})^{+\boldsymbol{x}}$ je Fisherova informačná matica po pridaní údaja $\boldsymbol{x}$, čiže $I(\boldsymbol{\beta})^{+\boldsymbol{x}} =$

$$\left(\boldsymbol{x}_1, \quad \ldots, \quad \boldsymbol{x}_n, \quad \boldsymbol{x}\right) diag\{\pi_{\boldsymbol{x}_1}(1-\pi_{\boldsymbol{x}_1}), \ldots, \pi_{\boldsymbol{x}_n}(1-\pi_{\boldsymbol{x}_n}), \pi_{\boldsymbol{x}}(1-\pi_{\boldsymbol{x}})\} \begin{pmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \\ \boldsymbol{x} \end{pmatrix},$$

kde $\pi_{\boldsymbol{x}_i} = \pi(\boldsymbol{x}_i, \boldsymbol{\beta})$.

Najprv si však ukážeme, ako pri predošlých stratégiách, hodnotu informatívnosti dát na príklade dát Iris.



Obr. 13: Informatívnosť údajov pre inicializačný model vypočítaná pomocou A-optimality je vyjadrená intenzitou farby. Inicializačný model bol vytvorený na základe náhodných piatich označených dát. Najinformatívnejší údaj sa nachádza v zelenom rámčeku.

Na obr. 13 vidíme inicializačný model vytvorený na piatich dátach, ktorých skutočnú kategóriu už poznáme. Všimnime si, že informatívnejšie dáta sú koncentrované okolo deliacej priamky, avšak za najinformatívnejší údaj nebol zvolený najbližší údaj pri priamke. A-optimalita totiž zohľadňuje aj to, ako môže označenie údaja ovplyvniť model, resp. jeho diskriminačnú hranicu. Ak by sme označili údaj, ktorý sa nachádza v strede oblasti určenej dátami, jeho kategória neovplyvní model (určujúci deliacu priamku) tak, ako by ho mohla ovplyvniť kategória údaja, ktorý sa nachádza na okraji tejto oblasti.

2.2.6 Expected classification entropy

Po naštudovaní predošlých, v praxi už aplikovaných stratégií výberu dát, sme po viacerých úvahách dospeli k novej stratégii, ktorú sme nazvali *expected classification entropy*. Výber údaja na označenie spočíva v tom, že sa snažíme vybrať taký údaj, ktorého kategória najviac ovplyvní kategórie ostatných doposiaľ nezaradených dát. Čiže sledujeme, či údaj, ktorého informatívnosť chceme vypočítať, po postupnom zaradení do jednotlivých kategórií, ovplyvní kategóriu danú modelom zvyšných neoznačených dát alebo nie. Vylepšením, ktoré sme pokladali za správnejšie riešenie je, že nesledujeme zmeny kategórií dát zo vzorky, ale dát, ktoré nanovo vygenerujeme z rozdelenia, ktoré sa "podobá" na rozdelenie dát v tréningovej vzorke. Po vytvorení klasifikačného modelu nás v praxi zaujíma totiž určenie kategórie nejakého nového nadobudnutého údaja.

Konkrétne táto stratégia prebieha tak, že najprv náhodne vygenerujeme N dát z rozdelenia daného vzorkou. Tieto novovygenerované dáta pochádzajú zo zmesi normálnych rozdelení, pričom strednými hodnotami jednotlivých rozdelení zmesi sú všetky doposiaľ označené aj neoznačené dáta. Takisto musíme zvoliť aj disperzie rozdelení, ktoré čím zvolíme väčšie, tým sa bude zmes rozdelení viac podobať na rovnomerné rozdelenie. Nasledujúcim krokom je, že údaj, ktorého informatívnosť chceme vypočítať, postupne zaradíme do všetkých kategórií a pre každú kategóriu model pretrénujeme. Pozrieme sa na tie novovygenerované dáta, ktoré menia svoju kategóriu v závislosti od kategórie sledo-

vaného údaja. Nemôžeme len jednoducho zrátať počet takýchto novovygenerovaných dát, pretože kategórie, ktoré ovplyvnia kategórie zvyšných dát, môžu mať len veľmi nízku pravdepodobnosť. Preto sme sa rozhodli vynásobiť počet takýchto novovygenerovaných dát entropiou kategórií sledovaného údaja, čím predídeme spomenutej situácii.

V binárnom prípade teda pre konkrétny sledovaný údaj sčítame počet takých novovygenerovaných dát, ktoré by v prípade kategórie 1 sledovaného údaja boli zaradené do jednej kategórie a v prípade kategórie 2 sledovaného údaja by boli zaradené do druhej, opačnej kategórie. Tento počet vynásobíme entropiou [2] pravdepodobností kategórií sledovaného údaja. Za najinformatívnejší zvolíme údaj, ktorého tento súčin bude maximálny.

Matematickým zápisom voľby najinformatívnejšieho údaja (aj v prípade viacerých kategórií) je:

$$\boldsymbol{x}^*_{ECE} = \arg\max_{\boldsymbol{x}\in U} \left(-\sum_{n=1}^{N}\sum_{i=1}^{c} P_i \ln P_i \right),$$

kde $P_i = \sum_{j=1}^{c} p_j(\boldsymbol{x}, \boldsymbol{\beta})$, kde pre j platí:

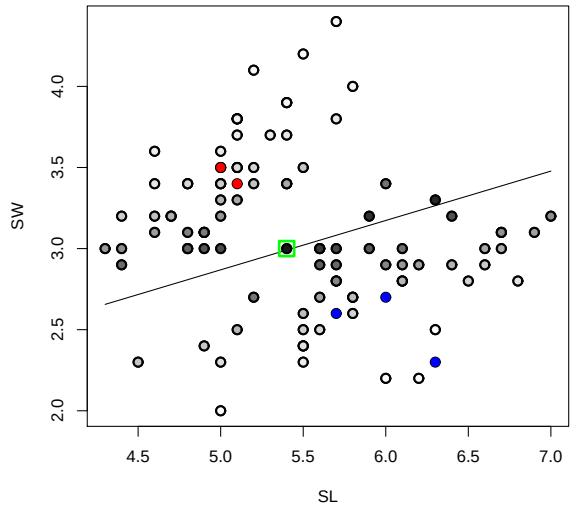
$$p_i(\boldsymbol{x}^{(n)}, \boldsymbol{\beta}^{+\langle \boldsymbol{x}, y_j \rangle}) = \max_k p_k(\boldsymbol{x}^{(n)}, \boldsymbol{\beta}^{+\langle \boldsymbol{x}, y_j \rangle}).$$

Vektory $\boldsymbol{x}^{(n)}, n \in \{1, \ldots, N\}$ sú postupne vektory "čŕt" pre všetky novovygenerované dáta. P_i je teda súčet pravdepodobností tých kategórií údaja $\boldsymbol{x}$, do ktorých keď hypoteticky zaradíme údaj $\boldsymbol{x}$ a model pretrénujeme, tak na základe tohto modelu bude novovygenerovaný údaj $\boldsymbol{x}_n$ zaradený do kategórie i.

Keďže pri hľadaní najinformatívnejšieho údaja musíme pre každý neoznačený údaj a pre každú jeho možnú kategóriu model pretrénovať, na nájdenie najinformatívnejšieho údaja potrebujeme viac času (kvôli výpočtovej zložitosti) ako pri niektorých iných stratégiách. Tento čas je ale v porovnaní s náročnosťou zistenia skutočnej kategórie dát zanedbateľný.

Opäť danú stratégiu, resp. informatívnosť dát, ilustrujeme na príklade dát

[2] Entropia údaja $\boldsymbol{x}$ je rovná $-(p_1(\boldsymbol{x}, \boldsymbol{\beta}) \ln p_1(\boldsymbol{x}, \boldsymbol{\beta}) + p_2(\boldsymbol{x}, \boldsymbol{\beta}) \ln p_2(\boldsymbol{x}, \boldsymbol{\beta}))$.



Obr. 14: Informatívnosť dát podľa stratégie *expected classification entropy* vyjadrená intenzitou farby. Najinformatívnejší údaj pre inicializačný model je vyznačený zelenou farbou.

Iris. Na obr. 14 vidíme inicializačný logistický model reprezentovaný priamkou oddeľujúcou jednotlivé kategórie, ktorý bol vytvorený na základe náhodne vybratých piatich dát. Kategórie ostatných dát sú zatiaľ neznáme. Ich informatívnosť je vyjadrená intenzitou sivej farby. Najinformatívnejší z nich je v zelenom rámčeku. Zistíme kategóriu tohto údaja a bude pridaný do trénovacej vzorky. Model bude pretrénovaný, opäť sa určí informatívnosť jednotlivých nezatriedených dát podľa stratégie *expected classification entropy* a celý proces sa zopakuje.

2.2.7 Porovnanie stratégií

Porovnanie stratégií vykonáme na základe percenta správne zatriedených dát. Po každom pridaní údaja do trénovacej vzorky sa pozrieme na to, koľko dát je na základe dovtedy vytvoreného modelu zaradených do správnej kategórie. Porovnanie vykonáme pre dva rôzne súbory dát. Jedným z nich bude už spomenutý súbor kvetov kosatcov (Iris).

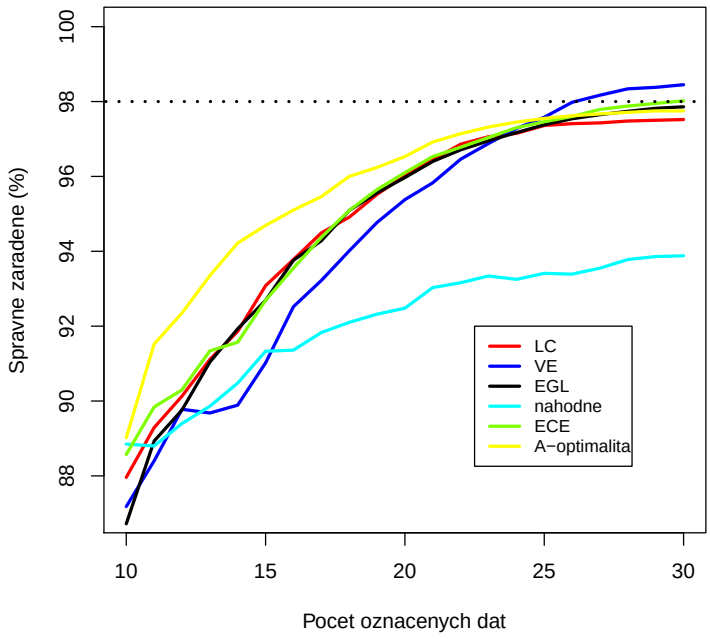
Iris dáta

Na rozdiel od predchádzajúceho použitia dát Iris, teraz budeme brať do úvahy všetky vysvetľujúce premenné, čiže dĺžku a šírku kališného lístka a taktiež dĺžku a šírku okvetného lístka (doteraz sme používali len dve z nich kvôli grafickej vizualizácii jednotlivých stratégií). Keďže používame binárnu logistickú regresiu, vždy môžeme vytvárať model len pre dve z troch možných druhov kosatcov. Postupne vytvoríme model pre všetky tri dvojice druhov. Všetky druhy kosatcov tvoria iba dva na prvý pohľad oddeliteľné zhluky. Jeden z nich tvorí druh Iris setosa a druhý sa skladá zo zvyšných dvoch druhov Iris versicolor a Iris virginica. Iris versicolor a Iris virginica teda nie sú ľahko separovateľné. Najprv sa pozrieme na porovnanie stratégií na dátach práve týchto dvoch druhov, Iris versicolor a Iris virginica.

Experiment budeme vykonávať tak, že každú z porovnávaných stratégií zrealizujeme vždy viackrát a výsledné percentá správne zatriedených dát určíme ako priemery zo všetkých realizácií. Čiže náhodne zvolíme dáta určitého počtu, na základe ktorých vytvoríme inicializačný model a postupne k nim budeme pridávať jednotlivé údaje podľa danej stratégie. Po pridaní dopredu zvoleného počtu dát skončíme. Náhodne vyberieme nové dáta a celý postup zopakujeme.

Na obr. 15 vidíme porovnanie jednotlivých stratégií s použitím už spomenutých dát kvetov kosatcov a konkrétne dvoch druhov Iris versicolor a Iris virginica.

Na x-vej osi je počet dát trénovacej vzorky, čiže počet dát, ktorých skutočnú kategóriu sme museli zistiť. Celkový počet dát týchto dvoch druhov kvetov kosatcov je 100, čiže, ako vidíme, na inicializačný model sme vždy vybrali náhodne 10% zo všetkých dát. Postupne podľa požiadaviek jednotlivých stratégií sme zisťovali druh ďalších 20-tich dát, čiže spolu sme použili 30% údajov na vytvorenie konečného modelu logistickej regresie. Každú zo stratégií sme zrealizovali 100-krát a priemer percenta správne zatriedených dát s použitím



Obr. 15: Porovnanie stratégií (dáta IRIS - kategórie Iris versicolor a Iris virginica) prostredníctvom závislosti priemerného percenta správne zatriedených dát od počtu označených dát, ktoré boli sekvenčne pridávané podľa jednotlivých stratégií. Priemerné percento je vypočítané zo 100 modelov pre každú zo stratégií. Každý model mal vlastnú náhodne vybratú vzorku na vytvorenie inicializačného modelu. Prerušovanou vodorovnou čiarou je vyjadrené percento správne zatriedených dát modelom, ktorý poznal kategórie všetkých dát.

daného počtu označených dát vidíme na y-vej osi. Úspešnosť incializačného modelu (s použitím 10 dát) nezáleží na vybranej stratégií a vidíme, že sa hýbe v rozmedzí 87,5 - 88,5%. Červenou farbou je znázornená stratégia *least confidence* (LC), tmavomodrou *vote entropy* (VE), čiernou *expected gradient length* (EGL), svetlomodrou stratégia, ktorá do trénovacej vzorky postupne pridáva vždy náhodný údaj z doposiaľ nezatriedených údajov (nahodne), zelenou *expected classification entropy* (ECE) a žltou stratégia A-optimalita (A-optimalita). Čierna čiarkovaná vodorovná čiara predstavuje percento správne zatriedených dát modelom, ktorý vytvoríme použitím všetkých 100 trénovacích

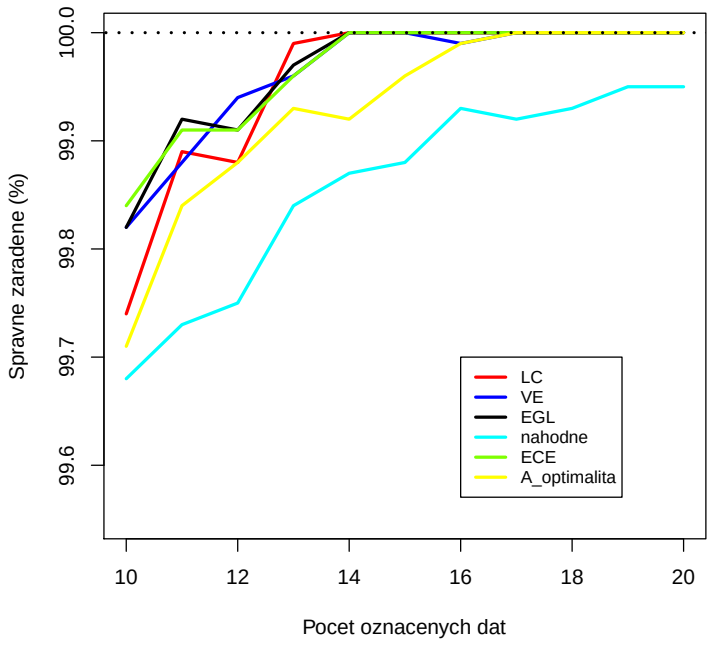
dát. Percento správne zaradených údajov modelom s použitím všetkých dát je rovné 98%. Keďže dáta nie sú separovateľné lineárnou hranicou, žiaden lineárny model nemôže dosiahnuť správne zatriedenie všetkých dát. Poznamenajme však, že z hľadiska počtu správne zaradených dát môže existovať aj lepšia lineárna separujúca hranica ako tá získaná logistickou regresiou, pretože cieľom logistickej regresie nie je minimalizovať počet nesprávne zatriedených dát. V našom konkrétnom prípade dát kvetov Iris versicolor a Iris virginica je najviac možné 99% správne zatriedenie. Ako môžeme vidieť, stratégiou *expected classification entropy* sme dosiahli hranicu 98% už s použitím len 30 dát, stratégiou *vote entropy* len s použitím 26 dát a pridávaním pár ďalších dát sme dokonca túto hranicu presiahli. Vidíme, že stratégie *least confidence*, *expected gradient length* a *expected classification entropy* sú z hľadiska počtu správne zaradených dát veľmi podobné. Zo začiatku (s malým počtom trénovacích dát) bol najlepší model vytváraný pomocou stratégie A-optimality. [3] s použitím 13 dát bola stratégia A-optimality v priemere až o približne 2% lepšia ako druhá najlepšia stratégia *expected classification entropy* (A-optimalita - 92,35%, ECE - 90,3%). Stratégia *vote entropy* bola zo začiatku najslabšou zo všetkých ostatých stratégií (náhodný výber dát na označovanie neberieme ako stratégiu). Po označení ďalších 13 dát (dokopy 23 dát) sa vyrovnala ostatným stratégiám a označovaním ďalších dát bola vždy najúspešnejšou. Všimnime si ďalej, že keď bolo označených 23 dát, všetky stratégie boli približne rovnako úspešné. Všetky stratégie (okrem *vote entropy*) označovaním ďalších dát mali v priemere skoro rovnaké percento správne zatriedených dát. Poznamenajme ešte, že úspešnosť stratégií *vote entropy* a taktiež *expected classification entropy* závisí od nastavenia parametrov. Pri stratégii *vote entropy* sa volí počet modelov K, ktoré vytvárame pomocou súčasnej trénovacej vzorky. Zvolené bolo $K = 8$. Pri metóde *expected classification entropy* musíme zvoliť počet novo-

[3] Pripomíname, že kvalitu modelu posudzujeme percentom správne zatriedených dát, avšak mohli by sme použiť aj iné miery kvality, napríklad vzdialenosťou separujúcej nadroviny súčasne vytvoreného modelu od separujúcej nadroviny modelu trénovaného na všetkých dátach. Väčšinou však v praxi je dôležité určenie výslednej kategórie.

vygenerovaných dát N, ktorých zmeny kategórií sledujeme a takisto disperziu normálnych rozdelení, z ktorých zmesi sme generovali nové dáta. Zvolili sme $N = 200$ a štandardnú odchýlku 0.2. Ako sme mohli čakať, najhoršie modely boli vytvárané v prípade, keď sme iba náhodne vyberali dáta, ktoré pridáme do trénovacej vzorky. Takýmto spôsobom sme zistením kategórie dokopy 30-tich dát dosiahli priemernú úspešnosť skoro 94%, zatiaľ čo stratégiami sme dosiahli približne 98% úspešnosť.

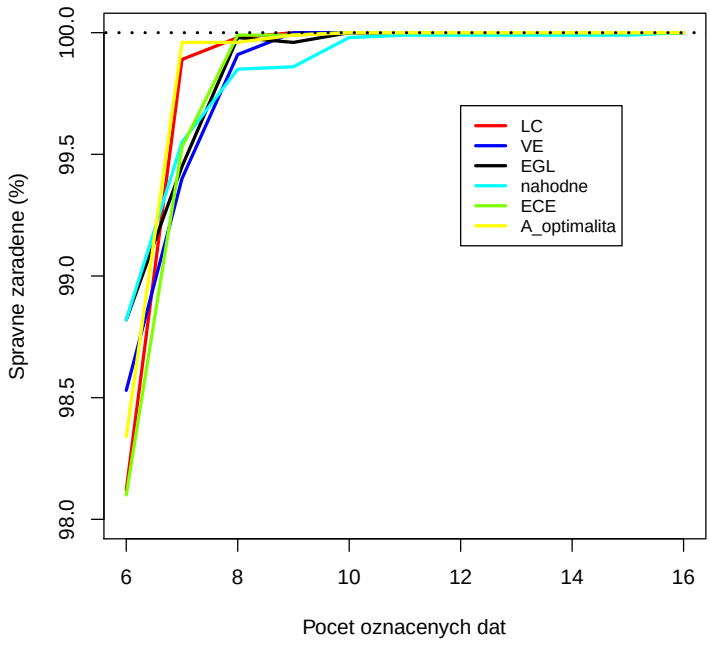
Porovnanie stratégií zrealizujeme aj pre druhy Iris setosa a Iris versicolor. Presne taký istý postup ako pri porovnávaní stratégií s použitím dát druhov Iris versicolor a Iris virginica sme použili aj na tieto dáta. Výsledok experimentu je na grafe na obr. 16. Keďže tieto dva druhy sú od seba ľahko oddeliteľné, už inicializačný model vytvorený pomocou náhodne vybratých 10-tich dát má vysokú úspešnosť určenia správnej kategórie (priemerne približne 99,8%). Ďalším následkom ľahkej separácie je, že po postupnom zistení kategórie už len ďalších 4 dát, väčšina modelov daných stratégií mala priemernú úspešnosť 100% (iba stratégia A-optimality mala priemernú úspešnosť modelov vytvorených na 14 dátach menšiu a to 99,92%). Náhodným výberom dát pridávaných do trénovacej vzorky sa ani po zistení kategórie 20 dát nedosiahla 100%-ná úspešnosť. Celkovo je náhodný výber viditeľne najhorším. Zatiaľ čo pri predchádzajúcom porovnaní na druhoch Iris versicolor a Iris virginica bola zo začiatku A-optimalita najlepšou stratégiou, v tomto prípade bola trochu horšou v porovnaní s ostatnými stratégiami, ktoré sú v tomto prípade navzájom podobné z hľadiska priemerných percent správne zatriedených dát.

Pre úplnosť uvedieme aj porovnanie stratégií pomocou dát druhov Iris setosa a Iris virginica. Tieto dva druhy sú však ešte ľahšie oddeliteľné ako predchádzajúce dva druhy. Už pri náhodnom výbere 10-tich dát model väčšinou dosiahol 100%-nú úspešnosť, preto sme museli začať s menším počtom náhodne vybratých dát na vytváranie inicializačných modelov, konkrétne vyberali sme vždy 6 dát. Výsledok experimentu sa nachádza na obr. 17. Už s použitím takéhoto malého počtu bola priemerná úspešnosť inicializačných modelov veľmi



Obr. 16: Porovnanie stratégií (dáta IRIS - kategórie Iris setosa a Iris versicolor).

vysoká (približne 98,5%). Po pridaní ďalších dvoch dát do trénovacej vzorky bola úspešnosť skoro všetkých modelov pomocou všetkých stratégií 100%. Po zistení druhu 8-mich a 9-tich dát je ešte malý rozdiel v použití stratégií oproti náhodnému výberu (použitím stratégií sme dosiahli lepšie modely), avšak po pridaní ďalšieho, čiže štvrtého dáta majú všetky stratégie 100%-nú úspešnosť správneho roztriedenia dát a aj náhodným výberom sme získali len zanedbateľne nižšiu úspešnosť (99,98%), čo znamená, že len dva zo 100 modelov vytvorených takýmto spôsobom nemali 100%-nú úspešnosť, ale len 99 % dát roztriedili správne. Náhodným pridávaním ďalších dát vždy už len jeden model zo 100 dosiahol len 99%-nú úspešnosť. Označením dokopy 16-tich dát už všetky modely vytvorené náhodným výberom 100%-ne správne zatriedili dáta.



Obr. 17: Porovnanie stratégií (dáta IRIS - kategórie Iris setosa a Iris virginica).

Diagnostika rakoviny prsníka - dáta

Dáta kvetov kosatcov sme použili na demonštrovanie použitia a úspešnosti jednotlivých stratégií sekvenčných metód klasifikácie dát. Tieto dáta nie sú typickým príkladom použitia sekvenčných metód, pretože spomenuté metódy je výhodné použiť v prípadoch, keď na určenie kategórie trénovacích dát potrebujeme vykonať nákladný experiment. V prípade kvetov kosatcov je určenie ich druhu nenáročné. Preto urobíme porovnanie stratégií aj na dátach, kde je použitie sekvenčných metód opodstatnené. Príkladom takýchto dát sú dáta diagnostiky rakoviny prsníka (angl. *breast cancer*). Tieto dáta boli prvýkrát použité v článku [20], ktorého autori sú tvorcami spomínaného súboru dát a v ktorom sa nachádza aj popis jednotlivých premenných. [4]

[4]Samotné dáta sú dostupné na stránke `http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)`.

V nasledujúcom texte sme využili informácie dostupné v [3]. Rakovina prsníka postihuje každú dvanástu ženu, jej výskyt stále narastá a je druhou najčastejšou príčinou úmrtia na rakovinu u žien. Pri podozrení na výskyt rakoviny prsníka sa žena najskôr podrobí vyšetreniam mamografom, ultrazvukom, prípadne magnetickou rezonanciou. Na základe výsledkov týchto vyšetrení sa dá konštatovať len podozrenie na výskyt rakoviny. Na jej definitívne potvrdenie sa vykonáva biopsia. Pomocou nej dokážu lekári zistiť druh nádoru, prípadne štádium rakoviny. Na vykonanie biopsie je potrebné odobrať vzorku podozrivého tkaniva a pomocou mikroskopu vykonať histologické vyšetrenie. Biopsia sa vykonáva ihlou alebo chirurgicky v celkovej alebo lokálnej anestézii. Výnimkou je tzv. tenkoihlová aspirácia (FNA - *fine-needle aspiration*), ktorá trvá len niekoľko minút a stačí ju vykonať ambulantne. FNA je najmenej invazívnou bioptickou metódou. Nevýhodou FNA je, že takýmto spôsobom sa získa len malá vzorka tkaniva, tkanivo sa môže počas odberu poškodiť a vyšetrenie je potom zložitejšie a diagnóza nepresnejšia. FNA takto vedie k vysokému percentu "falošne" negatívných výsledkov a je nutné vykonať kontrolné biopsie. Najdôveryhodnejšou metódou je chirurgická biopsia, pomocou ktorej sa dá zistiť o náleze najviac informácií. Jej nevýhodou je však, že po takomto chirurgickom výkone zostáva na prsníku jazva a môže zmeniť vzhľad prsníka. Takisto sú tu riziká spojené s akýmkoľvek iným chirurgickým zákrokom.

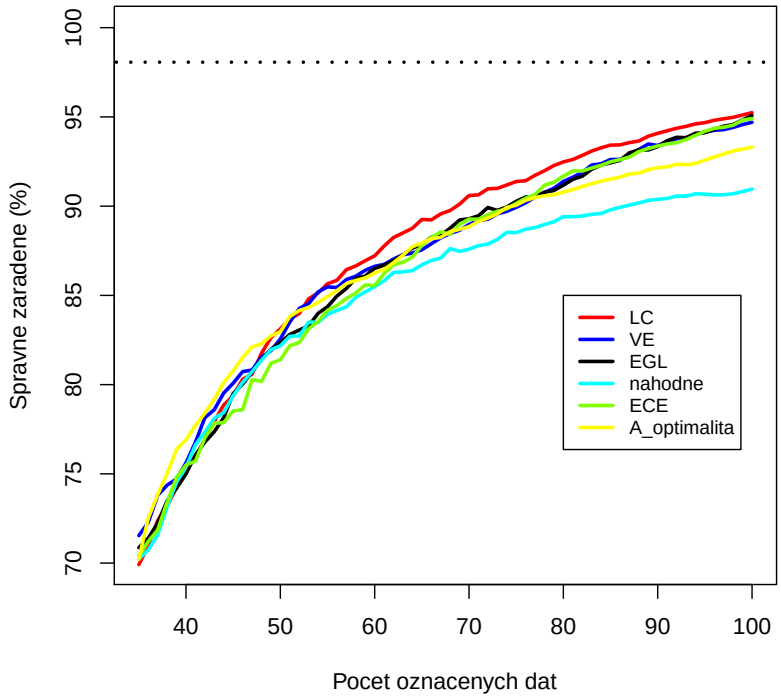
Pomocou dostupných dát získaných pomocou tenkoihlovej aspirácie sa budeme snažiť vytvoriť model, ktorý bude vedieť čo najpresnejšie diagnostikovať, či ide o malígny (zhubný) alebo benígny (nezhubný) nádor. Takýto model je používaný na lepšie predikovanie zhubnosti, resp. nezhubnosti nádoru a žena prípadne nemusí byť podrobená chirurgickej biopsii. My sa budeme snažiť postupovať tak, ako by mohli postupovať vedci na vytvorenie takéhoto modelu, ak by mali dostupné len tieto výsledky pozorovaní a výsledná informácia o zhubnosti, resp. nezhubnosti by im chýbala. Určite najdôveryhodnejší model by vznikol použitím informácie o všetkých dátach, avšak na zistenie takejto informácie je potrebné vykonať chirurgickú biopsiu, ktorej nevýhody sme

popísali. Preto je na vytvorenie takéhoto modelu vhodné použiť sekvenčné metódy, ktoré sa aplikujú v situáciách ako je táto, kedy obdržanie nezatriedených dát (pozorovania na bunkách vzorky tkaniva) je ľahké, ale zistenie ich druhu (malígny/benígny) je náročné (vykonanie chirurgickej biopsie).

V našej databáze sa nachádza 569 údajov výsledkov FNA, z ktorých 212 bolo identifikovaných ako malígne a 357 ako benígne potvrdené chirurgickou biopsiou. Každý údaj predstavuje jednu odobratú vzorku, v ktorej sa nachádza viacero buniek (približne 10 až 40 buniek). Na týchto bunkách bolo sledovaných 10 veličín, ako napríklad ich priemer, textúra buniek, ich obvod alebo symetria. Keďže v každej vzorke sa nachádzalo viacero buniek, ku každej vzorke je zaznamenaná priemerná hodnota každej z veličín v danej vzorke, štandardná odchýlka a najvyššia hodnota jednotlivých veličín (priemer z troch najvyšších hodnôt). Takýmto spôsobom máme ku každej vzorke dostupných 30 charakteristík, ktoré nám budú slúžiť ako vysvetľujúce premenné logistickej regresie. My budeme predpokladať, že skutočné zatriedenie medzi malígny alebo benígny nádor nepoznáme, ale v danej databáze sa táto informácia nachádza. Čiže ak bude stratégiami vyžiadané zistenie skutočného zatriedenia, poskytneme túto informáciu z dostupných dát. V praxi by musela byť vykonaná chirurgická biopsia.

Keďže sa ku každému údaju nachádza 30 charakteristík, model logistickej regresie bude mať až 31 parametrov. Preto musíme inicializačný model vytvoriť na väčšej vzorke. Zvolili sme 35 náhodných dát, ktorých kategórie sme zistili a pomocou nich sme model natrénovali. Výsledné porovnanie všetkých stratégií sa nachádza na obr. 18.

Porovnanie sme realizovali presne ako pri porovnaní jednotlivých stratégií na dátach Iris. Čiže na x-vej osi je počet dát, ktorých kategória je pre model doposiaľ známa a na y-vej sa nachádza priemerné percento správne zatriedených dát 100 modelov vytvorených podľa danej stratégie. Každý z modelov mal vlastnú trénovaciu vzorku na vytvorenie inicializačného modelu a sekvenčne boli k tejto vzorke pridávané dáta podľa požiadaviek danej stratégie.



Obr. 18: Porovnanie stratégií na dátach diagnostiky rakoviny prsníka popisujúcich vzorku tkaniva získanú tenkoihlovou aspiráciou. Percento správne zaradených dát pre daný počet označených dát je počítané ako priemer z percent správne zaradených dát 100 modelov, ktoré boli vytvorené sekvenčným výberom dát požadovaných na označenie podľa danej stratégie, pričom každý z modelov mal vlastnú náhodnú inicializačnú vzorku 35-tich dát. Prerušovaná vodorovná priamka predstavuje percento správne zatriedených dát modelom, ktorý poznal správne kategórie všetkých dát.

Farebné rozlíšenie a skratky stratégií sme ponechali tie isté ako pri porovnaní pomocou dát Iris. Na natrénovanie inicializačného modelu sme náhodne vyberali 35 dát, čiže približne 6% z celkového počtu 569. Priemerná úspešnosť inicializačných modelov sa hýbala v rozpätí približne 70 až 72.5%. K inicializačnej vzorke dát bolo postupne pridaných ďalších 65 dát, čiže výsledné modely boli vytvorené na vzorke približne 17.5% dát. Pomocou tohto relatívne malého percenta dát sme niektorými stratégiami dosiahli priemerne až 95%-né

správne zatriedenie všetkých dát, pričom ak model poznal kategóriu všetkých dát zo vzorky, dosiahol úspešnosť správneho roztriedenia približne 98%. Z grafu vidíme, že zo začiatku (malý počet označených dát) sa modely získané jednotlivými stratégiami od seba veľmi neodlišovali (z hľadiska priemerného percenta správne zatriedených dát), dokonca sa nelíšia ani od modelov vytváraných náhodným pridávaním dát do trénovacej vzorky. Aj keď len malým rozdielom, ale modely vytvárané pomocou stratégie A-optimality majú o čosi vyššie priemerné percentá správne zatriedených dát. Rozdiel medzi stratégiami je výraznejší až vo fáze, keď je označených spolu 60 dát. Od tohto počtu až po konečných 100 označených dát sú modely vytvárané náhodným pridávaním dát do trénovacej vzorky výrazne horšie oproti modelom vytváraným pomocou stratégií sekvenčných metód klasifikácie dát a zároveň modely vytvárané pomocou *least confidence* sú úspešnejšie v porovnaní s modelmi ostatných stratégií. Najväčší rozdiel môžeme pozorovať na modeloch trénovaných na 60-tich až 80-tich dátach, kedy sú modely stratégií *least confidence* priemerne o 2% správne zatriedených dát úspešnejšie oproti modelom všetkých ostatných stratégií, ktoré sú skoro rovnako úspešné. Ďalšiu výraznejšiu zmenu vidíme pri počte 80-tich označených dát, kedy sa rastúci trend krivky A-optimality aj stratégie *least confidence* výraznejšie spomalil oproti krivkám ostatných stratégií, z čoho vyplýva, že s rastúcim počtom označených dát rástol rozdiel úspešnosti modelov A-optimality od ostatných a zároveň klesal rozdiel úspešnosti medzi modelmi *least confidence* a modelmi ostatných stratégií. Takýto vývoj pokračoval až do označenia 100 dát, kedy modely všetkých stratégií okrem A-optimality dosiahli priemerne 95%-nú úspešnosť a modely A-optimality priemerne 93%-nú úspešnosť. Modelmi vytvorenými náhodným pridávaním dát do trénovacej vzorky bola dosiahnutá priemerne len 91%-ná úspešnosť.

Zhrnutie

Pomocou nášho experimentu vykonaného na štyroch rôznych súboroch dát môžeme povedať, že vhodnosť použitia jednotlivých stratégií záleží od konkrétnych dát. Vychádzajúc z porovnania na dátach druhov Iris versicolor a virginica a na dátach diagnostiky rakoviny prsníka je zrejmé, že použitím ľubovoľnej stratégie dosiahneme lepšiu úspešnosť správneho zatriedenia dát, ako keby sme model vytvorili len náhodným výberom dát. Taktiež je zaujímavé všimnúť si, že modely A-optimality mali pri menšom počte označených dát vždy vyššiu úspešnosť ako modely ostatných stratégií a s vyšším počtom označených dát naopak, nižšiu úspešnosť, takže sa javí ako užitočné používať metódu A-optimality pri menšom počte označovaných dát. Modely *vote entropy* sa na dátach Iris versicolor a virginica správali presne opačne, zo začiatku s nižšou úspešnosťou a pri väčšom počte označených dát s vyššou úspešnosťou ako modely iných stratégií. Na dátach diagnostiky rakoviny prsníka však takéto správanie stratégie *vote entropy* nepozorujeme. Modely stratégií *least confidence*, *expected gradient length* a nami navrhnutá stratégia *expected classification entropy* boli vždy skoro rovnako úspešné, iba na dátach rakoviny prsníka bola stratégia *least confidence* s malým rozdielom lepšia.

Záver

V tejto diplomovej práci sme sa zaoberali sekvenčnými metódami klasifikácie dát. Cieľom bolo porovnať rôzne prístupy k sekvenčnej konštrukcii rozhodovacieho pravidla pre diskriminačnú analýzu založenú na logistickej regresii.

Najprv sme sa teda zamerali na logistickú regresiu, konkrétne binárnu logistickú regresiu a ako sa binárna logistická regresia dá využiť na vytvorenie klasifikačného pravidla. V oblasti sekvenčných metód na vytvorenie klasifikačného pravidla bola pre nás základom publikácia [17], v ktorej sa nachádza popis najpoužívanejších scenárov a stratégií výberu dát sekvenčných metód.

Venovali sme sa popisu troch základných scenárov a to *membership query synthesis*, *stream-based selective sampling* a *pool-based sampling*. Pri aplikácii jednotlivých stratégií na konkrétnych dátach sme používali scenár *pool-based sampling*, čiže zistili sme informatívnosť každého z doposiaľ neoznačených údajov a za najinformatívnejší sme zvolili ten, ktorý bol vybratý na označenie podľa danej stratégie.

Hlavnú časť práce tvorili práve stratégie výberu dát, na základe ktorých sa určuje informatívnosť údajov pre súčasne vytvorený klasifikátor. Nepracovali sme len s už známymi stratégiami, ale navrhli sme aj jednu vlastnú stratégiu s názvom *expected classification entropy* založenú na princípe odlišnom od ostatných stratégií. Na základe porovnania stratégií, kde sme sledovali správnosť zatriedenia dát pomocou klasifikačného pravidla vytvoreného porovnávanou stratégiou, sa *expected classification entropy* javí ako rozumne navrhnutá stratégia. Nie je síce úspešnejšou ako všetky ostatné stratégie, ale to sa nedá tvrdiť o žiadnej inej stratégii, pretože ich vhodnosť použitia záleží od konkrétnych dát. Túto prácu by bolo možné rozvinúť vo viacerých smeroch. Napríklad, okrem A-optimality je možné aplikovať aj ďalšie metódy známe z oblasti optimálneho navrhovania experimentov a na základe rozsiahlejšej simulačnej štúdie identifikovať pravidlá pre výber najvhodnejšej z týchto metód.

Literatúra

[1] Abe, N., Mamitsuka, H.: Query Learning Strategies Using Boosting and Bagging. In *Proceedings of the International Conference on Machine Learning (ICML)*, p. 1-9. Morgan Kaufmann, 1998.
URL: `http://webia.lip6.fr/~amini/RelatedWorks/Abe98.pdf`

[2] Angluin, D.: Queries and concept learning. In *Machine Learning*, vol. 2, p. 319-342. Kluwer Academic Publishers, 1988.
URL: `http://machinelearning202.pbworks.com/f/AngluinQueriesConceptLearningfulltext.pdf`

[3] Biopsia [online]. 2011. Accelerate. [cit. 20.4.2013].
URL: `http://www.vedietviac.sk/biopsia`

[4] Cohn, D. et al.: Training Connectionist Networks with Queries and Selective Sampling. In *Advances in Neural Information Processing Systems (NIPS)*, Morgan Kaufmann, 1990.

[5] Dagan, I., Engelson, S.: Committee-based sampling for training probabilistic classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*, p. 150-157. Morgan Kaufmann, 1995.
URL: `http://webia.lip6.fr/~amini/RelatedWorks/Dag95.pdf`

[6] Fisher, A. R.: The use of multiple measurments in taxonomic problems. In *Annals of Eugenics*, vol. 7, no. 2, p. 179-188, 1936.
URL: `http://onlinelibrary.wiley.com/doi/10.1111/j.1469-1809.1936.tb02137.x/pdf`

[7] Greene, H. W.: *Econometric analysis*. 5.vyd. New Jersey: Prentice Hall, 2003. ISBN 0-13-066189-9

[8] Hosmer, W. D., Lemeshow, S.: *Applied Logistic Regression*. 2.vyd. New York: John Wiley & Sons, Inc, 2000. ISBN 0-471-35632-8

[9] Chen, Y., Mani, S.: Study of active learning in the challenge. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, vol. 1, no. 7, p. 18-23, 2010
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5596776&isnumber=5595732

[10] Izenman, J. A.: *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. 1.vyd. New York: Springer, 2008. ISBN 978-0-387-78188-4

[11] Lang, K., Baum, E.: Query learning can work poorly when a human oracle is used. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, p. 335-340. IEEE Press, 1992.

[12] Lewis, D., Gale, W.: A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 3-12. ACM/Springer, 1994.
URL: http://arxiv.org/pdf/cmp-lg/9407020.pdf

[13] McCallum, A., Nigam, K.: Employing EM in pool-based active learning for text classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, p. 359-367. Morgan Kaufmann, 1998.
URL: http://www.kamalnigam.com/papers/emactive-icml98.pdf

[14] Nielsen, F.: Cramer-Rao Lower Bound and Information Geometry. In *Connected at Infinity II: On the work of Indian mathematicians (R. Bhatia and C.S. Rajan, Eds.)*, special volume of Texts and Readings In Mathematics (TRIM). Hindustan Book Agency, 2013.

[15] Pázman, A., Lacko, V.: *Prednášky z regresných modelov*. Bratislava: Univerzita Komenského v Bratislave, 2012. ISBN 978-80-223-3070-1

[16] Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the International Confe-*

rence on Machine Learning (ICML), p. 441-448. Morgan Kaufmann, 2001. URL: http://www-poleia.lip6.fr/~amini/RelatedWorks/Roy01.pdf

[17] Settles, B.: *Active Learning Literature Survey*, Computer Sciences Technical Report 1648. University of Wisconsin–Madison, 2009. URL: http://apophenia.wdfiles.com/local--files/start/settles_active_learning.pdf

[18] Settles, B., Craven, M., Ray, S.: Multiple-instance active learning. In *Advances in Neural Information Processing Systems (NIPS),* vol. 20, p. 1289-1296. MIT Press, 2008. URL: http://books.nips.cc/papers/files/nips20/NIPS2007_1072.pdf

[19] Seung, S. H., Opper, M., Sompolinsky, H.: Query by committee. In *Proceedings of the ACMWorkshop on Computational Learning Theory*, p. 287-294, 1992. URL: http://hebb.mit.edu/people/seung/papers/query.pdf

[20] Street, N. W., Wolberg, H. W., Mangasarian, L. O.: Nuclear feature extraction for breast tumor diagnosis. In *International Symposium on Electronic Imaging: Science and Technology*, vol. 1905, p. 861-870. San Jose, CA, 1993.

[21] Zhu, X.: *Semi-Supervised Learning with Graphs*. Dizertačná práca. Pittsburgh: Carnegie Mellon University, 2005. URL: http://pages.cs.wisc.edu/~jerryzhu/pub/thesis.pdf

Príloha 1

V nasledujúcom texte sa venujeme optimálnemu navrhovaniu experimentov, ktoré úzko súvisí so sekvenčnými metódami tvorby klasifikačného pravidla. Budeme vychádzať z [15]. Je zrejmé, že zvýšením počtu meraní dosiahneme zvýšenie presnosti odhadov v experimentoch opísaných regresným modelom. Väčšinou však máme počet meraní ohraničený (resp. máme obmedzené celkové náklady na experiment). Cieľom optimálneho navrhovania experimentov je práve navrhnúť, kde tieto pokusy realizovať. Vznikajú tak dve otázky, čo považovať za mieru presnosti odhadu a ako zabezpečiť, aby bola čo najväčšia. Pozrieme sa predovšetkým na prvý problém, ktorý vedie k voľbe kritérií optimality.

Pre každý pokus $\boldsymbol{x} \in \chi$, kde χ je množina všetkých možných pokusov, výsledok pokusu y_x získame z modelu

$$y_x = \boldsymbol{f}^T(\boldsymbol{x})\boldsymbol{\theta} + \epsilon_x,$$

kde

$$E(\epsilon_x) = 0, Var(\epsilon_x) = \sigma^2\lambda^{-1}(\boldsymbol{x}),$$

$\boldsymbol{\theta} \in \mathbb{R}^m$ je vektor neznámych parametrov a $\lambda(\boldsymbol{x})$ sa nazýva efektívnosť pokusu $\boldsymbol{x}$, pretože ak je $\lambda(\boldsymbol{x})$ veľké, meranie je presné a podstatne prispieva k určeniu odhadov $\boldsymbol{\theta}$. Návrh experimentu o rozsahu N je N-tica bodov $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ množiny χ. Niektoré z bodov $\boldsymbol{x}_i$ sa môžu opakovať, čiže pokusy v experimente môžeme opakovať, avšak jednotlivé pokusy sa vykonávajú nezávisle. Celý experiment môžeme prepísať v maticovom tvare $\boldsymbol{y} = \boldsymbol{F}\boldsymbol{\theta} + \boldsymbol{\epsilon}$, kde

$$\boldsymbol{y} = (y_{x_1}, \ldots, y_{x_N})^T,$$

$$\boldsymbol{\epsilon} = (\epsilon_{x_1}, \ldots, \epsilon_{x_N})^T,$$

$$\boldsymbol{F} = \begin{pmatrix} \boldsymbol{f}^T(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{f}^T(\boldsymbol{x}_N) \end{pmatrix},$$

$$E(\boldsymbol{\epsilon}) = \mathbf{0},$$
$$Var(\boldsymbol{\epsilon}) = \sigma^2 \boldsymbol{W} = \sigma^2 \begin{pmatrix} \lambda^{-1}(\boldsymbol{x}_1) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \lambda^{-1}(\boldsymbol{x}_N) \end{pmatrix}.$$

Cieľom experimentu je čo najpresnejšie odhadnúť parametre $\theta_1, \ldots, \theta_m$, čiže tak, aby disperzie odhadov alebo ich lineárnych kombinácií boli čo najmenšie. Z Gaussovej-Markovovej vety vyplýva, že najlepším lineárnym nevychýleným odhadom parametra $\boldsymbol{\theta}$ je odhad metódou najmenších štvorcov (MNŠ)

$$\hat{\boldsymbol{\theta}} = \boldsymbol{M}^{-1} \boldsymbol{F}^T (\sigma^2 \boldsymbol{W})^{-1} \boldsymbol{y},$$

kde

$$\begin{aligned} \boldsymbol{M} &= \boldsymbol{F}^T (\sigma^2 \boldsymbol{W})^{-1} \boldsymbol{F} \\ &= \sigma^{-2} \sum_{i=1}^{N} \boldsymbol{f}(\boldsymbol{x}_i) \boldsymbol{f}^T(\boldsymbol{x}_i) \lambda(\boldsymbol{x}_i) \end{aligned}$$

je informačná matica modelu. Otázkou je, ako zvoliť $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ tak, aby bola variancia tohto odhadu v istom zmysle čo najmenšia. Keďže $Var(\hat{\boldsymbol{\theta}}) = \sigma^2 \boldsymbol{M}^{-1}$, tak minimalizovať varianciu odhadu znamená maximalizovať informačnú maticu. K maximalizácii informačnej matice sa dá pristupovať rôzne. Ak počet meraní v bode $\boldsymbol{x}$ označíme $N(\boldsymbol{x})$, potom informačnú maticu môžeme prepísať nasledovne

$$\begin{aligned} \boldsymbol{M} &= \sigma^{-2} \sum_{\boldsymbol{x} \in \chi} N(\boldsymbol{x}) \boldsymbol{f}(\boldsymbol{x}) \boldsymbol{f}^T(\boldsymbol{x}) \lambda(\boldsymbol{x}) \\ &= \sigma^{-2} N \sum_{\boldsymbol{x} \in \chi} \xi(\boldsymbol{x}) \boldsymbol{f}(\boldsymbol{x}) \boldsymbol{f}^T(\boldsymbol{x}) \lambda(\boldsymbol{x}), \end{aligned}$$

kde $\xi(\boldsymbol{x}) = N(\boldsymbol{x})/N$. Konštanta N pri maximalizácii nezohráva žiadnu rolu, čiže môžeme ju vynechať. Ak zavedieme $\boldsymbol{f}^*(\boldsymbol{x}) = \lambda^{1/2}(\boldsymbol{x}) \boldsymbol{f}(\boldsymbol{x})$, tak informačná matica bude mať tvar $\sigma^{-2} \sum_{\boldsymbol{x} \in \chi} \boldsymbol{f}^*(\boldsymbol{x}) \boldsymbol{f}^{*T}(\boldsymbol{x}) \xi(\boldsymbol{x})$. Uvedomme si, že informačná matica je jednoznačne definovaná mierou $\xi(\boldsymbol{x})$. V prípade, ak by sme mali

ohraničené celkové náklady C na experiment a pre každý pokus $\boldsymbol{x} \in \chi$ stanovené náklady $c(\boldsymbol{x})$, informačná matica sa dá prepísať do tvaru

$$\begin{aligned}\boldsymbol{M}(\xi) &= \sigma^{-2} C \sum_{\boldsymbol{x}\in\chi} \frac{c(\boldsymbol{x})N(\boldsymbol{x})}{C} \boldsymbol{f}(\boldsymbol{x})\boldsymbol{f}^T(\boldsymbol{x}) \frac{\lambda(\boldsymbol{x})}{c(\boldsymbol{x})} \\ &= \sigma^{-2} C \sum_{\boldsymbol{x}\in\chi} \xi(\boldsymbol{x})\boldsymbol{f}^*(\boldsymbol{x})\boldsymbol{f}^{*T}(\boldsymbol{x}),\end{aligned}$$

kde $\xi(\boldsymbol{x}) = \frac{c(\boldsymbol{x})N(\boldsymbol{x})}{C}$ sú relatívne náklady pokusu v bode $\boldsymbol{x}$ a $f^*(\boldsymbol{x}) = \left(\frac{\lambda(\boldsymbol{x})}{c(\boldsymbol{x})}\right)^{1/2} f(\boldsymbol{x})$.

Definícia 2.1. *Hovoríme, že*

1. *návrh ξ je rovnomerne nie horší ako návrh η ($\xi \precsim \eta$), ak*

$$\forall h \in \mathbb{R}^m \qquad Var_\xi(h^T\hat{\boldsymbol{\theta}}) \leq Var_\eta(h^T\hat{\boldsymbol{\theta}})$$

2. *návrh ξ je ekvivalentný návrhu η ($\xi \sim \eta$), ak*

$$\forall h \in \mathbb{R}^m \qquad Var_\xi(h^T\hat{\boldsymbol{\theta}}) = Var_\eta(h^T\hat{\boldsymbol{\theta}})$$

3. *návrh ξ je rovnomerne lepší ako návrh η ($\xi \prec \eta$), ak platí 1. A zároveň*

$$\exists h^* \in \mathbb{R}^m \qquad Var_\xi(h^{*^T}\hat{\boldsymbol{\theta}}) < Var_\eta(h^{*^T}\hat{\boldsymbol{\theta}}).$$

Ideálne by bolo nájsť návrh, ktorý je rovnomerne najlepší, to sa však dá splniť v máloktorej úlohe.

Kritériá optimality

V tejto podkapitole sa pozrieme na tie najbežnejšie aplikované kritériá optimality. Najskôr však uvedieme definíciu kritéria optimality.

Definícia 2.2. *Kritériom optimality sa nazýva funkcia ϕ, pre ktorú platí, že ak $\xi \precsim \eta$, čiže $\boldsymbol{M}(\xi) \succeq \boldsymbol{M}(\eta)$, tak $\phi(\boldsymbol{M}(\xi)) \leq \phi(\boldsymbol{M}(\eta))$.*

Okrem splnenia definície sa vyžaduje aj štatistická interpretácia kritéria.

Definícia 2.3. *Návrh ξ^* sa nazýva ϕ optimálny, ak platí*

$$\phi(\boldsymbol{M}(\xi^*)) = \min_{\xi} \phi(\boldsymbol{M}(\xi)).$$

Medzi najznámejšie kritériá optimality patria:

- **D-optimalita**

$$\phi(\boldsymbol{M}) = \begin{cases} -\ln(det(\boldsymbol{M})), & \boldsymbol{M} \text{ je regulárna} \\ +\infty, & \text{inak.} \end{cases}$$

- **A-optimalita**

$$\phi(\boldsymbol{M}) = \begin{cases} tr(\boldsymbol{M}^{-1}), & \boldsymbol{M} \text{ je regulárna} \\ +\infty, & \text{inak.} \end{cases}$$

- **c-optimalita**

$$\phi(\boldsymbol{M}) = \begin{cases} \boldsymbol{h}^T \boldsymbol{M}^{-} \boldsymbol{h}, & \boldsymbol{h} \in S(\boldsymbol{M}) \\ +\infty, & \text{inak.} \end{cases}$$

- **E-optimalita**

$$\phi(\boldsymbol{M}) = \begin{cases} \frac{1}{\lambda_{min}(\boldsymbol{M})} = \lambda_{max}(\boldsymbol{M}^{-1}), & \boldsymbol{M} \text{ je regulárna} \\ +\infty, & \text{inak,} \end{cases}$$

  kde $\lambda_{min}(\boldsymbol{M})$ je najmenšie vlastné číslo informačnej matice $\boldsymbol{M}$ a $\lambda_{max}(\boldsymbol{M}^{-1})$ je najväčšie vlastné číslo inverzie tejto matice.

- **G-optimalita**

$$\phi(\boldsymbol{M}) = \begin{cases} \max_{\boldsymbol{x} \in \chi} \boldsymbol{f}^T(\boldsymbol{x}) \boldsymbol{M}^{-1} \boldsymbol{f}(\boldsymbol{x}), & \boldsymbol{M} \text{je regulárna} \\ +\infty, & \text{inak.} \end{cases}$$

D-optimalita je najpopulárnejším kritériom optimality v optimálnom navrhovaní experimentov. Aplikovaním tohto kritéria minimalizujeme objem elipsoidu spoľahlivosti pre neznáme parametre $\boldsymbol{\theta}$. Takýmto spôsobom dochádza k minimalizácii kovariančnej matice $cov(\hat{\boldsymbol{\theta}})$ a teda k presnejšiemu odhadu parametrov. A-optimalitou zase znižujeme priemernú hodnotu variancií odhadov neznámych parametrov.

Príloha 2

Nasledujúci kód v programovacom jazyku "R" sme použili pri výpočte percenta správne zatriedených dát modelom používajúcim zvolenú stratégiu. Výstupom je vektor "percento_spravne_zaradenych", ktorého prvá hodnota predtavuje percento správne zatriedených dát inicializačným modelom a každá ďalšia i-ta hodnota predstavuje percento správne zatriedených dát modelom vytvoreným na inicializačnej vzorke, ku ktorej bolo pridaných $i - 1$ hodnôt.

```
#vstupy:

#metoda = nazov metody: "nahodne", "LC", "VE", "EGL", "A_optimalita" alebo "ECE"
#x = matica, ktora ma v riadkoch vektory "crt"
#y = vektor kategorii (hodnoty 1 alebo 2)
#pocet_oznacenych_na_zaciatku = kolko dat bude nahodne vybratych na vytvorenie
    inicializacneho modelu
#hranica = kolko dalsich dat postupne pridame do trenovacej vzorky

#zadavame len v pripade VE:
#K = kolko roznych modelov chceme vytvorit na vzorke doteraz oznacenych dat

#zadavame len v pripade ECE:
#N = pocet novovygenerovanych dat
#stand_dev = standardna odchylka norm. rozdeleni pri vytvarani novych dat


active_learning<-function(metoda, x, y, pocet_oznacenych_na_zaciatku, hranica, K, N,
    stand_dev){

  nahodne="nahodne"
  LC<-"LC"
  VE<-"VE"
  EGL<-"EGL"
  EER<-"EER"
  EER_logloss<-"EER_logloss"
  A_optimalita<-"A_optimalita"
  ECE<-"ECE"

  pocet_kategorii<-2

  n<-length(x[,1])

  #pocet koeficientov beta
  pocet_parametrov<<-length(x[1,])+1
```

```
#funkcia na pretrenovanie modelu na doposial oznacenych datach L
#vystupom je betaHAT_L
pretrenuj<-function(L)
{
MODEL_L <- glm((y[L]-1)~1+x[L,], family=binomial(link="logit"))
betaHAT_L <- MODEL_L$coef
return(betaHAT_L)
}


#funkcia na najdenie minima ucelovej hodnoty
najdi_index_minima<-function(hodnoty_ucelovej_funkcie)
{
minimum<-min(hodnoty_ucelovej_funkcie,na.rm=TRUE)
#najdenie data (jeho poradia v U) s minimalnou hodnotou ucelovej funkcie
najneistejsi<-which(hodnoty_ucelovej_funkcie==minimum)
#najdenie data s minimalnou hodnotou ucelovej funkcie
index<-U[najneistejsi[1]]
    return(index)

}


#funkcia na vypocet poctu spravne zaradenych dat na zaklade modelu s parametrom beta
vrat_pocet_spravne_zaradenych<-function(beta)
{
#matica X pre vsetky data
X_vsetky<-cbind(rep(1,n),x)
#vektor e^(-beta*X^T)
e_vsetky<-exp(-beta%*%t(X_vsetky))
#vektor pravdepodobnosti zaradenia dat medzi modre
p2_vsetky<-as.vector(1/(1+e_vsetky))
#vektor s 1 na mieste tych dat, ktore sa podla sucasneho modelu zaraduju medzi modre
vektor_modrych<-(p2_vsetky>0.5)*1
#pocet spravne zaradenych dat
pocet_spravne_zaradenych<-sum(vektor_modrych==(y-1))
return(pocet_spravne_zaradenych)
}


#spravny model
MODEL <- glm((y-1)~1+x, family=binomial(link="logit"))
betaHAT <- MODEL$coef

#percento spravne zaradenych dat v modeli, kde oznacime vsetky data
percento_spravne_zaradenych_vsetky_data<<-vrat_pocet_spravne_zaradenych(betaHAT)/n*100

#neoznacene data
U<-c(1:n)

#oznacene data (na zaciatku sa nahodne vyberie "pocet_oznacenych_na_zaciatku" dat)
L_cervene<-sample(c(1:n)[y==1],pocet_oznacenych_na_zaciatku/2)
```

```
L_modre<-sample(c(1:n)[y==2],pocet_oznacenych_na_zaciatku/2)
L<-c(L_cervene,L_modre)

#ktory udaj sa prida do L
index<-NULL

#neoznacene data (uz bez tych 10 nahodne vybratych)
U<-U[-L]

#pretrenovanie a vykreslenie modelu
betaHAT_L <- pretrenuj(L)

percento_spravne_zaradenych<-c(rep(0,hranica+1))

#percento spravne zaradenych
percento_spravne_zaradenych[1]<-vrat_pocet_spravne_zaradenych(betaHAT_L)/n*100



for (p in 1:hranica)
{
#metoda, ktora na oznacenie zvoli lubovolny udaj z U
if(metoda==nahodne)
{
  index<-sample(U,1)
}


#metoda "Least confidence"
if(metoda=="LC")
{
  #pole s hodnotami ucelovej funkcie pre vsetky data z U
  hodnoty_ucelovej_funkcie<-c(rep(0,max(U)))
  #hladanie najneistejsieho zo vsetkych neoznacenych pre sucasny model
  for (i in U)
  {
    #e^(-beta*x)
    e <- exp(-betaHAT_L%*%c(1,x[i,]))
    #pravdepodobnost, ze udaj i patri medzi modre
    p2<-1/(1+e)
    #pravdepodobnost, ze udaj i patri medzi cervene
    p1<-1-p2
    #max(p1,p2)=pravdepodobnost pravdepodobnejsej kategorie
    hodnoty_ucelovej_funkcie[i]<-max(p1,p2)
  }
  hodnoty_ucelovej_funkcie<-hodnoty_ucelovej_funkcie[U]
  #najdenie minima ucelovej hodnoty
  index<-najdi_index_minima(hodnoty_ucelovej_funkcie)
}


#metoda "Vote entropy"
```

```
if(metoda==VE)
{
  #V_1=pocet hlasov zaradenia do 1. (cervenej) kategorie pre kazdy udaj z U
  V_1<-c(rep(0,length(U)))

  #vytvorenie K roznych modelov a zratanie hlasov
  for (k in 1:K)
  {
    #nahodny vyber z L na vytvorenie Lk=trenovacia vzorka pre k-ty model
    Lk<-NULL
    while (length(unique(x[Lk,])[,1])<pocet_parametrov)
    {
      na_vyber_cervene<-L[y[L]==1]
      na_vyber_modre<-L[y[L]==2]
      Lk_cervene<-sample(na_vyber_cervene,length(na_vyber_cervene),replace=TRUE)
      Lk_modre<-sample(na_vyber_modre,length(na_vyber_modre),replace=TRUE)
      Lk<-unique(c(Lk_cervene,Lk_modre))
    }

    #natrenovanie modelu na Lk
    betaHAT_Lk <- pretrenuj(Lk)

    poradie_v_U<-0
    #vypocet poctu hlasov 1. kategorie
    for (i in U)
    {
      poradie_v_U<-poradie_v_U+1
      #e^(-beta_Lk*x)
      e <- exp(-betaHAT_Lk%*%c(1,x[i,]))
      #p1=pravdepodobnost, ze udaj i patri do 1. kategorie podla modelu Lk
      p1<-1-1/(1+e)
      #ak je p1>0.5, priratame hlas pre 1. kategorii pre udaj i
      if (p1 > 0.5) V_1[poradie_v_U]<-V_1[poradie_v_U]+1
    }
  }
  #vypocet poctu hlasov 2. kategorie
  V_2<-K-V_1

  #hodnoty_ucelovej_funkcie=vektor hodnot ucelovej funkcie pre kazdy udaj z U (hladame
      min)
  hodnoty_ucelovej_funkcie<-V_1/K*log(V_1/K)+V_2/K*log(V_2/K)
  #najdenie minima ucelovej hodnoty
  index<-najdi_index_minima(hodnoty_ucelovej_funkcie)
  #ak sa modely nelisili v ziadnom udaji
  if (is.na(index)) index<-sample(U,1)
}


#metoda "Expected Gradient Length"
if(metoda==EGL)
{
  #pole s hodnotami ucelovej funkcie pre vsetky data z U
```

```
  hodnoty_ucelovej_funkcie<-c(rep(0,max(U)))
  for (i in U)
  {
    #e^(-beta*x)
    e <- exp(-betaHAT_L%*%c(1,x[i,]))
    #pravdepodobnost zaradenia medzi modre
    p2<-1/(1+e)
    #pravdepodobnost zaradenia medzi cervene
    p1<-1-p2
    #hladame maximalne p1*p2*norma(x) (minimalne -p1*p2*norma(x))
    hodnoty_ucelovej_funkcie[i]<-(-p1*p2*sqrt(1+sum(x[i,]^2)))
  }
  hodnoty_ucelovej_funkcie<-hodnoty_ucelovej_funkcie[U]
  #najdenie minima ucelovej hodnoty
  index<-najdi_index_minima(hodnoty_ucelovej_funkcie)
}


#metoda A-optimalita
if(metoda==A_optimalita)
{
  #pole s hodnotami ucelovej funkcie pre vsetky data z U
  hodnoty_ucelovej_funkcie<-c(rep(0,max(U)))
  #matica X pre vsetky data z L
  X_L<-cbind(rep(1,length(L)),x[L,])
  #vektor e^(-beta*X_L^T)
  e_L<-exp(-betaHAT_L%*%t(X_L))
  #vektor pravdepodobnosti zaradenia dat L medzi modre
  p2_L<-1/(1+e_L)
  #vektor pravdepodobnosti zaradenia dat L medzi cervene
  p1_L<-1-p2_L
  for (i in U)
  {
    #k X_L sa prida dato i - vytvoria maticu X
    X<-rbind(X_L,c(1,x[i,]))
    #e^(-beta*x)
    e <- exp(-betaHAT_L%*%c(1,x[i,]))
    #pravdepodobnost, ze udaj i patri do 2.kategorie podla modelu MODEL_L
    p2<-1/(1+e)
    #pravdepodobnost, ze udaj i patri do 1.kategorie podla modelu MODEL_L
    p1<-1-p2
    #vytvorenie diagonalnej matice W s prvkami p1*p2 na diagonale
    W<-diag(c(as.vector(p1_L*p2_L),p1*p2))
    #I=Fisherova informacna matica
    I<-t(X)%*%W%*%X
    #I^(-1)
    otestuj<-try(solve(I),TRUE)
    if(isTRUE(all.equal(class(otestuj),"try-error")))
      hodnoty_ucelovej_funkcie[i]<-NaN
    else
    {
      inverzna_I<-solve(I)
```

```
      #hodnota ucelovej funkcie = stopa (I^(-1))
      hodnoty_ucelovej_funkcie[i]<-stopa<-sum(diag(inverzna_I))
    }
  }
  hodnoty_ucelovej_funkcie<-hodnoty_ucelovej_funkcie[U]
  #najdenie minima ucelovej hodnoty
  index<-najdi_index_minima(hodnoty_ucelovej_funkcie)
}


#metoda ECE=Expected Classification Entropy
if(metoda==ECE)
{
  #pole s hodnotami ucelovej funkcie pre vsetky data z U
  hodnoty_ucelovej_funkcie<-c(rep(0,max(U)))
  #povodna kategoria novovygenerovanych dat (potrebna len k metode ECE)
  povodna_kategoria_nove<-c(rep(0,N))
  #matica novovygenerovanych dat
  x_nove<-matrix(0,N,length(x[1,]))
  #vytvorime N novych nahodnych dat
  for(k in 1:N)
  {
    #nahodne_vybrate=udaj, okolo ktoreho sa vytvori novy udaj
    nahodne_vybrate<-sample(n,1)
    #prva suradnica noveho udaja
    for(index_suradnice in 1:length(x[1,]))
    {
      x_nove[k,index_suradnice]<-rnorm(1,mean=x[nahodne_vybrate,index_suradnice],sd=
          stand_dev)
    }

  }
  for (i in U)
  {
    #e^(-beta*x)
    e <- exp(-betaHAT_L%*%c(1,x[i,]))
    #pravdepodobnost, ze udaj i patri do 2.kategorie podla modelu MODEL_L
    p2<-1/(1+e)
    #pravdepodobnost, ze udaj i patri do 1.kategorie podla modelu MODEL_L
    p1<-1-p2
    #pretrenujeme model, ak by i patrilo do cervenej
    MODEL_L_ak_cervena <- glm(c(y[L]-1,0)~1+x[c(L,i),], family=binomial(link="logit"))
    betaHAT_L_ak_cervena <- MODEL_L_ak_cervena$coef
    #pretrenujeme model, ak by i patrilo do modrej
    MODEL_L_ak_modra <- glm(c(y[L]-1,1)~1+x[c(L,i),], family=binomial(link="logit"))
    betaHAT_L_ak_modra <- MODEL_L_ak_modra$coef
    for (k in 1:N)
    {
      e_nove_ak_i_cervene <- exp(-betaHAT_L_ak_cervena%*%c(1,x_nove[k,]))
      #pravdepodobnost, ze udaj k patri medzi modre, ak by i bolo cervene
      p2_nove_ak_i_cervene<-1/(1+e_nove_ak_i_cervene)
      #pravdepodobnost, ze udaj k patri medzi cervene, ak by i bolo cervene
```

```
        p1_nove_ak_i_cervene<-1-p2_nove_ak_i_cervene
        #zistime kategoriu udaju k, ak by i bolo cervene
        if (p2_nove_ak_i_cervene>0.5) kategoria_nove_ak_i_cervene<-"modra"
        else kategoria_nove_ak_i_cervene<-"cervena"

        e_nove_ak_i_modre <- exp(-betaHAT_L_ak_modra%*%c(1,x_nove[k,]))
        #pravdepodobnost, ze udaj k patri medzi modre, ak by i bolo modre
        p2_nove_ak_i_modre<-1/(1+e_nove_ak_i_modre)
        #pravdepodobnost, ze udaj k patri medzi cervene, ak by i bolo modre
        p1_nove_ak_i_modre<-1-p2_nove_ak_i_modre
        #zistime kategoriu udaja k, ak by i bolo modre
        if (p2_nove_ak_i_modre>0.5) kategoria_nove_ak_i_modre<-"modra"
        else kategoria_nove_ak_i_modre<-"cervena"

        #hodnoty ucelovej funkcie zvysujeme len ak su rozdielne kategorie pre k v
            zavislosti
        #od toho, ci je i modre alebo cervene
        if(kategoria_nove_ak_i_cervene != kategoria_nove_ak_i_modre)
          hodnoty_ucelovej_funkcie[i]<-hodnoty_ucelovej_funkcie[i]+p1*log(p1)+p2*log(p2)
      }
    }

    hodnoty_ucelovej_funkcie<-hodnoty_ucelovej_funkcie[U]
    #najdenie minima ucelovej hodnoty
    index<-najdi_index_minima(hodnoty_ucelovej_funkcie)
  }



  #pridanie udaja index do L
  L<-c(L,index)

  #odobratie udaja index z U
  if (!is.null(index))  U<-setdiff(U,index)

  #pretrenovanie modelu na novom L
  betaHAT_L <- pretrenuj(L)

  #percento spravne zaradenych
  percento_spravne_zaradenych[p+1]<-vrat_pocet_spravne_zaradenych(betaHAT_L)/n*100
  }

  return(percento_spravne_zaradenych)
}
```