UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Ivan Drljepan

Klasifikacija s porazdeljeno predstavitvijo

DIPLOMSKO DELO NA UNIVERZITETNEM ŠTUDIJU

Mentor: izr. prof. dr. Marko Robnik-Šikonja

Ljubljana, 2013

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00060/2012
Datum: 06.11.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **IVAN DRLJEPAN**

Naslov: **KLASIFIKACIJA S PORAZDELJENO PREDSTAVITVIJO**
DISTRIBUTED REPRESENTATION BASED CLASSIFICATION

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

Podobnost je temelj induktivnega sklepanja in učenja. Za napovedovanje ta princip v direktni obliki uspešno uporablja metoda najbližjih sosedov. Težava pristopa je, da postane z naraščanjem števila učnih primerov počasen, z naraščanjem števila dimenzij problema pa tudi nezanesljiv. Počasnemu iskanju podobnih primerov se poskušamo izogniti tako, da namesto direktne primerjave z vektorji predstavljenih učnih primerov izračunamo posebne zgoščevalne funkcije, ki podobne primere preslikajo v podobne kode.
Implementirajte klasifikacijsko metodo najbližjih sosedov, ki uporablja porazdeljeno predstavitev, temelječo na zgoščevalnih funkcijah, za hitro iskanje podobnih primerov. Rešitev eksperimentalno ovrednotite na več različnih podatkovnih množicah in jo primerjate s klasičnim pristopom. Ugotovite prednosti in slabosti predstavljene metode ter določite pogoje, pri katerih je njena uporaba koristna.

Mentor:
prof. dr. Marko Robnik Šikonja

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:
prof. dr. Nikolaj Zimic

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani **Ivan Drljepan**,

z vpisno številko **63960080**,

sem avtor diplomskega dela z naslovom:

Klasifikacija s porazdeljeno predstavitvijo

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom

  prof. dr. Marka Robnik-Šikonje,

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,

- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne _____ Podpis avtorja: _____

ZAHVALA

Zahvaljujem se mentorju prof. dr. Marku Robnik-Šikonji za pomoč, svetovanje in vodenje pri izdelavi diplomskega dela.

Kazalo

Povzetek

Strojno učenje se srečuje z vedno večjimi množicami podatkov. Nekatere uspešne metode za reševanje takih problemov potrebujejo preveč časa in/ali prostora, da bi bila njihova uporaba načrtna.

Cilj diplomskega dela je implementacija in testiranje klasifikacijske metode s porazdeljeno predstavitvijo, pri kateri je hitrost klasifikacije neodvisna od števila učnih primerov. Pokazali smo, da implementacija, ki ohranja konstantnost časa klasifikacije, zahteva pri visokodimenzionalnih problemih preveč prostora, da bi bila praktično izvedljiva. Z uporabo razpršenih tabel smo ohranili skorajda konstantno hitro klasifikacijo pri nizkodimenzionalnih problemih. To je mogoče zaradi majhne porabe pomnilnika, ki je pri tej metodi odločilnega pomena za hitrost klasifikacije, vendar pa pri nizkodimenzionalnih problemih z velikim številom učnih primerov klasifikacijska točnost zaradi nasičenosti prostora pade. Pri več dimenzijah se točnost izboljša na račun večje porabe pomnilnika in časa.

Empirična evaluacija je pokazala, da je v primerjavi s sorodno metodo najbližjih sosedov klasifikacija s porazdeljeno predstavitvijo hitrejša in manj prostorsko zahtevna, medtem ko pri klasifikacijski točnosti med njima nismo izmerili statističnih razlik. Ugotovili smo, da je metoda primerna za sekvenčne probleme in da obstajajo problemi, ki so za to metodo povsem neprimerni. Metoda tako ni povsem splošna, a pod določenimi pogoji lahko reši problem hitreje, porabi za to manj prostora in ohranja primerljivo točnost klasifikacije.

Ključne besede: strojno učenje, klasifikacija, razpršene tabele, "big data", porazdeljena predstavitev

Abstract

Machine learning is increasingly met with datasets that require learning on a large number of learning samples. In solving these problems, some successful methods require too much time and/or space, for them to be viable.

The aim of the thesis was the implementation and testing of the distributed representation based classification method of which classification speed is independent of the number of learning samples. We show that an implementation, which preserves a constant classification time, in case of high-dimensional problems requires too much space for it to be practical. By using hash tables we preserved an almost constant, fast classification for low-dimensional problems. It is made possible by a low memory consumption which is crucial for this method's classification speed. However, with low-dimensional problems, high number of learning samples causes learning saturation, which results in a drop of the classification rate. With more dimensions classification rate improves, but on account of higher memory consumption and longer classification time.

Empirical evaluation has shown that, compared to the related nearest neighbors method, distributed representation based classification is faster and uses less space, while classification rates show no statistically significant differences. We determined that the method is suitable for sequential problems and that there are existing problems which are entirely unsuitable for it. Thus the method does not offer a general solution, however, under certain circumstances, it can solve problems faster, requires less space and at the same time maintain comparable classification rate.

Key words: machine learning, classification, hash tables, big data, distributed representation

1. Uvod

Strojno učenje [6, pogl. 1] je veja raziskav umetne inteligence. Med drugim se uporablja za analizo podatkov in odkrivanje zakonitosti v podatkovnih bazah, za avtomatsko generiranje baz znanja za ekspertne sisteme, za razpoznavanje naravnega jezika in prevajanje, klasifikacijo tekstov in spletno rudarjenje, razpoznavanje govora, pisave, slik itd.

Osnovni princip strojnega učenja je avtomatsko opisovanje pojavov iz podatkov. Rezultat učenja so pravila, funkcije, relacije, sistemi enačb, verjetnostne porazdelitve ipd. Naučeni modeli poskušajo razlagati podatke, iz katerih so bili zgenerirani in se lahko uporabljajo za odločanje pri opazovanju modeliranega procesa v bodočnosti (napovedovanje, diagnosticiranje, nadzor, preverjanje, simulacije itd.).

Ena najpogostejših uporab strojnega učenja je klasifikacija ali uvrščanje. Naloga klasifikatorja je za problem, opisan z množico atributov, določiti, kateremu izmed možnih razredov pripada. Atributi so neodvisne številske ali diskretne spremenljivke, s katerimi opisujemo probleme, razred pa je odvisna diskretna spremenljivka, ki ji določimo vrednost glede na vrednost neodvisnih spremenljivk.

Klasifikatorje ločimo glede na način predstavitve funkcije, ki preslika prostor atributov v razred. Najbolj pogosti klasifikatorji so odločitvena drevesa, odločitvena pravila, naivni Bayesov klasifikator, klasifikator z najbližjimi sosedi itd.

Z razvojem tehnologije in rastjo procesorske moči so nekateri klasifikacijski problemi postali obvladljivi in želimo jih rešiti s čim bolj uspešnimi metodami klasifikacije. Znano je, da nobena metoda ni najboljša na vseh problemih. Prav tako ni metode, ki bi bila hitra, natančna, fleksibilna in varčna s prostorom.

Danes količina podatkov, ki jih človeštvo proizvaja, raste z neverjetno hitrostjo. Srečujemo se z nepregledno količino podatkov, ki jim pravimo "big data". Nekatere metode, ki učinkovito delujejo na problemih z nekaj sto primeri, odpovejo, ko se lotimo problemov z nekaj milijoni primerov. Želimo metode, ki delujejo ne glede na količino podatkov.

Kobayashi in Nakagawa sta predstavila metodo klasifikacije [4], ki daje spodbudne rezultate, saj obljublja hitro učenje in klasifikacijo tudi pri problemih z veliko učnimi primeri. Po uspešnosti se lahko postavi ob bok metodi k-najbližjih sosedov, ki pa se pri mnogo učnih primerih zelo upočasni pri klasifikaciji. Osnovna ideja je, da vektor značilk predstavimo z

množico vektorjev, tako da ga razpršimo v eksponentni prostor (N-1 N-dimenzionalnih prostorov). Metoda temelji na funkciji podobnosti, ki jo algoritem uporablja za odločanje.

Metoda je linearna glede na funkcijo podobnosti in deluje nad numeričnimi atributi, saj jih je treba ovrednotiti in jim določiti vrstni red glede na vrednost. Vsak primer spremenimo v vektor, katerega elementi so permutacija N vrednosti, ki določa, kako se bo vektor razpršil. Vsak vektor ima identičnih N elementov.

Zgornji odstavek je ključen za razumevanje delovanja metode. V drugem poglavju je metoda razložena natančneje. Kakšne težave lahko nastopijo ob dobesedni interpretaciji ideje, je opisano v tretjem poglavju, kjer je predstavljen potek implementacije metode, ki je bila v prvotni izvedbi izjemno požrešna tako prostorsko kot časovno.

Učinkovitost metode smo testirali na več množicah podatkov. Preizkusili smo tudi nekaj modifikacij. Rezultati se nahajajo v četrtem poglavju, v katerem so izpostavljene dobre in slabe lastnosti metode in primerjava rezultatov z drugimi znanimi metodami klasifikacije. Rezultate razložimo glede primernosti metode in parametrov njene uspešnosti.

Na koncu s pomočjo analize vseh rezultatov obravnavamo konkurenčnost in delovanje metode, ko je ta soočena z veliko količino učnih primerov. Vključimo tudi razmislek o možnih izboljšavah algoritma.

2. Klasifikacija z razpršenimi funkcijami

Za metode strojnega učenja po navadi velja, da so pri večjem številu učnih primerov natančnejše. Večje število učnih primerov po navadi pomeni tudi večjo kompleksnost preiskovalnega prostora in s tem daljše čase odločanja, raste pa tudi prostorska zahtevnost.

Metoda razpršenega kodiranja ima načeloma lastnost, da število učnih primerov ne vpliva na hitrost odločanja. Poglejmo, kako je metoda zasnovana.

2.1. Razpršena predstavitev vektorjev

Naj bo x vektor značilk, prostor Q pa naj bo definiran kot diskretni hipersferični prostor.

Q(x,p) predstavlja p najbližjih sosedov vektorja x v prostoru Q.

Pregledamo lahko N_Q med seboj različnih prostorov $Q_0, Q_1 \dots, Q_{N_Q-1}$, ki kreirajo $Q_k(x,p_k)$, kjer k = 0 … N_Q-1. Tedaj je izbranih N_e elementov.

$$N_e = \left| \sum_k Q_k(x, p_k) \right|$$

$$\{e_j\} \equiv \cup\, Q_k(x, p_k) \;\; (j = 0 \dots, N_e - 1)$$

Množico $\{e_j\}$ imenujemo "razpršena predstavitev x" in vsebuje njegovo informacijo.

Zanimajo nas primeri, za katere velja:

$$x \;\cong\; \alpha \sum_j e_j$$

ter Q in p, ki zadoščata zgornjemu izrazu. Koeficient α tu nima posebnega pomena, ker so vektorji e_j v hipersferi.

2.2. Matematično ozadje

Naj bo v vektor v n-dimenzionalnem hipersferičnem prostoru S. Hipersfera ali n-sfera je posplošena površina navadne sfere v poljubni dimenziji. Za poljubno naravno število n je hipersfera definirana kot množica točk v (n+1)-dimenzionalnem evklidskem prostoru, ki so oddaljene od središča na razdalji r, kjer je r poljubno realno število in predstavlja radij.

$$S^n = \{\, x \in \mathbb{R}^{n+1} : \|x\| = r \,\}$$

V prostoru S izberemo vektor v. Velja:

$$W_n(\theta) = \{u|(u,v)/|u| \cdot |v| \leq \theta, u \in S\}$$

$$W_n(\pi) = S$$

Vrednost $W_n(\theta)/W_n(\pi)$ se giblje med 0 in 1 ter postaja vse večja, ko narašča θ. Pri večjih n močno narašča, ko je θ blizu π /2. Temu pravimo tendenca k ortogonalnosti.

Pri danih točkah x_1 in x_2 lahko za mero podobnosti uporabimo naslednji izraz:

$$\left| \left(\bigcup_k Q_k(x_1, p_k) \right) \cap \left(\bigcup_k Q_k(x_2, p_k) \right) \right|$$

Zaradi tendence k ortogonalnosti v večdimenzionalnih prostorih je vrednost bolj zanesljiva, ko je razdalja med x_1 in x_2 majhna. Glede na to lahko sestavimo takšno podatkovno strukturo za prepoznavanje vzorcev, da vedno obstaja učni vzorec blizu neznanemu vhodnemu vzorcu.

Če so točke v prostoru razdeljene naključno, je verjetnost obstoja točke v nekem lokalnem območju odvisna od povprečne gostote točk. Če je porazdelitev gosta, je verjetnost za obstoj točke statistično stabilna. Da zagotovimo zadovoljivo gostoto, moramo zato pripraviti dovolj veliko število učnih primerov.

2.3. Metoda podobnosti v eksponentnem prostoru (*power space similarity method*)

Definiramo razpršeno predstavitev:

$$x \in N!^{\pm}, Q_k = {}_N C_1^{\pm} \ \ldots, \ {}_N C_{N-1}^{\pm},$$

kjer velja:

$$N!^{\pm} \equiv \left\{ x = (x_0, x_1 \ \ldots, x_{N-1}) | x_i \in \{2k - N + 1\} (k = 0 \ldots, N-1), x_i \neq x_j (i \neq j) \right\}$$

$${}_N C_k^{\pm} \equiv \{x | x \in \{-1, +1\}^N, |x| = 2k - N\} \ (k = 1, \ldots, N-1)$$

Element a, za katerega velja $Q_k(x) = \{a\}$, je določen s pravilom:

$$a = (a_i), x = (x_i), a_i = \begin{cases} +1; & x_i \geq N + 1 - 2k \\ & \\ -1; & \text{sicer} \end{cases} \qquad (i = 1 \ldots, N)$$

Na ta način dobimo točke x_0, x_1 ..., x_{N-1}, ki predstavljajo najbližje sosede x v ${}_NC_1^{\pm}$, ${}_NC_2^{\pm}$ ..., ${}_NC_{N-1}^{\pm}$

Na tem mestu definiramo osnovni algoritem za prepoznavanje vzorcev. Q(x,p) predstavimo kot element množice $\{0,1\}^{|Q|}$. Če take binarne vektorje naredimo v Q_k prostoru, dobimo visoko dimenzionalni binarni vektor tako, da vse te vektorske elemente sestavimo. Tako dobimo:

$$b = (b_i) \in \{0,1\}^{\sum |Q_k|}$$

Ko x tako preslikamo v visoko dimenzionalni prostor, kjer je dimenzija večja od števila učnih primerov, ta vedno postane linearno ločljiv. Tako je možno zgraditi linearno diskriminantno funkcijo z vhodnim nivojem $\sum|Q_k|$ in izhodnim nivojem N_c elementov, kjer je N_c število razredov. Razrede definiramo kot C_j (j = 0 ..., N_C-1).

Da bi prepoznali neznani vzorec x, ga najprej pretvorimo v b in izračunamo izraz:

$$\text{score}(x, C_j) = \sum_i b_i w_{ij}$$

Razred C_k z zadovoljivim $k = \underset{j}{\text{argmax}}\{\text{score}(x, Cj)\}$ je rezultat.

w_{ij} so koeficienti uteži, ki omogočajo, da so učni primeri prepoznani. Dobimo jih po spodnjem postopku.

Vse w_{ij} postavimo na 0, nato pa vsak učni primer x, ki spada v razred C_i, pretvorimo v binarni vektor b. Če je element b_j vektorja b enak 1, postavimo vrednost w_{ij} na 1.

Za učne primere pri izračunu score() dobimo najvišji možni rezultat, in tako vedno prepoznamo pravilni razred, razen če dobimo najvišji možni rezultat pri več kot enem razredu.

Če vzamemo, da je število učnih primerov N_L, potem obstaja algoritem s časom učenja $O(N_e*N_L)$ in čas klasifikacije $O(N_e)$. Velikost tako dobljene predstavitve ne presega $O(N_e*N_L)$. Se pravi, da je čas učenja sorazmeren številu učnih primerov, čas klasifikacije pa je konstanten in neodvisen od velikosti slovarja.

2.4. Primer razpršenega kodiranja

Vzamemo poljuben vektor značilk v prostoru R^5: v = (302,111, 92, 1093, 28) in ga kvantiziramo v vektor x v prostoru $N!^{\pm}$. Prostor $5!^{\pm}$ je sestavljen iz permutacij vrednosti –4, –2, 0, 2, 4.

Poiščemo permutacijo, ki ima enak vrstni red elementov, kot ga ima vektor značilk.

$x = (2, 0, -2, 4, -4) \in 5!^{\pm}$

Poiščemo razpršeno predstavitev:

$$e_k = (a_1 ..., a_N); k = 1 ... N\text{-}1$$

$$a_i = \begin{cases} +1; x_i \geq N + 1 - 2k \\ -1; \text{sicer} \end{cases}$$

Razpršena predstavitev vektorja x je torej:

$e_1 = (-1, -1, -1, +1, -1) \in {}_5C_1^{\pm}$

$e_2 = (+1, -1, -1, +1, -1) \in {}_5C_2^{\pm}$

$e_3 = (+1, +1, -1, +1, -1) \in {}_5C_3^{\pm}$

$e_4 = (+1, +1, +1, +1, -1) \in {}_5C_4^{\pm}$

Velja:

$$x = \sum_{j=1}^{4} e_j$$

Pretvorba v binarni vektorski prostor:

$e_1 \in \{(+1, -1, -1, -1, -1) ... \underline{(-1, -1, -1, +1, -1)} ..., (-1, -1, -1, -1, +1)\}$

$e_2 \in \{(+1, +1, -1, -1, -1) ... \underline{(+1, -1, -1, +1, -1)} ..., (-1, -1, -1, +1, +1)\}$

$e_3 \in \{ (+1, +1, +1, -1, -1), \underline{(+1, +1, -1, +1, -1)} ..., (-1, -1, +1, +1, +1)\}$

$e_4 \in \{\underline{(+1, +1, +1, +1, -1)} ... (-1, +1, +1, +1, +1)\}$

Binarni vektor ima vrednost 1 tam, kjer se nahaja ustrezna permutacija v množici vseh permutacij z enako vsoto elementov. Ta množica je pravzaprav urejena množica vseh točk prostora ${}_5C_k^{\pm}$. Tako dobimo sledeče vektorje:

b_{e_1} = (00010)

b_{e_2} = (0001000000)

b_{e_3} = (0100000000)

b_{e_4} = (10000)

Te sestavimo v b = (0001000010000000100000000010000)

2.5. Primer odločanja z metodo podobnosti v eksponentnem prostoru

Vzemimo 10 učnih primerov, 5 za vsak razred.

$v_1 = (103, 55, 27, 304, 3, C_1) \rightarrow x_1 = (2, 0, -2, 4, -4)$

$v_2 = (213, 167, 26, 55, 214, C_2) \rightarrow x_2 = (2, 0, -4, -2, 4)$

$v_3 = (13, 17, 2, 51, 1, C_1) \rightarrow x_3 = (0, 2, -2, 4, -4)$

$v_4 = (372, 327, 62, 50, 542, C_1) \rightarrow x_4 = (2, 0, -2, -4, 4)$

$v_5 = (211, 123, 16, 153, 333, C_2) \rightarrow x_5 = (2, -2, -4, 0, 4)$

$v_6 = (311, 321, 62, 77, 8354, C_2) \rightarrow x_6 = (0, 2, -4, -2, 4)$

$v_7 = (5, 3, 2, 4, 1, C_1) \rightarrow x_7 = (4, 0, -2, 2, -4)$

$v_8 = (5, 23, 83, 4, 93, C_2) \rightarrow x_8 = (-2, 0, 2, -4, 4)$

$v_9 = (28, 21, 19, 16, 25, C_1) \rightarrow x_9 = (4, 0, -2, -4, 2)$

$v_{10} = (196, 83, 214, 58, 29, C_2) \rightarrow x_{10} = (2, 0, 4, -2, -4)$

Za vsak primer poiščemo pripadajoči binarni vektor razpršene predstavitve. Tako dobimo vektorje $b_1 - b_{10}$, ki jih razporedimo po razredih, v katere spadajo učni primeri.

w_{C1} in w_{C2} sta vektorja uteži za vsakega od razredov, ki ju dobimo tako, da element vektorja w postavimo na 1, če ima vsaj en vektor učnih primerov za dani razred na tem mestu vrednost 1, drugače mu dodelimo vrednost 0.

Binarni vektorji in s pomočjo njih dobljeni vektorji uteži so sledeči:

C_1	C_2
$b_1 = (00010000100000001000000010000)$	$b_2 = (00001000000100000010000000100)$
$b_3 = (00010000100000001000000010000)$	$b_5 = (00001000000100000000010000100)$
$b_4 = (00001000000100000010000001000)$	$b_6 = (00001000000010000010000000100)$
$b_7 = (10000000100000001000000010000)$	$b_8 = (00001000000001000000100001000)$
$b_9 = (10000000000100000010000001000)$	$b_{10} = (00100010000000010000000010000)$
$w_{C1} = (10011000110100001001000011000)$	$w_{C2} = (00101010000111010001011001100)$

Vzemimo zdaj testni primer t, za katerega želimo napovedati razred, in poiščimo njegov binarni vektor b_t.

$t = (5, 3, 4, 1, 2) \rightarrow x_t = (4, 0, 2, -4, -2)$

b_t = (10000010000000010000000001000)

S pomočjo tega vektorja izračunamo podobnost z obema razredoma:

$$\text{score}(t, C_j) = \sum_i b_{t_i} w_{ij}$$

Večja kot je vrednost, bolj je primer podoben učnim primerom danega razreda. V našem primeru imamo:

score(t,C_1) = 2

score(t,C_2) = 3

Metoda se torej odloči, da primer t klasificira v razred C_2.

3. Implementacija

Uspešnost metode je odvisna od implementacije. Ne gre samo za pravilno, temveč tudi za čim bolj optimalno delovanje metode glede porabe pomnilnika in časa izvajanja. Zastavili smo si, da bo naloga izvedena v okolju R. Ko se prvič srečaš z nekim programskih jezikom, potrebuješ nekaj časa, da se privadiš njegovim posebnostim in začneš izkoriščati prednosti ter se izogibati slabostim. Da bi okolje dobro obvladal, potrebuješ še mnogo izkušenj. Težko je torej pričakovati, da bo ta naloga izpeljana povsem optimalno, vseeno pa je bilo vloženega precej truda in časa v razmislek, kako iz omejenega znanja potegniti največ.

3.1. Implementacija osnovne ideje

Implementacija algoritma po teoretični predlogi sicer vodi do pravilnega delovanja, vendar pa uporabljeni prijemi presegajo tehnološke meje, ki jih imamo na voljo. To se je kmalu pokazalo tudi za našo implementacijo metode porazdeljene predstavitve, opisane v poglavjih 2.4. in 2.5.

Pri izbiri baz podatkov za teste smo želeli najprej izbrati razumljivo množico, ki ima vse atribute v istem obsegu vrednosti. Baza Hill-Valley [3], ki ustreza tej zahtevi, ima kar 100 atributov. Naivna implementacija klasifikacijske metode, ki dobesedno sledi postopku, opisanem v prejšnjem poglavju , ni uporabna za primere z veliko atributi. Veliko pomeni že vse, kar je več kot 20, predhodne študije [5] pa so objavile rezultate tudi na dimenzijah velikosti 48.

3.2. Ključne težave implementacije

Osnovna implementacija algoritma je imela dve veliki težavi, ki sta nastopili pri povečevanju števila atributov. Poraba pomnilnika in časovna zahtevnost sta prehitro narasli čez sprejemljivo mejo.

Prva težava se je nahajala na mestu, kjer je bilo treba ustvariti vse permutacije, da bi lahko razpršeno predstavitev pravilno pretvorili v binarni vektor. Preprost izračun je odkril sledeče:

– vseh permutacij dveh števil za dimenzijo 24 je 2^{24},

– vsaka permutacija ima 24 števil,

– vsako število zasede 64 bitov,

– 2^{24} x 24 x 8 B = 3,2 GB.

To pomeni, da 6 GB pomnilnika ne zadostuje niti za hranjenje permutacij dolžin, večjih od 24.

Druga večja zahteva po pomnilniku je vektor uteži oziroma binarni vektor opisa razreda, ki zahteva 2^N*8 B pomnilnika za vsak razred. To pomeni, da za 24 atributov in problem z dvema razredoma klasifikacijski model zasede približno 268 MB. Od tod dobimo, da 16 GB pomnilnika ni dovolj niti za hranjenje opisov dveh razredov pri dimenziji N = 30.

Za orientacijo smo izmerili še časovno zahtevnost. Pri 24 atributih je računanje vseh permutacij trajalo skoraj 10 minut. Razpršeno kodiranje enega samega učnega primera pa 77139 sekund, kar je približno 21 ur in pol. Za okoli 600 učnih primerov, kolikor jih ima ena od testnih baz, bi učenje potekalo dobro leto in pol!

Optimizacija implementacije ni prinesla omembe vrednega izboljšanja, in postalo je jasno, da je treba spremeniti pristop.

Prečesali smo program in iskali najbolj časovno potratne dele. Prednjačilo je iskanje trenutne permutacije v prej narejenem seznamu vseh permutacij.

Problem smo rešili v dveh korakih.

3.3. Permutacije

Izpustili smo kreiranje seznama vseh permutacij, saj algoritem, prikazan v kodi 1, za poljubno binarno permutacijo poišče zaporedno številko kombinacije z enakim številom enic.

```
ie<-0
idx<-1
for (i in 1:N){
      if (b[i]==1){
            ie<-ie+1
            if (i>1) idx<-idx+C[i-1,ie]
      }
}
```

Koda 1. Izračun zaporedne številke kombinacije (idx).

Algoritem zapišemo z enačbo:

$$I = 1 + \sum_{i,B_i=1} C(i-1, N_i),$$

kjer je

B – binarni niz,

I – zaporedna številka oziroma indeks permutacije, ki jo predstavlja B,

i – indeks elementa v B (skrajno levi element ima indeks 1),

N_i – število elementov z vrednostjo 1, ki imajo indeks manjši ali enak i,

C(x,y) – število kombinacij y elementov na x mestih.

Primer:

B = 10101

$i,B_i=1 = \{1,3,5\}$

I = 1+ C(0,1) + C(2,2) + C(4,3) = 1 + 0 + 1 + 4 = 6

Če pogledamo vse možne kombinacije dolžine 5 s tremi enicami v vrstnem redu, kot v Tabeli 1, vidimo, da je dana permutacija B na 6. mestu.

C(5,3) = 10

I	**B**
1	11100
2	11010
3	10110
4	01110
5	11001
6	**10101**
7	01101
8	10011
9	01011
10	00111

Tabela 1. Seznam vseh permutacij niza 11100 in njenih indeksov.

S tem smo lahko izpustili kreiranje permutacij in močno zmanjšali porabo pomnilnika. Hkrati tudi ni bilo več potrebe po iskanju permutacij v seznamu, saj smo zaporedno številko izračunali, in tako pravi bit postavili hitreje. Za N = 24 se čas učenja enega primera zmanjša iz 21,5 ure na 1,5 sekunde.

3.4. Klasifikacijski model

Opis razreda, ki je bil predstavljen z binarnim nizom, smo nadomestili z razpršeno tabelo, v kateri se hranijo, kot je vidno v kodi 2, števila, izračunana z algoritmom iz prvega koraka, oziroma enice iz binarnega niza. Poraba pomnilnika za hranjenje klasifikacijskega modela, ki vsebuje opise razredov s pomočjo razpršenih tabel, se zmanjša v najslabšem primeru na (N-1)*N_L*8B, kjer je N število atributov in N_L število učnih primerov. Dejanska poraba je mnogo manjša, saj opisi primerov niso popolnoma različni, ampak imajo običajno mnogo podobnosti.

S pomočjo takšne predstavitve je iskanje po opisu precej hitrejše in se s tem klasifikacija vsakega primera pospeši za približno faktor 1000.

```
hashIt <- function(H,r){
      for(i in 1:length(r)){
            if (!is.null(H[i][[1]])){
                  if (!(r[[i]] %in% H[[i]])) H[[i]] <- c(H[[i]],r[[i]])
            }
            else  H[[i]]<-r[[i]]
      }
      return(H)
}
```

Koda 2. Razpršitev z razpršeno tabelo; r je vektor indeksov permutacij.

Klasifikacijski model je zdaj zasnovan kot skupek razpršenih tabel, kjer vsaka predstavlja en vektor uteži, opisuje en razred in ima N-1 seznamov, kjer je N število atributov, s katerimi so opisani primeri. Vsak element seznama predstavlja indeks N-bitne binarne permutacije, pri čemur ima vsak seznam indekse permutacij porazdeljene predstavitve z enakim številom enic. Vsak seznam ima vedno vsaj en element.

Pri tej implementaciji ni nujno res, da število učnih primerov ne vpliva na hitrost odločanja. Če imamo 10 učnih primerov, bodo seznami lahko precej krajši, kot če imamo 100000 učnih primerov. Seveda je to odvisno od tega, kako oddaljeni so med seboj primeri istega razreda. Teoretično je možno, da je ob majhni varianci znotraj razreda čas iskanja enak. Ne glede na to pa čas, razen v posebnih primerih, kjer so si primeri med seboj popolnoma različni, ne narašča linearno.

razred 1	razred 2
[1] 1, 8, 7, 5, 3	[1] 1, 8, 6, 7
[2] 16, 28, 20, 22, 2, 7, 36, 29, 33, 18	[2] 16, 27, 23, 11, 28, 13, 21
[3] 21, 55, 27, 76, 51, 22, 63, 84, 78,72,82,29,6	[3] 51, 83, 50, 52, 46, 56, 13, 12, 14, 34
[4] 91, 125, 97, 66, 57, 112, 126, 21, 62, 121,99,76	[4] 56, 126, 55, 60, 66, 67, 48, 6, 12, 8, 69
[5] 112, 126, 118, 50, 42, 77, 44, 122, 120, 97	[5] 47, 124, 36, 40, 50, 49, 30, 27, 18, 2, 56
[6] 75, 84, 81, 26, 70, 17, 83	[6] 23, 76, 13, 18, 21, 25, 9, 19, 4, 28, 3
[7] 31, 36, 34, 7, 32, 3	[7] 5, 32, 2, 3, 4, 12, 8, 1
[8] 7, 9, 8, 1, 4	[8] 6, 8, 3, 1, 2

Primer prenovljenega klasifikacijskega modela

S tem smo algoritem pripravili na testiranje, saj je baza Hill-Valley postala obvladljiva celo s 100 atributi.

Preizkusili smo tudi dve modifikaciji opisanega modela. Pri prvi smo izločili vse podvojitve indeksov, tako da so v klasifikacijskem modelu ostali v vsakem seznamu le indeksi, ki so se pojavili v primerih natanko enega razreda. S tem smo želeli za klasifikacijo uporabiti le nedvoumne indekse. Žal se je izkazalo, da je na ta način ostalo zelo malo indeksov in je bila klasifikacija slabša. Idejo smo opustil, čeprav bi morda v hibridni verziji lahko služila kot dodatna utež.

Pri drugi modifikaciji gre za dodeljevanje uteži vsakemu elementu v tabeli. Več je učnih primerov z neko permutacijo, tem pomembnejši je element, zato take bolj utežimo. Uporabili smo linearne uteži, tako da smo utež povečali za 1 vsakič, ko se je v učnem primeru pojavil indeks pripadajoče permutacije. Ideja je, da preprečimo, da nek indeks, ki ga je enemu razredu prispeval en sam primer, vpliva na klasifikacijo enako kot taisti indeks, ki ga je drugemu razredu prispevalo 500 primerov. Če je število učnih primerov za oba razreda enako, potem je ta ideja dovolj. Ker pa ima lahko vsak razred zelo različno število učnih primerov, smo uteži vsakič namesto za 1 povečali za $1/N_{Lc}$, kjer je N_{Lc} število učnih primerov razreda, v katerega spada trenutni primer. Na ta način zavarujemo manjšinske razrede.

Poglejmo preprost primer:

Imamo 5 učnih primerov razreda R1 in 2 učna primera razreda R2, ki pri učenju prispevajo naslednje indekse permutacij:

R_1	R_2
b_1 = [1, 2, 8, 4]	b_6 = [1, 3, 5, 1]
b_2 = [1, 10, 6, 1]	b_7 = [5, 9, 5, 3]
b_3 = [1, 10, 8, 4]	
b_4 = [3, 3, 4, 2]	
b_5 = [2, 7, 10, 2]	

Po učenju je klasifikacijski model videti takole:

R_1	R_2
[1] 1, 3, 2	[1] 1, 5
[2] 2, 10, 3, 7	[2] 3, 9
[3] 8, 6, 4, 10	[3] 5
[4] 4, 1, 2	[4] 1, 3

Klasificirati želimo testni primer b_t = [2, 3, 5, 4]

score(t,R_1) = 3

score(t,R_2) = 2

Osnovna metoda se odloči, da primer klasificira v razred R_1.

Verzija z utežmi sproti ustvarja še tabelo uteži, ki je po učenju videti takole:

R_1	R_2
[1] 3, 1, 1	[1] 1, 1
[2] 1, 2, 1, 1	[2] 1, 1
[3] 2, 1, 1, 1	[3] 2
[4] 2, 1, 2	[4] 1, 1

scoreW(t,R_1) = 1 + 1 + 2 = 4

scoreW(t,R_2) = 1 + 2 = 3

Tudi na ta način bi se metoda odločila za razred R_1, kljub temu da sta oba učna primera v R_2 prav tako imela tretji indeks 5. Zato popravimo scoreW tako, da ga delimo s številom učnih primerov tistega razreda.

scoreWr(t,R_1) = scoreW(t,R_1) / N_{LR1} = 4/5 = 0,8

scoreWr(t,R_2) = scoreW(t,R_2) / N_{LR2} = 3/2 = 1,5

Zdaj vsak zadetek v razredu R_2 nosi večjo težo kot zadetek v R_1, in to se pozna na rezultatu klasifikacije, ki se tokrat odloči za razred R_2.

Ta modifikacija je dala mešane rezultate, ki so bili dovolj zanimivi, da smo jo uporabili pri testiranju. Nahaja se pod oznako **DR-W**, osnovna metoda pa je označena z **DR**.

4. Testiranje

Testiranje je potekalo v več fazah.

Merili smo pravilnost odločanja metode na več množicah podatkov in opazovali vpliv števila učnih primerov.

Merili smo tudi porabo pomnilnika v odvisnosti od velikosti učne množice in števila atributov.

Zanimala nas je tudi hitrost učenja in hitrost klasifikacije v odvisnosti od velikosti učne množice in števila atributov.

4.1. Množice podatkov

V nadaljevanju naštete množice podatkov smo uporabljali na dva načina. Če je bil problem predhodno razdeljen na učno in testno množico, smo vsa testiranja opravili na istih dveh razbitjih, sicer smo podatke naključno razdelili na učno in testno množico. Množice, ki imajo manj kot 1000 primerov, smo testirali na dva načina:

– z 2-kratnim prečnim preverjanjem,

– z metodo izpusti enega (**LOO**, leave one out cross-validation).

BIRDS character pattern database (**BIRDS)** [1]

Primeri opisujejo skenirane ročno pisane numerične simbole. Vsak primer je slika enega simbola. Vrednosti atributov dobimo tako, da skeniran simbol razdelimo na H*W sektorjev in preštejemo število pik v vsakem sektorju. Za potrebe testiranja sem bazo pripravil v petih različnih dimenzijah: 3*5, 4*6, 5*7, 6*8 in 7*9.

Število primerov: 202958.

Število atributov: 15, 24, 35, 48, 63 + razred.

Breast Cancer Wisconsin [3]

Vsebuje tri ločene množice:

Wisconsin Breast Cancer Database (BCW) [8]

Napovedovanje malignega ali benignega raka na dojki.

Število primerov: 683.

Število atributov: 9 + razred.

Wisconsin Diagnostic Breast Cancer (WDBC)

Diagnoza malignega ali benignega raka na dojki.

Število primerov: 569.

Število atributov: 30 + razred.

Wisconsin Prognostic Breast Cancer (WPBC)

Prognoza ponovne pojavitve raka na dojki.

Število primerov: 194.

Število atributov: 32 + razred.

Protein localization sites (**E.coli)** [3]

Točka lokalizacije proteinov v celici.

Število primerov: 336.

Število atributov: 7 + razred.

Glass identification database (Glass) [3]

Študija klasifikacije vrste stekla je nastala pri kriminološki preiskavi. Na kraju zločina najdeno steklo je lahko uporabljeno kot dokaz, če je pravilno identificirano.

Število primerov: 214.

Število atributov: 9 + razred.

Hill-Valley dataset (Hill-Valley) [3]

Vsak zapis predstavlja 100 točk v dvodimenzionalnem grafu. Ko so točke izrisane po vrsti (od 1 do 100) kot Y koordinate, dobimo izrisan hrib (izboklino v terenu) ali dolino (vboklino v terenu). Uporabljena je bila različica s šumnimi podatki.

Število primerov: učna množica 606 + testna množica 606.

Število atributov: 100 + razred.

Image segmentation data (Segmentation) [3]

Definicija majhnega segmenta slike. Primeri so bili izbrani naključno iz baze sedmih slik zunanjosti. Slike so bile ročno označene tako, da je vsak piksel pripadal nekemu razredu. Primeri so velikosti 3*3.

Število primerov: učna množica 210 + testna množica 2100.

Število atributov: 19 + razred.

Iris plants database (Iris) [3]

Podatki vsebujejo 3 razrede s po 50 primeri vsakega razreda, kjer vsak razred predstavlja eno vrsto perunike. En razred je linearno ločljiv od drugih dveh; slednja nista linearno ločljiva.

Število primerov: 150.

Število atributov: 4 + razred.

Poker hand dataset (Poker) [3]

Vsak zapis je primer roke, ki vsebuje 5 igralnih kart izmed 52. Vsaka karta je opisana z dvema atributoma (barvo in številko) za skupno 10 atributov. Razred opisuje vrsto "poker roke". Vrstni red kart je pomemben, zato je možnih 480 kraljevih lestvic namesto štirih.

Število primerov: učna množica 25010 + testna množica 1000000.

Število atributov: 10 + razred.

4.2. Ostale metode

Za primerjavo uspešnosti metode smo pod enakimi pogoji testirali tudi nekatere druge metode, ki so vključene v knjižnici CORElearn. S primerjavo rezultatov je lažje oceniti, kje ležijo prednosti in slabosti posamezne metode.

Naivni Bayesov klasifikator (BAYES) [6, pogl. 7]

Naivni Bayesov klasifikator predpostavlja pogojno neodvisnost vrednosti različnih atributov pri danem razredu. Učni algoritem s pomočjo učne množice podatkov aproksimira pogojne verjetnosti. Znanje klasifikatorja je sestavljeno iz tabele aproksimacij apriornih verjetnosti razredov, ki jih dobimo z Laplacevim približkom, ter tabele pogojnih verjetnosti razredov pri danih vrednostih posameznih atributov, ki jih dobimo z m-oceno.

Odločitvena drevesa (TREE) [6, pogl. 7]

Odločitveno drevo je poseben primer množice odločitvenih pravil. Sestavljeno je iz notranjih vozlišč, ki ustrezajo atributom, vej, ki ustrezajo podmnožicam vrednosti atributov, in listov, ki ustrezajo razredom. Vsaka izmed poti od korena do lista ustreza odločitvenemu pravilu, pogoji pa so med seboj konjunktivno povezani. Tako lahko vsako drevo izrazimo z množico odločitvenih pravil. V bolj posplošeni obliki vsebujejo odločitvena drevesa v notranjih vozliščih funkcije več atributov.

Naključni gozdovi (RF) [6, str. 80]

Generiramo 100 ali več odločitvenih dreves, tako da se v vsakem vozlišču naključno izbere log(N)+1 atributov, kjer je N število vseh atributov, ki postanejo kandidati za najboljši atribut. Učni primeri za posamezna drevesa se izbirajo z naključno izbiro z vračanjem. Za klasifikacijo novega primera vsako drevo prispeva svoj glas razredu, v katerega bi primer klasificiralo. Iz vseh glasov dobimo verjetnostno porazdelitev po razredih.

Naključni gozdovi z uteženim glasovanjem (RF-NEAR) [7]

V metodi RF niso vsa drevesa enako kriva za napačno klasifikacijo posameznih primerov. Ta metoda za vsak testni primer poišče najbolj podobne primere. Ker se podobnost meri znotraj dreves, je treba hraniti še informacije o podobnosti za vse učne primere. Pri klasifikaciji novega primera se vsem primerom v listu, v katerega drevo klasificira primer, mera podobnosti poveča za ena. Iz vseh dreves izberemo nekaj najbolj podobnih primerov in jih klasificiramo z drevesi, pri katerih ti primeri niso bili uporabljeni za učenje. Nato na vseh drevesih v gozdu izmerimo odstopanje glasovanja za pravi razred od glasovanja za najverjetnejši razred. Drevesa, kjer je to odstopanje negativno, izpustimo iz končne klasifikacije. Uporabimo uteženo glasovanje, kjer so uteži povprečna odstopanja na podobnih primerih, ko ti niso bili med učnimi primeri.

K-najbližjih sosedov (knn) [6, pogl. 8]

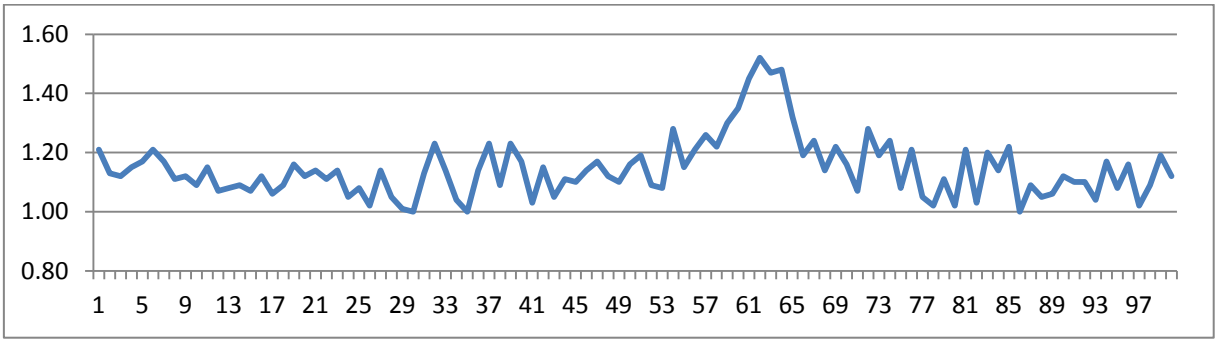
Algoritem za dani novi primer poišče v množici učnih primerov K najbolj podobnih primerov in oceni verjetnostno porazdelitev iz porazdelitve teh K primerov po razredih. Pri tej metodi gre za tako imenovano leno učenje, saj se množica učnih primerov le shrani. Glavnina procesiranja je pri klasifikaciji, in tako je njena časovna zahtevnost večja kot pri drugih metodah učenja.

Z razdaljo utežen klasifikator k-najbližjih sosedov (knnKERN) [6, pogl. 8]

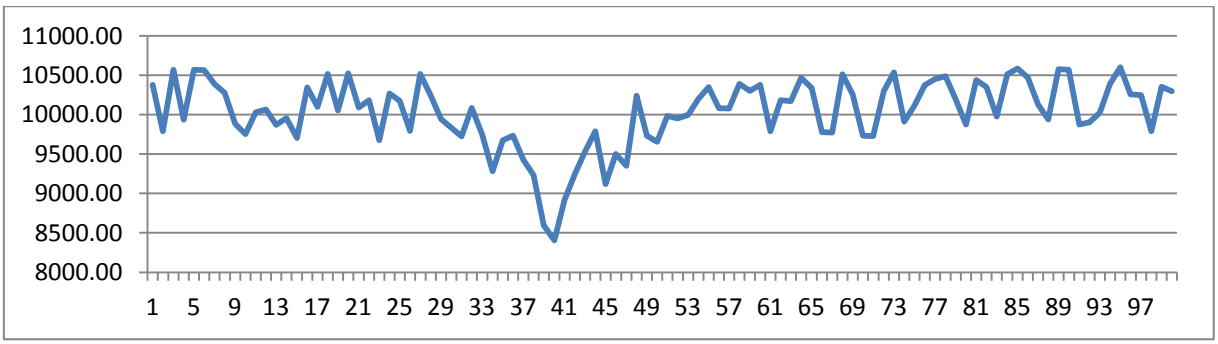
Gre za modificiran algoritem k-najbližjih sosedov. Razlika je v tem, da je vpliv vsakega učnega primera na klasifikacijo novega primera utežen z razdaljo po Gaussovi funkciji. Z uporabo uteži lahko na klasifikacijo vplivajo vsi učni primeri.

4.3. Klasifikacijska točnost

Hill-Valley dataset



Slika 1. Vizualizacija primera iz množice Hill-Valey, ki predstavlja hrib.



Slika 2. Vizualizacija primera iz množice Hill-Valey, ki predstavlja dolino.

Rezultati, ki jih je dala razpršena metoda, so odlični, če jih primerjamo z ostalimi metodami, ki, kot je vidno v Tabeli 2, pri tem problemu popolnoma odpovedo. Vrednosti atributov niso ključne za klasifikacijo, saj se, kot vidimo na Slikah 1 in 2, vrednosti lahko zelo razlikujejo, toda pri nekaterih zaporednih atributih te vrednosti odstopajo od ostalih atributov navzgor oziroma navzdol, torej gre za globalni maksimum oziroma minimum. Ker metoda DR upošteva vrstni red vrednosti atributov, bo ugotovila, kje se nahajata.

	Valley			Hill			Skupno		
	Sr	Nr	A	Sr	Nr	A	S	N	A
DR	295	299	0,987	301	307	0,980	596	606	0,983
DR-W	274	299	0,916	288	307	0,938	562	606	0,927
BAYES	149	299	0,498	164	307	0,534	313	606	0,517
RF	171	299	0,572	161	307	0,524	332	606	0,548
RF-NEAR	177	299	0,592	162	307	0,528	339	606	0,559
TREE	182	299	0,609	133	307	0,433	315	606	0,520
knn	244	299	0,816	58	307	0,189	302	606	0,498
knnKERN	239	299	0,799	60	307	0,195	299	606	0,493

Tabela 2. Klasifikacijska točnost (A) za problem Hill-Valey; Nr je število primerov razreda, Sr je število pravilno klasificiranih primerov za razred.

Razložili smo, od kod dobri rezultati, vendar pa je precej odvisno od kvalitete in raznolikosti učnih primerov. Na primer, če vzamemo samo prvih 100 učnih primerov namesto vseh 606, klasifikacijska točnost pade na 0,7. Za DR je pri tem problemu pomembno učenje čim več različnih primerov, takih, ki imajo globalni ekstrem na čim bolj različnih mestih v zaporedju atributov, da se pokrijejo vse možnosti. Ni vsa klasifikacija odvisna od ene točke, vendar pa veliko pomeni dejstvo, da lahko garantiramo zadetek v prvem ali zadnjem seznamu permutacij, ki tu predstavljata globalna ekstrema (indeks atributa z največjo oziroma najmanjšo vrednostjo), in se izognemo situaciji, ki denimo nastopi pri testu z le 100 učnimi primeri, ko ostanemo pri nekaterih testnih primerih brez zadetkov. To pomeni, da taki testni primeri nimajo nikakršne podobnosti s katerim koli od učnih primerov, in se mora metoda pri klasifikaciji odločati med dvema ničlama, kar pomeni, da je za take testne primere uspeh klasifikacije naključen.

Odločitev za posamezen razred ni vedno prepričljiva. Najbolj dominantna odločitev je npr. rezultat 12 : 2. Če upoštevamo, da gre za 100 atributov in da je število možnih permutacij ogromno ter da se je klasifikacijski model za vsak razred polnil le s 300 učnimi primeri, potem 12 zadetkov od 100 pomeni kar veliko. Več pa je takih primerov, kjer je podobnost le za 1 ali 2 v korist enega razreda.

Verzija z utežmi je tu slabša. To si razlagamo tako, da so, z izjemo globalnega ekstrema, ostali deli grafa podobni pri obeh razredih in imajo večjo možnost pojavitve kot ekstremi in zato večjo utež.

Sprva nismo razumeli globljega pomena rezultatov za ta problem in smo se z velikim optimizmom lotili novih problemov. Slika se je hitro spremenila.

Wisconsin Prognostic Breast Cancer

Trije problemi znotraj množice Breast Cancer Wisconsin so hkrati podobni in tudi precej različni. Pri prvem problemu se DR izkaže solidno, a, tako kot vse metode, šepa pri klasifikaciji razreda R. Slednje popravi DR-W, toda na račun razreda N. Ker pa je slednjih

primerov precej več, je skupen rezultat slabši. Zanimivo je, da metoda knn napove vse primere kot razred N z izjemo enega, in še tistega napačno.

	N			R			ALL		
	Sr	Nr	A	Sr	Nr	A	S	N	A
DR	126	148	0.848	12	46	0.250	138	194	0.711
DR-W	75	148	0.507	34	46	0.739	109	194	0.562
BAYES	115	148	0.777	16	46	0.348	131	194	0.675
RF	145	148	0.980	6	46	0.130	151	194	0.778
RF-NEAR	139	148	0.939	9	46	0.196	148	194	0.763
TREE	125	148	0.845	12	46	0.261	137	194	0.706
knn	147	148	0.993	0	46	0.000	147	194	0.758
knnKERN	116	148	0.784	12	46	0.261	128	194	0.660

Tabela 3. Klasifikacijska točnost (A) za problem WPBC; Nr je število primerov razreda, Sr je število pravilno klasificiranih primerov za razred.

Pri LOO le metodi RF, ki sta že bili najboljši, opazno izboljšata rezultat. DR izgubi na klasifikacijski točnosti razreda N in se s tem odreže slabše tudi od metode TREE.

	N			R			SKUPAJ		
	Sr	Nr	A	Sr	Nr	A	S	N	A
DR	114	148	0,770	14	46	0,304	128	194	0,657
DR-W	75	148	0,507	34	46	0,739	109	194	0,562
BAYES	104	148	0,703	19	46	0,413	123	194	0,634
RF	147	148	0,993	9	46	0,196	156	194	0,804
RF-NEAR	146	148	0,986	11	46	0,239	157	194	0,809
TREE	123	148	0,831	17	46	0,370	140	194	0,722
knn	147	148	0,993	2	46	0,043	149	194	0,768
knnKERN	117	148	0,791	11	46	0,239	128	194	0,660

Tabela 4. Klasifikacijska točnost (A) za problem WPBC z LOO; Nr je število primerov razreda, Sr je število pravilno klasificiranih primerov za razred.

Wisconsin Breast Cancer Database

	2			4			ALL		
	Sr	Nr	A	Sr	Nr	A	S	N	A
DR	391	444	0.880	203	239	0.847	594	683	0.870
DR-W	422	444	0.950	208	239	0.870	630	683	0.922
BAYES	427	444	0.962	236	239	0.987	663	683	0.971
RF	432	444	0.973	228	239	0.954	660	683	0.966
RF-NEAR	432	444	0.973	229	239	0.958	661	683	0.968
TREE	426	444	0.959	221	239	0.925	647	683	0.947
knn	436	444	0.982	207	239	0.866	643	683	0.941
knnKERN	434	444	0.977	216	239	0.904	650	683	0.952

Tabela 5. Klasifikacijska točnost (A) za problem BCW; Nr je število primerov razreda, Sr je število pravilno klasificiranih primerov za razred.

Drugi problem se izkaže za najlažjega, saj vse metode dobro delujejo, le DR je slabši. Utežena verzija se izkaže za boljšo, a je še vedno za dva odstotka slabša od preostalih metod. Najboljši rezultat je dosegla metoda BAYES.

Pri LOO vse metode malenkost izboljšajo svoj rezultat, le BAYES in DR-W ostajata enaka. Izjema je tokrat DR, ki ponovno izrazito poslabša klasifikacijsko točnost. Razlog je bržkone v nasičenosti, saj je pri dimenziji 9 prostora precej malo in v tem primeru povečanje števila učnih primerov briše meje med razredoma, saj se pojavljajo primeri iz drugih razredov preblizu. Glede na to, da sta ostali množici večje dimenzije, učinek pa je enak, to vsekakor ni nujno. Velja omeniti, da je metoda DR-W precej odporna na prenasičenje, tako da večanje števila primerov ne vpliva na njeno učinkovitost.

	2			4			SKUPAJ		
	Sr	Nr	A	Sr	Nr	A	S	N	A
DR	362	444	0,815	194	239	0,810	556	683	0,814
DR-W	422	444	0,950	208	239	0,870	630	683	0,922
BAYES	427	444	0,962	236	239	0,987	663	683	0,971
RF	432	444	0,973	232	239	0,971	664	683	0,972
RF-NEAR	432	444	0,973	232	239	0,971	664	683	0,972
TREE	430	444	0,968	224	239	0,937	654	683	0,958
knn	435	444	0,980	215	239	0,900	650	683	0,952
knnKERN	434	444	0,977	222	239	0,929	656	683	0,960

Tabela 6. Klasifikacijska točnost (A) za problem BCW z LOO; Nr je število primerov razreda, Sr je število pravilno klasificiranih primerov za razred.

Wisconsin Diagnostic Breast Cancer

Ta primer je po rezultatih podoben BCW, le da je uspešnost nekoliko nižja. Razlika je v tem, da DR-W ne prinese bistvene izboljšave.

	M			B			SKUPAJ		
	Sr	Nr	A	Sr	Nr	A	S	N	A
DR	158	212	0,743	321	357	0,899	479	569	0,842
DR-W	202	212	0,953	284	357	0,794	486	569	0,854
BAYES	194	212	0,915	337	357	0,944	531	569	0,933
RF	196	212	0,925	346	357	0,969	542	569	0,953
RF-NEAR	195	212	0,920	347	357	0,972	542	569	0,953
TREE	188	212	0,887	337	357	0,944	525	569	0,923
knn	182	212	0,858	353	357	0,989	535	569	0,940
knnKERN	193	212	0,910	343	357	0,961	536	569	0,942

Tabela 7. Klasifikacijska točnost (A) za problem WDBC; Nr je število primerov razreda, Sr je število pravilno klasificiranih primerov za razred.

	M			B			SKUPAJ		
	Sr	Nr	A	Sr	Nr	A	S	N	A
DR	134	212	0,632	302	357	0,845	436	569	0,766
DR-W	202	212	0,953	283	357	0,793	485	569	0,852
BAYES	196	212	0,925	337	357	0,944	533	569	0,937
RF	199	212	0,939	349	357	0,978	548	569	0,963
RF-NEAR	199	212	0,939	349	357	0,978	548	569	0,963
TREE	190	212	0,896	345	357	0,966	535	569	0,940
knn	186	212	0,877	350	357	0,980	536	569	0,942
knnKERN	195	212	0,920	346	357	0,969	541	569	0,951

Tabela 8. Klasifikacijska točnost (A) za problem WDBC z LOO; Nr je število primerov razreda, Sr je število pravilno klasificiranih primerov za razred.

Iris plants database

Iris izpostavi verjetno največjo slabost metode DR. Ta se dobro vidi, če pogledamo nekaj učnih primerov za vsakega od treh razredov.

Iris-setosa				Iris-versicolor				Iris-virginica			
4,8	3,4	1,6	0,2	7	3,2	4,7	1,4	6,3	3,3	6	2,5
4,8	3	1,4	0,1	6,4	3,2	4,5	1,5	5,8	2,7	5,1	1,9
4,3	3	1,1	0,1	6,9	3,1	4,9	1,5	7,1	3	5,9	2,1
5,8	4	1,2	0,2	5,5	2,3	4	1,3	6,3	2,9	5,6	1,8
5,7	4,4	1,5	0,4	6,5	2,8	4,6	1,5	6,5	3	5,8	2,2
1	2	3	4	1	3	2	4	1	3	2	4

Tabela 9. Vrstni red vrednosti atributov za primere Iris.

Če natančno pogledamo vrednosti atributov za posamezne primere, vidimo, da so po velikosti vedno v istem vrstnem redu. To ne velja le za teh 5 izbranih primerov, temveč za vseh 50 primerov vsakega od razredov. To pomeni, da so z zornega kota metode DR primeri za posamezni razred identični. To dejstvo je lepo vidno v klasifikacijskem modelu.

Iris-setosa	Iris-verisicolor	Iris-virginica
[1] 1	[1] 1	[1] 1
[2] 1	[2] 2	[2] 2
[3] 1	[3] 1	[3] 1

Klasifikacijski model metode DR za problem Iris.

Vidimo, da ima vsak seznam natanko eno vrednost, kar je tudi minimum, če obstaja vsaj en učni primer za razred, torej so primeri znotraj vsakega razreda identični. Samo po sebi to dejstvo ni slabost, saj bi to garantiralo stoodstotno klasifikacijsko točnost metode. Problem je, da vrstni red ni unikaten za vsak razred. Vrstni red je za razred versicolor identičen razredu virginica, kar pomeni, da sta za metodo DR oba razreda enakovredna, in tako primer za enega

od teh dveh razredov klasificira s petdesetodstotno točnostjo. Razred setosa je unikaten, tako da DR ta razred vedno klasificira pravilno. Oboje je vidno v Tabelah 10 in 11.

	Iris-setosa			Iris-versicolor			Iris-virginica			ALL		
DR	50	50	1,000	26	50	0,520	26	50	0,520	102	150	0,680
BAYES	47	50	0,940	37	50	0,740	41	50	0,820	125	150	0,833
RF	50	50	1,000	47	50	0,940	47	50	0,940	144	150	0,960
RF-NEAR	50	50	1,000	47	50	0,930	45	50	0,900	142	150	0,943
TREE	50	50	1,000	47	50	0,940	45	50	0,900	142	150	0,947
knn	50	50	1,000	45	50	0,900	44	50	0,880	139	150	0,927
knnKERN	50	50	1,000	47	50	0,940	46	50	0,910	143	150	0,950

Tabela 10. Klasifikacijska točnost za problem Iris.

	Iris-setosa			Iris-versicolor			Iris-virginica			ALL		
DR	50	50	1,000	26	50	0,520	26	50	0,520	102	150	0,680
BAYES	47	50	0,940	41	50	0,820	44	50	0,880	132	150	0,880
RF	50	50	1,000	47	50	0,940	46	50	0,920	143	150	0,953
RF-NEAR	50	50	1,000	47	50	0,940	46	50	0,920	143	150	0,953
TREE	50	50	1,000	47	50	0,940	48	50	0,960	145	150	0,967
knn	50	50	1,000	47	50	0,940	45	50	0,900	142	150	0,947
knnKERN	50	50	1,000	47	50	0,940	46	50	0,910	143	150	0,950

Tabela 11. Klasifikacijska točnost za problem Iris z LOO.

Pri tem problemu vidimo, kako pomemben je vrstni red vrednosti atributov za delovanje DR. Na to lahko vpliva tudi zaloga vrednosti posameznega atributa. Če je presek zaloge vrednosti nekega atributa z zalogami vrednosti ostalih atributov prazen, potem je ta atribut za delovanje metode DR irelevanten in ga lahko izpustimo. Če to velja za vse atribute, potem so vsi primeri identični in se metoda DR izrodi v met n-strane kocke, kjer je n število možnih razredov. Da bi se temu izognili, bi bilo treba pri takem problemu vse atribute predhodno skalirati na isti interval vrednosti.

Pri problemu Iris temu ni tako, saj se zaloge vrednosti delno prekrivajo. Celo znotraj primerov za posamezne razrede je nekaj malega prekrivanja, a slednje dejstvo pri primerih, ki so v tej množici, nikdar ne vpliva na vrstni red vrednosti atributov.

Zaradi tega uteži ne spremenijo ničesar in ima DR-W enako klasifikacijsko točnost. Ostale metode so med sabo primerljive, le BAYES zaostaja.

Protein localization sites (E.coli)

DR je ponovno najslabša metoda. Ker gre za problem s samo 7 atributi, smo se malce poglobili v množico podatkov, da bi morda našli razlog za slabšo klasifikacijo. Rezultat tega je nekaj dejstev:

– Med vsemi 336 primeri je kar 51 takih, ki so z zornega kota DR identični, pripadajo pa drugemu razredu, kar pomeni, da je za te primere klasifikacija petdesetodstotno natančna, če se en nahaja med učnimi primeri. Slednje je v različici LOO vedno res in od tod izvira tudi padec uspešnosti metode. Tako smo našli še eno možno razlago za slabšo klasifikacijo v prejšnjih problemih.

– DR je edini, ki pri razredu omL doseže stoodstotno točnost. Razlog je v tem, da so si primeri precej podobni in niti eden nima identičnega dvojnika v drugem razredu.

– Razred imL ima le dva primera, ki pa sta si precej različna med seboj in zato se metoda vseeno ne more odločiti pravilno.

– V nasprotju z imL ima imS oba svoja primera taka, da sta dvojnika primerov z različnega razreda.

	cp	im	pp	imU	om	omL	imL	imS	ALL
DR	0,885	0,649	0,654	0,457	0,825	1,000	0,000	0,000	0,731
DR-W	0,577	0,461	0,721	0,543	0,725	1,000	0,000	0,000	0,577
BAYES	0,916	0,571	0,808	0,486	0,750	0,000	0,000	0,000	0,741
RF	0,979	0,844	0,808	0,486	0,800	0,200	0,000	0,000	0,836
RF-NEAR	0,979	0,844	0,827	0,514	0,850	0,400	0,000	0,000	0,848
TREE	0,958	0,831	0,750	0,400	0,600	0,000	0,000	0,000	0,792
knn	0,979	0,779	0,788	0,200	0,250	0,200	0,000	0,000	0,756
knnKERN	0,979	0,779	0,808	0,314	0,300	0,400	0,000	0,000	0,777

Tabela 12. Klasifikacijska točnost za problem E.coli.

– E.coli je prvi problem, pri katerem je različica DR-W pridobila z LOO, in to več kot druge metode.

	cp	Im	pp	imU	Om	omL	imL	imS	ALL
DR	0.857	0.571	0.625	0.400	0.750	0.800	0.000	0.000	0.692
DR-W	0.678	0.597	0.769	0.829	0.950	1.000	0.000	0.000	0.702
BAYES	0.916	0.545	0.808	0.486	0.850	0.200	0.000	0.000	0.744
RF	0.986	0.818	0.846	0.543	0.900	0.600	0.000	0.000	0.857
RF-NEAR	0.986	0.844	0.846	0.543	0.850	0.600	0.000	0.000	0.860
TREE	0.958	0.701	0.769	0.800	0.500	0.000	0.000	0.000	0.801
knn	0.986	0.870	0.846	0.257	0.350	0.600	0.000	0.000	0.807
knnKERN	0.986	0.844	0.846	0.429	0.300	0.800	0.000	0.000	0.818

Tabela 13. Klasifikacijska točnost za problem E.coli z LOO.

Glass identification database

Večina primerov ima svojega dvojnika v drugem razredu, kar je tu posledica slabo prekrivajočih zalog vrednosti večine atributov, kot to lahko vidimo v Tabeli 14, če pogledamo srednjo vrednost in standardno deviacijo. Najbolj izstopa atribut Si, ki ima vedno najvišjo

vrednost in je s tem za metodo irelevanten, tako da bi ga lahko brez izgube informacije izpustili. Vidimo, da DR-W bolje prenaša to težavo, saj imajo določeni primeri veliko dvojnikov znotraj istega razreda in s tem pri klasifikaciji tehtnico nagnejo v prid svojega razreda, kar posledično pomeni tudi večjo verjetnost zadetka.

Atribut	Min	Max	Mean	SD
Fe	0	0,51	0,057	0,0974
Ba	0	3,15	0,175	0,4972
K	0	6,21	0,4971	0,6522
Al	0,29	3,5	1,4449	0,4993
RI	1,5112	1,5339	1,5184	0,003
Mg	0	4,49	2,6845	1,4424
Ca	5,43	16,19	8,957	1,4232
Na	10,73	17,38	13,4079	0,8166
Si	69,81	75,41	72,6509	0,7745

Tabela 14. Zaloge vrednosti atributov problema Glass.

	1	2	3	4	5	6	7	Skupaj
DR	0,286	0,289	0,176	0,000	0,385	0,444	0,724	0,360
DR-W	0,936	0,000	0,118	0,000	0,538	0,556	0,759	0,472
BAYES	0,629	0,500	0,118	0,000	0,462	0,667	0,828	0,561
RF	0,857	0,750	0,176	0,000	0,462	0,667	0,828	0,729
RF-NEAR	0,800	0,763	0,176	0,000	0,692	0,667	0,862	0,734
TREE	0,829	0,553	0,059	0,000	0,538	0,333	0,759	0,621
knn	0,857	0,539	0,000	0,000	0,077	0,222	0,828	0,598
knnKERN	0,857	0,632	0,059	0,000	0,538	0,444	0,759	0,664

Tabela 15. Klasifikacijska točnost za problem Glass.

	1	2	3	4	5	6	7	Skupaj
DR	0,214	0,276	0,235	0,000	0,308	0,444	0,448	0,287
DR-W	0,929	0,000	0,000	0,000	0,538	0,556	0,759	0,463
BAYES	0,671	0,513	0,059	0,000	0,692	0,889	0,759	0,589
RF	0,871	0,789	0,412	0,000	0,692	0,778	0,862	0,790
RF-NEAR	0,871	0,789	0,412	0,000	0,769	0,778	0,862	0,794
TREE	0,800	0,645	0,412	0,000	0,692	0,778	0,828	0,710
knn	0,857	0,684	0,000	0,000	0,385	0,556	0,828	0,682
knnKERN	0,857	0,697	0,000	0,000	0,615	0,778	0,828	0,710

Tabela 16. Klasifikacijska točnost za problem Glass z LOO.

Kot smo že omenili pri problemu Iris, lahko težave z zalogami vrednosti atributov rešimo tako, da vse atribute skaliramo. Lastnosti atributov pri tem primeru kažejo na to, da bi bilo skaliranje smiselno. Vse atribute sem skaliral normalno:

$$x_s = \frac{x-\bar{x}}{s},$$

kjer je s vzorčni standardni odklon in $\bar{x}$ povprečna vrednost atributa.

Rezultati v Tabeli 17 kažejo, da se pri tako preoblikovanih atributih DR obnese bolje. Klasifikacija nad skalirnami atributi je označena s predpono s.

	1	2	3	4	5	6	7	**Skupaj**
sDR	0.700	0.658	0.118	0.000	0.577	0.722	0.828	0.650
DR	0.286	0.289	0.176	0.000	0.385	0.444	0.724	0.360
sDR-W	0.607	0.349	0.559	0.000	0.692	0.778	0.862	0.558
DR-W	0.936	0.000	0.118	0.000	0.538	0.556	0.759	0.472
LOO	1	2	3	4	5	6	7	**Skupaj**
sDR	0.650	0.592	0.059	0.000	0.615	0.722	0.828	0.607
DR	0.214	0.276	0.235	0.000	0.308	0.444	0.448	0.287
sDR-W	0.657	0.395	0.529	0.000	0.846	0.833	0.862	0.600
DR-W	0.929	0.000	0.000	0.000	0.538	0.556	0.759	0.463

Tabela 17. Klasifikacijska točnost za problem Glass, kjer so vsi atributi skalirani. Z zeleno so označene vrednosti, pri katerih je skaliranje atributov izboljšalo klasifikacijo.

Image segmentation data

Tudi ta problem ni naklonjen metodi DR. Za razliko od prejšnjih dveh pa tokrat ni problem število identičnih primerov med razredi, temveč premajhno število učnih primerov za dimenzijo 19, saj so ti znotraj razreda med seboj precej različni. Ostale metode delujejo.

	BRICKFACE	**SKY**	**FOLIAGE**	**CEMENT**	**WINDOW**	**PATH**	**GRASS**	**SKUPAJ**
DR	0,770	0,667	0,850	0,475	0,610	0,258	0,990	0,662
DR-W	0,483	0,983	0,690	0,197	0,273	0,980	0,987	0,656
BAYES	0,560	0,997	0,467	0,273	0,570	0,933	0,900	0,671
RF	0,983	1,000	0,910	0,923	0,873	0,993	1,000	0,955
RF-NEAR	0,983	1,000	0,913	0,920	0,863	0,997	0,997	0,953
TREE	0,823	1,000	0,827	0,890	0,693	0,970	0,933	0,877
knn	0,983	0,997	0,753	0,487	0,723	0,843	0,993	0,826
knnKERN	0,983	1,000	0,873	0,613	0,763	0,980	0,997	0,887

Tabela 18. Klasifikacijska točnost za problem Segmentation.

Pri tem problemu smo ravno tako opazili precej raznolike atribute in znova poizkusili s skaliranjem. Tudi tu se je klasifikacijska točnost dvignila, in sicer nad obe knn metodi.

	BRICKFACE	**SKY**	**FOLIAGE**	**CEMENT**	**WINDOW**	**PATH**	**GRASS**	**SKUPAJ**
sDR	0.972	0.997	0.743	0.763	0.840	0.987	0.993	0.898
DR	0.770	0.667	0.850	0.475	0.610	0.258	0.990	0.662
sDR-W	0.893	0.997	0.817	0.387	0.543	0.990	0.993	0.802
DR-W	0.483	0.983	0.690	0.197	0.273	0.980	0.987	0.656

Tabela 19. Klasifikacijska točnost za problem Glass, kjer so vsi atributi skalirani. Z zeleno so označene vrednosti, pri katerih je skaliranje atributov izboljšalo klasifikacijo.

Poker hand dataset

S kratkim razmislekom ugotovimo, da je problem neprimeren za metodo DR, kar kažejo tudi porazni rezultati v Tabeli 20. Razlog je v tem, da vrstni red vrednosti atributov ne pomeni ničesar, temveč vpliva na pripadnost razredu njihova kombinacija. Od tod sklepamo, zakaj je edini razred, ki da dober rezultat, barvna lestvica, saj so atributi za barve vedno enaki in zaradi predstavitve (1-4) pogosto predstavljajo prvih pet v vrstnem redu vrednosti vseh atributov, ostali pa so lahko različni. Kombinacij je veliko, vendar je velik del vrstnega reda določen. V bazi sem med testnimi primeri našel le en primer, kjer atributi barve niso med prvimi petimi in zato je bil le en primer napačno klasificiran. Seveda to velja le za DR-W, ki postavi veliko utež na vsakega od petih atributov barve znotraj razreda 8. Tudi sicer se ta metoda bolje obnese na bolj kompleksnih razredih (od lestvice oziroma razreda 4 naprej) od vseh ostalih metod.

	0	1	2	3	4	5	6	7	8	9	**Skupaj**
DR	0,257	0,267	0,217	0,176	0,120	0,142	0,093	0,000	0,000	0,000	0,256
DR-W	0,031	0,001	0,033	0,006	0,192	0,224	0,096	0,291	0,917	0,000	0,019
BAYES	0,999	0,001	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,501
RF	0,861	0,536	0,002	0,010	0,000	0,002	0,000	0,000	0,000	0,000	0,658
TREE	0,786	0,517	0,018	0,061	0,006	0,000	0,000	0,000	0,000	0,000	0,614
knn	0,750	0,384	0,002	0,001	0,000	0,070	0,000	0,000	0,000	0,000	0,538
knnKERN	0,692	0,465	0,009	0,006	0,000	0,146	0,000	0,000	0,000	0,000	0,544

Tabela 20. Klasifikacijska točnost za problem Poker.

Vse to ne zmanjša dejstva, da je metoda neprimerna za reševanje tega problema, saj atributi nastopajo v parih in vsak par pomeni povsem enako, ne glede na katerem od petih mest se nahaja. Metoda nima načina, da bi to povezavo prepoznala ali upoštevala.

Skupni rezultati

Na hitro poglejmo, kako se je metoda odrezala pri prvi fazi testiranja. Problem Poker sem izločil, ker metoda RF-NEAR na tem primeru zaradi prevelikega števila testnih primerov ni delovala.

Klasifikacijska točnost	BAYES	RF	RF-NEAR	TREE	knn	knnKERN	DR	DR - W
Hill-Valley	0.5165	0.5479	0.5594	0.5198	0.4983	0.4934	0.9818	0.9274
WPBC	0.6753	0.7784	0.7629	0.7062	0.7577	0.6598	0.7206	0.6018
WPBC LOO	0.6340	0.8041	0.8093	0.7216	0.7680	0.6598	0.6959	0.5619
BCW	0.9707	0.9663	0.9678	0.9473	0.9414	0.9517	0.8631	0.9217
BCW LOO	0.9707	0.9722	0.9722	0.9575	0.9517	0.9605	0.8148	0.9224
WDBC	0.9332	0.9525	0.9525	0.9227	0.9402	0.9420	0.8383	0.8541
WDBC LOO	0.9367	0.9631	0.9631	0.9402	0.9420	0.9508	0.7680	0.8524
Iris	0.8333	0.9600	0.9433	0.9467	0.9267	0.9500	0.6667	0.6667
Iris LOO	0.8800	0.9533	0.9533	0.9667	0.9467	0.9500	0.6667	0.6667
E.coli	0.7411	0.8363	0.8482	0.7917	0.7560	0.7768	0.7307	0.5774
E.coli LOO	0.7440	0.8571	0.8601	0.8006	0.8065	0.8185	0.6920	0.7024
Glass	0.5607	0.7290	0.7336	0.6215	0.5981	0.6636	0.6495	0.5584
Glass LOO	0.5888	0.7897	0.7944	0.7103	0.6822	0.7103	0.6075	0.6005
Segmentation	0.6714	0.9548	0.9533	0.8767	0.8257	0.8871	0.8983	0.8024
Povprečje	**0.7612**	**0.8618**	**0.8624**	**0.8164**	**0.8101**	**0.8124**	**0.7567**	**0.7297**
RANG	**6**	**2**	**1**	**3**	**5**	**4**	**7**	**8**

Tabela 21. Primerjava klasifikacijske točnosti vseh metod.

Brez dvoma sta obe metodi naključnih gozdov razred zase v ospredju. Obe metodi razpršenega kodiranja pa razred zase na začelju.

Rang	BAYES	RF	RF-NEAR	TREE	knn	knnKERN	DR	DR - W
Hill-Valley	6	4	3	5	7	8	1	2
WPBC	6	1	2	5	3	7	4	8
WPBC LOO	7	2	1	4	3	6	5	8
BCW	1	3	2	5	6	4	8	7
BCW LOO	3	1	1	5	6	4	8	7
WDBC	5	1	1	6	4	3	8	7
WDBC LOO	6	1	1	5	4	3	8	7
Iris	6	1	4	3	5	2	7	7
Iris LOO	6	2	2	1	5	4	7	7
E.coli	6	2	1	3	5	4	7	8
E.coli LOO	6	2	1	5	4	3	8	7
Glass	7	2	1	5	6	3	4	8
Glass LOO	8	2	1	3	5	3	6	7
Segmentation	8	1	2	5	6	4	3	7
Povprečje	**5.8**	**1.8**	**1.6**	**4.3**	**4.9**	**4.1**	**6.0**	**6.9**
RANG	**6**	**2**	**1**	**4**	**5**	**3**	**7**	**8**

Tabela 22. Primerjava klasifikacijske točnosti vseh metod skozi rang.

Slika se ne spremeni, če pogledamo Tabelo 21 še na dva drugačna načina: z rangiranimi rezultati v Tabeli 22 in z relativno uspešnostjo v Tabeli 23, kjer najboljša metoda šteje kot stoodstotna uspešnost.

Relativna točnost	BAYES	RF	RF-NEAR	TREE	knn	knnKERN	DR	DR - W
Hill-Valley	0.526	0.558	0.570	0.529	0.508	0.503	1.000	0.945
WPBC	0.868	1.000	0.980	0.907	0.974	0.848	0.926	0.773
WPBC LOO	0.783	0.994	1.000	0.892	0.949	0.815	0.860	0.694
BCW	1.000	0.995	0.997	0.976	0.970	0.980	0.889	0.949
BCW LOO	0.998	1.000	1.000	0.985	0.979	0.988	0.838	0.949
WDBC	0.980	1.000	1.000	0.969	0.987	0.989	0.880	0.897
WDBC LOO	0.973	1.000	1.000	0.976	0.978	0.987	0.797	0.885
Iris	0.868	1.000	0.983	0.986	0.965	0.990	0.694	0.694
Iris LOO	0.910	0.986	0.986	1.000	0.979	0.983	0.690	0.690
E.coli	0.874	0.986	1.000	0.933	0.891	0.916	0.861	0.681
E.coli LOO	0.865	0.997	1.000	0.931	0.938	0.952	0.804	0.817
Glass	0.764	0.994	1.000	0.847	0.815	0.904	0.885	0.761
Glass LOO	0.741	0.994	1.000	0.894	0.859	0.894	0.765	0.756
Segmentation	0.703	1.000	0.999	0.918	0.865	0.929	0.941	0.840
Povprečje	**0.847**	**0.965**	**0.965**	**0.910**	**0.904**	**0.906**	**0.845**	**0.809**
RANG	**6**	**2**	**1**	**3**	**5**	**4**	**7**	**8**

Tabela 23. Primerjava relativne klasifikacijske točnosti vseh metod.

BIRDS

Za zaključek prve faze testiranja smo izbrali množico podatkov BIRDS. To smo storil iz dveh razlogov:

1. Gre za problem, ki je bil uporabljen v člankih [4] [5], ki sta predstavila metodo DR. Zaradi slabih rezultatov pri večini problemov, na katerih smo testirali, smo se želeli prepričati, da česa ne počnemo narobe. Če bi rezultati tudi v tem primeru močno zaostajali za ostalimi metodami, bi se morali pošteno zamisliti.

2. Gre za fleksibilno množico podatkov, ki vsebuje veliko primerov, in se jo da prirediti na praktično poljubno dimenzijo, kar pomeni, da si z njo lahko pomagamo pri testiranju vseh pomembnih lastnosti metode.

Težava s to množico podatkov je bila, da smo jo morali najprej pretvoriti iz strnjene binarne oblike v tako, ki je primerna za učenje. To pomeni, da smo binarno sliko razbili na N kvandrantov. Vsak kvandrant je predstavljal en atribut posameznega primera, njegova vrednost pa je število pik (enic), ki se nahajajo znotraj njega.

Najprej smo želeli preveriti, kako na klasifikacijsko točnost vpliva število učnih primerov in kako število atributov. Pri vsaki dimenziji smo vzeli 10 različnih delitev učne in

testne množice. Od 5 odstotkov učnih in 95 odstotkov testnih do 95 odstotkov učnih in 5 odstotkov testnih primerov.

Pri dimenziji 15 se DR sicer odreže najslabše, če gledamo povprečno vrednost, vendar pa je razlog majhna dimenzija, ki omogoča majhen prostor. Ta s povečanjem števila učnih primerov kmalu postane prenasičen, saj primeri zabrišejo meje med razredi. To se vidi v padajoči uspešnosti klasifikacije pri večanju deleža učne množice. Enak trend se pojavi tudi pri metodah knn.

3x5	BAYES	RF	TREE	knn	knnKERN	DR	DR - W
5% U - 95% T	0,7101	0,8961	0,7795	0,8509	0,8489	0,7945	0,7091
15% U - 85% T	0,7139	0,9194	0,8262	0,8227	0,8526	0,7134	0,7098
25% U - 75% T	0,7131	0,9287	0,8445	0,8047	0,8084	0,6651	0,7095
35% U - 65% T	0,7137	0,9339	0,8542	0,7735	0,7676	0,6287	0,7090
45% U - 55% T	0,7141	0,9367	0,8616	0,7371	0,7753	0,6058	0,7095
55% U - 45% T	0,7142	0,9402	0,8674	0,7646	0,7484	0,5869	0,7083
65% U - 35% T	0,7144	0,9420	0,8722	0,7501	0,7470	0,5727	0,7087
75% U - 25% T	0,7152	0,9438	0,8760	0,6995	0,6813	0,5587	0,7095
85% U - 15% T	0,7137	0,9455	0,8779	0,7244	0,7046	0,5477	0,7090
95% U - 5% T	0,7160	0,9473	0,8814	0,7033	0,6956	0,5389	0,7089
Povprečje	**0,7138**	**0,9334**	**0,8541**	**0,7631**	**0,7630**	**0,6212**	**0,7091**
RANG	**5**	**1**	**2**	**3**	**4**	**7**	**6**

Tabela 24. Klasifikacijska točnost za problem BIRDS s 15 atributi.

Stvari se izboljšajo pri 24 atributih. Le RF ima boljši rezultat. Toda tudi tu pri velikem številu učnih primerov pride do nasičenja in klasifikacijska točnost začne padati. Zanimivo je, da je različica z utežmi edina poslabšala rezultat.

4x6	BAYES	RF	TREE	knn	knnKERN	DR	DR - W
5% U - 95% T	0,7606	0,9280	0,8114	0,8933	0,8979	0,9148	0,5954
15% U - 85% T	0,7623	0,9477	0,8569	0,9209	0,9216	0,9340	0,5951
25% U - 75% T	0,7631	0,9547	0,8747	0,9224	0,9238	0,9375	0,5944
35% U - 65% T	0,7634	0,9593	0,8853	0,9232	0,9208	0,9384	0,5938
45% U - 55% T	0,7637	0,9620	0,8919	0,9167	0,9287	0,9376	0,5940
55% U - 45% T	0,7632	0,9641	0,8983	0,9227	0,9228	0,9362	0,5951
65% U - 35% T	0,7634	0,9660	0,9021	0,9180	0,9174	0,9352	0,5946
75% U - 25% T	0,7630	0,9671	0,9055	0,9048	0,9086	0,9342	0,5951
85% U - 15% T	0,7634	0,9690	0,9081	0,9139	0,9192	0,9323	0,5940
95% U - 5% T	0,7659	0,9683	0,9110	0,9147	0,9186	0,9328	0,5953
Povprečje	**0,7632**	**0,9586**	**0,8845**	**0,9151**	**0,9180**	**0,9333**	**0,5947**
RANG	**6**	**1**	**5**	**4**	**3**	**2**	**7**

Tabela 25. Klasifikacijska točnost za problem BIRDS s 24 atributi.

Najboljše rezultate sem zabeležil pri dimenziji 35. Število atributov glede na ločljivost skeniranih primerov je tu v najboljšem ravnovesju za DR. Knn ima sicer neznatno boljšo povprečno klasifikacijsko točnost, toda večanje učne množice pripelje DR do večje točnosti.

5x7	BAYES	RF	TREE	knn	knnKERN	DR	DR - W
5% U - 95% T	0,7880	0,9432	0,8262	0,9015	0,9153	0,8891	0,5068
15% U - 85% T	0,7895	0,9589	0,8723	0,9341	0,9379	0,9264	0,5120
25% U - 75% T	0,7900	0,9647	0,8889	0,9434	0,9487	0,9385	0,5049
35% U - 65% T	0,7897	0,9686	0,8991	0,9460	0,9513	0,9456	0,5106
45% U - 55% T	0,7899	0,9708	0,9053	0,9488	0,9526	0,9501	0,5094
55% U - 45% T	0,7904	0,9729	0,9096	0,9519	0,9536	0,9533	0,5099
65% U - 35% T	0,7898	0,9744	0,9143	0,9542	0,9549	0,9556	0,5103
75% U - 25% T	0,7898	0,9749	0,9172	0,9513	0,9533	0,9580	0,5006
85% U - 15% T	0,7908	0,9750	0,9195	0,9530	0,9540	0,9583	0,5110
95% U - 5% T	0,7895	0,9770	0,9222	0,9523	0,9540	0,9608	0,5114
Povprečje	**0,7898**	**0,9680**	**0,8975**	**0,9436**	**0,9476**	**0,9436**	**0,5087**
RANG	**6**	**1**	**5**	**3**	**2**	**4**	**7**

Tabela 26. Klasifikacijska točnost za problem BIRDS s 35 atributi.

Z dodatnim večanjem dimenzije klasifikacijska točnost pri ostalih metodah še naprej raste, medtem ko pri DR začne padati.

6x8	BAYES	RF	TREE	knn	knnKERN	DR	DR - W
5% U - 95% T	0,7915	0,9453	0,8274	0,8941	0,9122	0,8316	0,4284
15% U - 85% T	0,7933	0,9633	0,8762	0,9333	0,9419	0,8869	0,4261
25% U - 75% T	0,7940	0,9691	0,8914	0,9441	0,9502	0,9070	0,4288
35% U - 65% T	0,7949	0,9723	0,9024	0,9495	0,9554	0,9189	0,4266
45% U - 55% T	0,7941	0,9746	0,9100	0,9540	0,9587	0,9255	0,4293
55% U - 45% T	0,7949	0,9762	0,9136	0,9568	0,9606	0,9316	0,4270
65% U - 35% T	0,7949	0,9778	0,9173	0,9581	0,9625	0,9352	0,4297
75% U - 25% T	0,7940	0,9784	0,9199	0,9588	0,9642	0,9397	0,4264
85% U - 15% T	0,7952	0,9789	0,9238	0,9615	0,9655	0,9431	0,4271
95% U - 5% T	0,7973	0,9811	0,9250	0,9615	0,9647	0,9439	0,4273
Povprečje	**0,7944**	**0,9717**	**0,9007**	**0,9472**	**0,9536**	**0,9163**	**0,4277**
RANG	**6**	**1**	**5**	**3**	**2**	**4**	**7**

Tabela 27. Klasifikacijska točnost za problem BIRDS z 48 atributi.

Pri 63 atributih klasifikacijska točnost pri DR že precej pade. Razlog za to spet tiči v sami metodi. Z večanjem števila atributov dobimo manjše sektorje, ki posledično pokrivajo manjše število pik. Pri tem zato prihaja do veliko večjih odstopanj v vrstnem redu vrednosti atributov in se zato precej manj primerov znotraj istega razreda prekriva, kar vpliva na klasifikacijo, saj se pri testnih primeri hitro najde nova specifična razporeditev. To pomeni, da če je skeniran simbol pomaknjen le za nekaj točk v katero koli smer, lahko to pripelje do popolnoma drugačnega primera, ne glede na to, da je napisan isti simbol. Podobno velja za

debelino črte. Prostor se krepko poveča in povečevanje števila primerov še vedno izboljšuje klasifikacijsko točnost, toda bolj bi pomagalo, če bi simbole skenirali z višjo ločljivostjo.

7x9	BAYES	RF	TREE	knn	knnKERN	DR	DR - W
5% U - 95% T	0,8002	0,9493	0,8283	0,8964	0,9089	0,7780	0,3841
15% U - 85% T	0,7994	0,9654	0,8732	0,9321	0,9425	0,8426	0,3831
25% U - 75% T	0,8006	0,9711	0,8912	0,9451	0,9522	0,8675	0,3824
35% U - 65% T	0,8011	0,9735	0,9007	0,9510	0,9579	0,8818	0,3843
45% U - 55% T	0,8005	0,9761	0,9084	0,9559	0,9608	0,8915	0,3842
55% U - 45% T	0,8005	0,9778	0,9135	0,9590	0,9641	0,8991	0,3850
65% U - 35% T	0,8015	0,9791	0,9181	0,9598	0,9656	0,9044	0,3842
75% U - 25% T	0,8006	0,9797	0,9214	0,9628	0,9677	0,9121	0,3831
85% U - 15% T	0,8013	0,9800	0,9238	0,9640	0,9683	0,9154	0,3830
95% U - 5% T	0,8000	0,9809	0,9254	0,9662	0,9705	0,9160	0,3843
AVG	**0,8006**	**0,9733**	**0,9004**	**0,9492**	**0,9559**	**0,8808**	**0,3838**
RANG	**6**	**1**	**4**	**3**	**2**	**5**	**7**

Tabela 28. Klasifikacijska točnost za problem BIRDS s 63 atributi.

Problem BIRDS je do neke mere pisan na kožo metodi DR, saj ima vrstni red atributov nek smiseln pomen. Da bi lahko metoda konkurirala in prekosila ostale metode, morajo biti izpolnjeni določeni pogoji. Tako večanje dimenzije kot števila učnih primerov ima lahko na klasifikacijo pozitiven ali negativen vpliv. Pri tem problemu je vidno, da pri majhni dimenziji ne smemo pretiravati z učnimi primeri in da jih pri veliki dimenziji potrebujemo čim več.

4.4. Primerjava metod

Poglejmo še, ali so razlike glede uspešnosti klasifikacije med metodami statistično značilne. Za primerjavo metod sem uporabil Wilcoxonov test predznačenih rangov, kot je priporočeno [2]. Testiramo predpostavko, da metodi delujeta enako dobro.

Za test sem uporabil 14 množic podatkov in zato velja po Wilcoxonovi tabeli kritičnih vrednosti, da mora manjša izmed vsot rangov pozitivnih in negativnih razlik za statistično značilnost na nivoju 0,05 biti manjša ali enaka 21, na nivoju 0,02 manjša ali enaka 16 ter manjša ali enaka 13 na nivoju 0,01. Ta vsota je označena s T.

	T
BAYES	42
TREE	39
RF	14
knn	30
knnKERNEL	27

Tabela 29. Wilcoxonov test

BAYES, TREE, knn in knnKERNEL imajo T večji od 21, torej predpostavka za te metode zdrži. Drugače je z metodo RF, ki je značilna na nivoju 0,02.

RF-NEAR sem lahko testiral le na 8 množicah in zato pri T = 8 ne kaže statistične značilnosti, saj bi moral biti T največ 4.

	DR	BAYES	RAZLIKA	RANG
BCW	0.863	0.971	0.108	7
BIRDS 15	0.621	0.714	0.093	5
BIRDS 24	0.933	0.763	-0.170	11
BIRDS 35	0.944	0.790	-0.154	9
BIRDS 48	0.916	0.794	-0.122	8
BIRDS 63	0.881	0.801	-0.080	3
E.coli	0.731	0.741	0.010	1
Glass	0.650	0.561	-0.089	4
Hill-Valley	0.982	0.517	-0.465	14
Iris	0.667	0.833	0.167	10
Poker	0.256	0.501	0.244	13
Segmentation	0.898	0.671	-0.227	12
WDBC	0.838	0.933	0.095	6
WPBC	0.721	0.675	-0.045	2

Tabela 30. Primerjava DR in BAYES

	DR	TREE	RAZLIKA	RANG
BCW	0.863	0.947	0.084	9
BIRDS 15	0.621	0.854	0.233	11
BIRDS 24	0.933	0.885	-0.049	7
BIRDS 35	0.944	0.897	-0.046	6
BIRDS 48	0.916	0.901	-0.016	2
BIRDS 63	0.881	0.900	0.020	3
E.coli	0.731	0.792	0.061	8
Glass	0.650	0.621	-0.028	5
Hill-Valley	0.982	0.520	-0.462	14
Iris	0.667	0.947	0.280	12
Poker	0.256	0.614	0.358	13
Segmentation	0.898	0.877	-0.022	4
WDBC	0.838	0.923	0.084	10
WPBC	0.721	0.706	-0.014	1

Tabela 31. Primerjava DR in TREE

	DR	knn	RAZLIKA	RANG
BCW	0.863	0.941	0.078	9
BIRDS 15	0.621	0.763	0.142	11
BIRDS 24	0.933	0.915	-0.018	2
BIRDS 35	0.944	0.944	0.000	1
BIRDS 48	0.916	0.947	0.031	4
BIRDS 63	0.881	0.949	0.068	7
E.coli	0.731	0.756	0.025	3
Glass	0.650	0.598	-0.051	6
Hill-Valley	0.982	0.498	-0.483	14
Iris	0.667	0.927	0.260	12
Poker	0.256	0.538	0.282	13
Segmentation	0.898	0.826	-0.073	8
WDBC	0.838	0.940	0.102	10
WPBC	0.721	0.758	0.037	5

Tabela 32. Primerjava DR in knn

	DR	knnKERN	RAZLIKA	RANG
BCW	0.863	0.952	0.089	9
BIRDS 15	0.621	0.763	0.142	11
BIRDS 24	0.933	0.918	-0.015	4
BIRDS 35	0.944	0.948	0.004	1
BIRDS 48	0.916	0.954	0.037	5
BIRDS 63	0.881	0.956	0.075	8
E.coli	0.731	0.777	0.046	6
Glass	0.650	0.664	0.014	3
Hill-Valley	0.982	0.493	-0.488	14
Iris	0.667	0.950	0.283	12
Poker	0.256	0.544	0.288	13
Segmentation	0.898	0.887	-0.011	2
WDBC	0.838	0.942	0.104	10
WPBC	0.721	0.660	-0.061	7

Tabela 33. Primerjava DR in knnKERN

	DR	RF	RAZLIKA	RANG
BCW	0.863	0.966	0.103	8
BIRDS 15	0.621	0.933	0.312	12
BIRDS 24	0.933	0.959	0.025	2
BIRDS 35	0.944	0.968	0.024	1
BIRDS 48	0.916	0.972	0.055	3
BIRDS 63	0.881	0.973	0.092	7
E.coli	0.731	0.836	0.106	9
Glass	0.650	0.729	0.079	6
Hill-Valley	0.982	0.548	-0.434	14
Iris	0.667	0.960	0.293	11
Poker	0.256	0.658	0.402	13
Segmentation	0.898	0.955	0.056	4
WDBC	0.838	0.953	0.114	10
WPBC	0.721	0.778	0.058	5

Tabela 34. Primerjava DR in RF.

	DR	RF-NEAR	RAZLIKA	RANG
BCW	0.863	0.968	0.105	4
Ecoli	0.731	0.848	0.118	6
Glass	0.650	0.734	0.084	3
Hill-Valley	0.982	0.559	-0.422	8
Iris	0.667	0.943	0.277	7
Segmentation	0.898	0.953	0.055	2
WDBC	0.838	0.953	0.114	5
WPBC	0.721	0.763	0.042	1

Tabela 35. Primerjava DR in RF-NEAR

Z izjemo metode RF lahko trdimo, da metode s stališča klasifikacijske točnosti delujejo enako dobro kot DR, medtem ko metoda RF deluje bolje. To bi lahko trdili tudi za RF-NEAR, ki redno kaže boljše rezultate tudi od RF.

4.5. Prostorska zahtevnost

V drugi fazi testiranja smo se osredotočili na porabo pomnilnika. Želeli smo opazovati porabo glede na število učnih primerov in dimenzijo problema. Testirali smo na problemu BIRDS.

Prvotno smo želeli meriti maksimalno porabo pomnilnika med delovanjem metode, toda R nekonsistentno sprošča prostor po končanem delu. Pridobljeni podatki so bili nezanesljivi, saj so močno nihali pod enakimi pogoji . Na koncu smo se odločili, da merimo le velikost pomnilnika, ki ga zaseda klasifikacijski model. Ta je enak zasedenemu pomnilniku po zaključku učenja in pred klasifikacijo novih primerov.

Iz Slik 3 in 4 vidimo, da večje število učnih primerov opazno vpliva na porabo pomnilnika pri vseh metodah, medtem ko to ne velja nujno za večanje števila atributov. Nekaj ključnih točk:

– DR ima prepričljivo najskromnejšo porabo pomnilnika pri dimenziji 15: 157 KB in 450 KB. Najbližje je BAYES s 633 KB in 3,1 MB

– Število atributov na porabo pomnilnika pri DR vpliva precej bolj kot število učnih primerov. Razlog ni le to, da je treba voditi več seznamov, temveč so pri večji dimenziji primeri bolj različni in se ponovi manj indeksov v seznamih, torej jih je treba hraniti več. Podobno velja za število razredov. Manj kot jih je, manjša bo poraba, kar pa ne velja za knn.

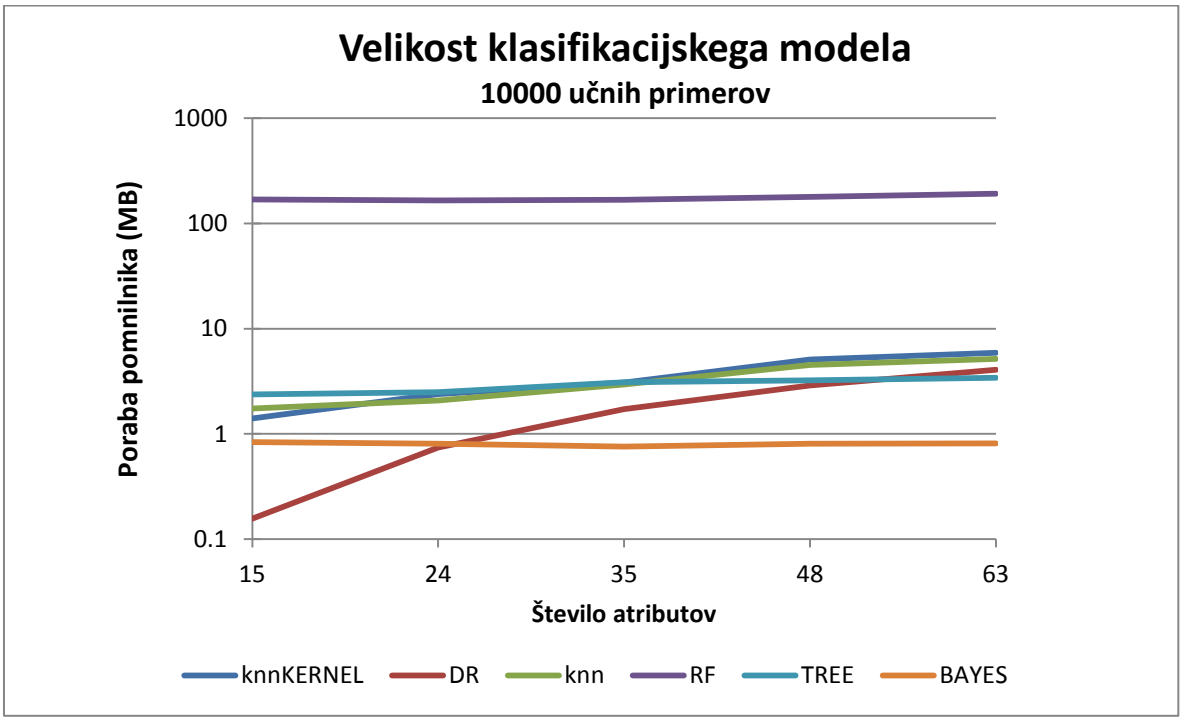
– V primerjavi s knn pri DR poraba pomnilnika raste počasneje do dimenzije 35, od tam naprej pa malce hitreje, kar je morda še en indikator, kje je optimum metode za ta problem.

– RF je razred zase pri porabi. Pri najmanj učnih primerih se giblje od 169 MB do 192 MB, pri največ primerih pa od 2,6 GB do 2,9 GB.

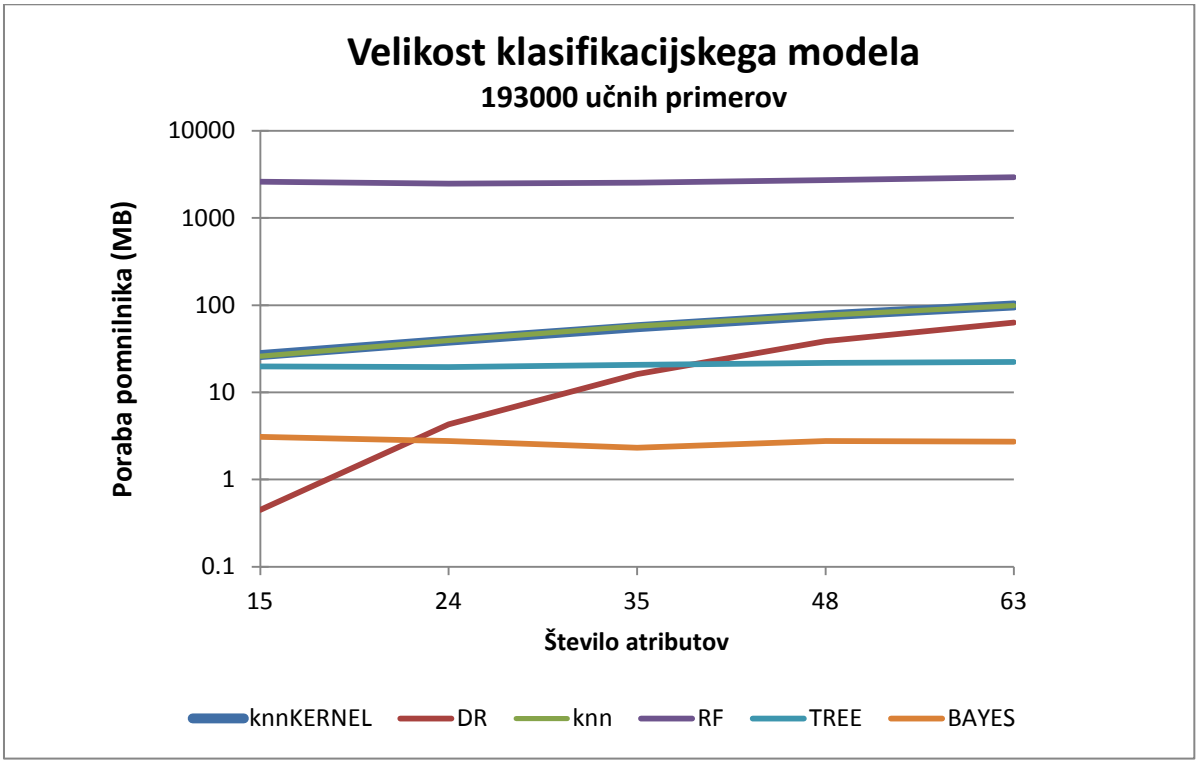
N_L=10000	**15**	**24**	**35**	**48**	**63**
DR	0,16	0,74	1,72	2,88	4,06
knn	1,75	2,07	2,95	4,53	5,17
N_L=193000					
DR	0,45	4,31	16,15	38,78	63,12
knn	26,04	39,27	57,39	76,67	98,82

Tabela 36. Primerjava porabe pomnilnika pri DR in knn v MB. N_L je velikost učne množice.

– BAYES porabi enako pomnilnika ne glede na število atributov, na TREE pa ima zelo majhen vpliv.



Slika 3. Poraba pomnilnika; 10000 učnih primerov.



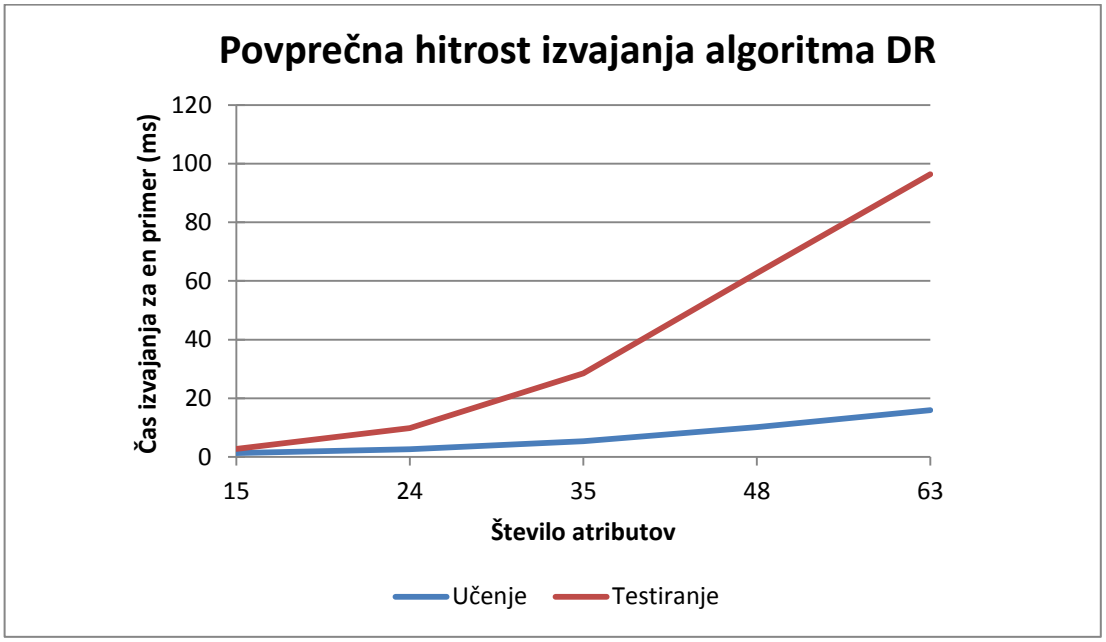
Slika 4. Poraba pomnilnika; 193000 učnih primerov.

Glede na rezultate je predpostavka, da število učnih primerov ne vpliva na čas klasifikacije DR, povezana s tem, koliko prostora je treba preiskati. Opaziti je, da se prostor

povečuje in bo čas težko konstanten. Pri majhni dimenziji smo temu zelo blizu, pri večjih pa ne. Test porabe časa bo to lahko potrdil ali zanikal.

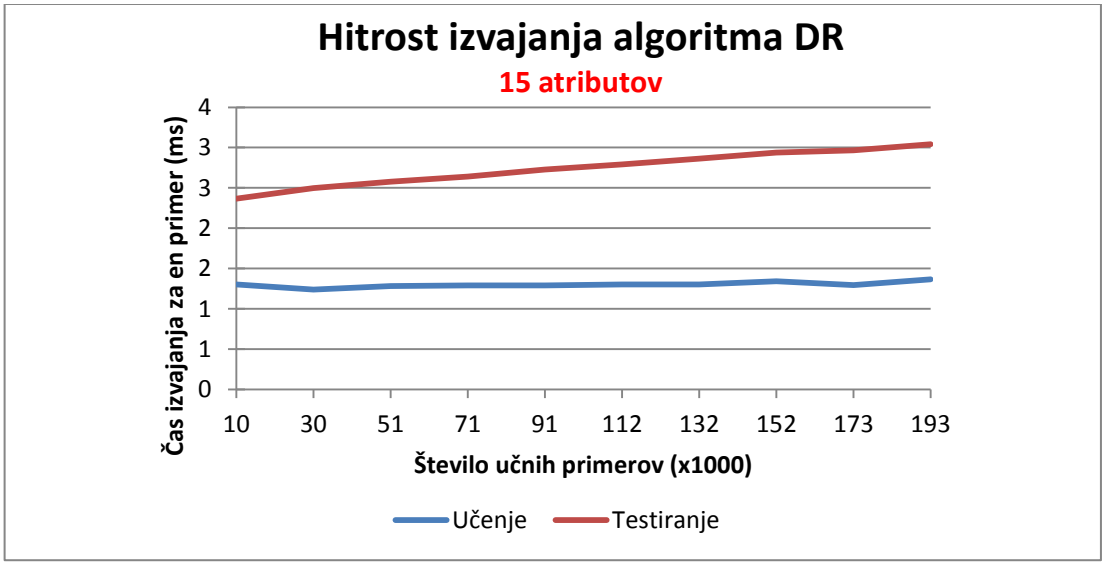
4.6. Časovna zahtevnost

Če si ogledamo rezultate porabe časa, vidimo, da je povezana s porabo prostora. Čas pri povečevanju dimenzije nad 35 veliko hitreje narašča.

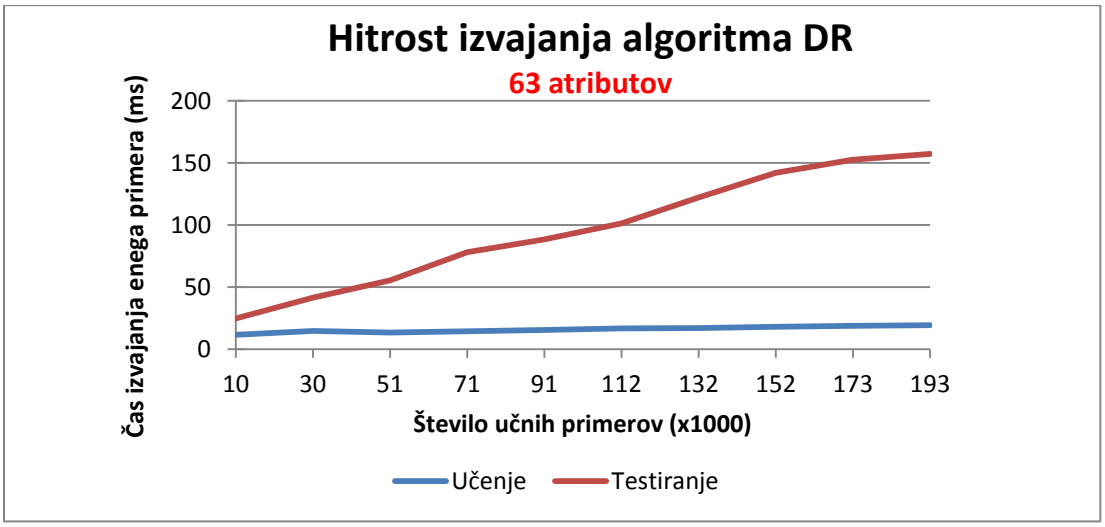


Slika 5. Povprečna hitrost izvajanja DR.

Pri povečevanju števila učnih primerov čas klasifikacije do neke točke konstantno narašča, potem pa se rast upočasni, ker se pojavlja vse manj unikatnih primerov. Večja kot je dimenzija, bolj se bo na hitrosti poznalo večanje števila učnih primerov.

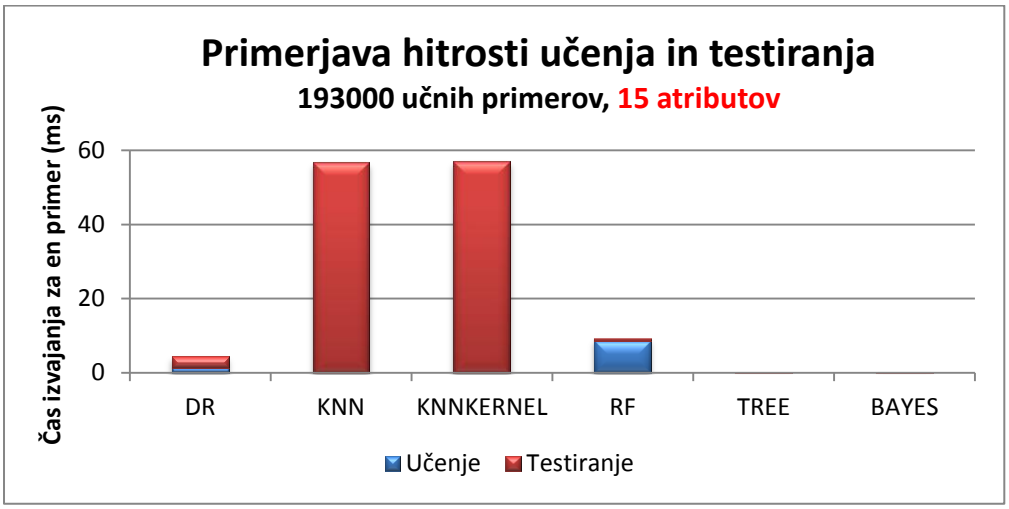


Slika 6. Hitrost izvajanja DR; N = 15.

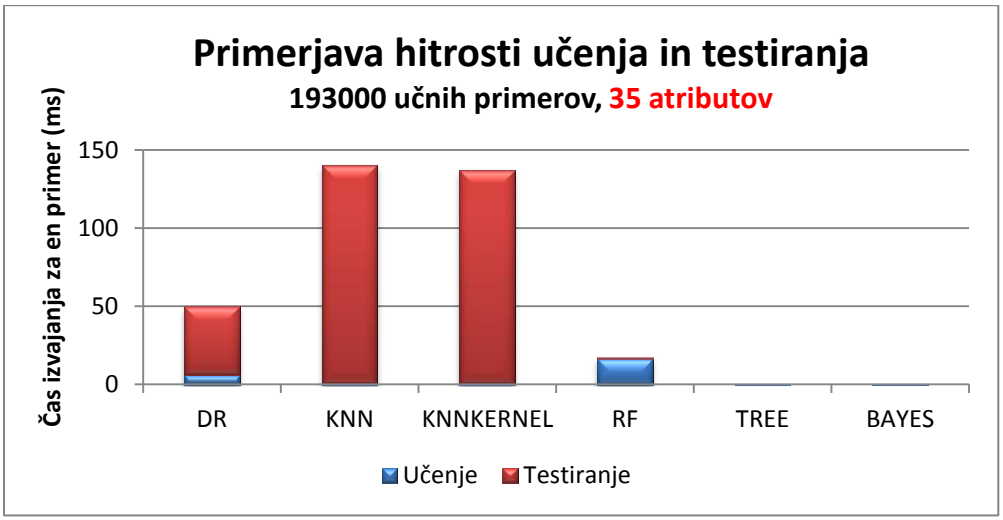


Slika 7. Hitrost izvajanja DR; N = 63.

Hitrost učenja je težava samo pri RF in DR, le da pri prvem hkrati predstavlja tudi bolj ali manj celotno časovno zahtevnost. Hitrost klasifikacije RF je konstantno okoli 0,5 ms, medtem ko pri DR narašča s številom atributov, a še vedno počasneje kot pri knn.

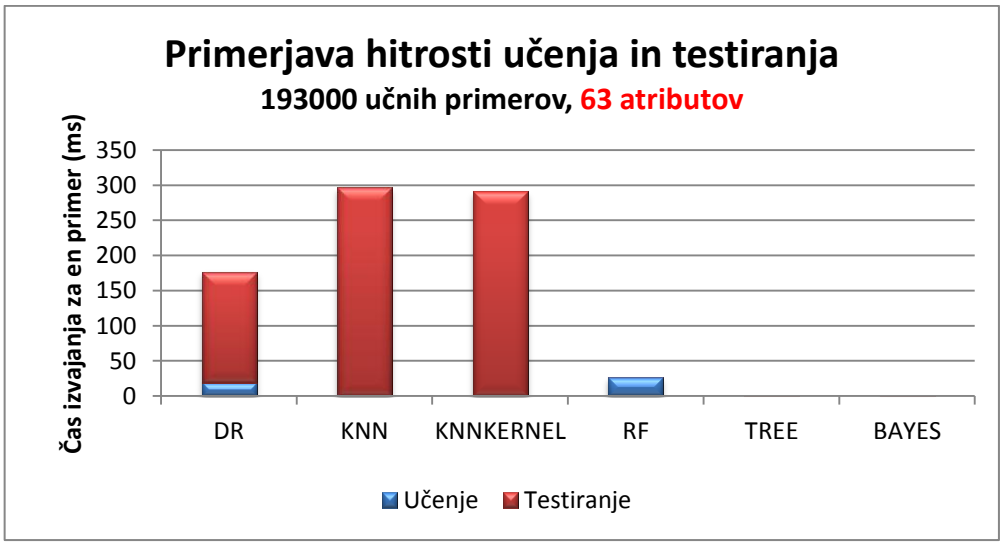


Slika 8. Primerjava hitrosti učenja in testiranja; N = 15.



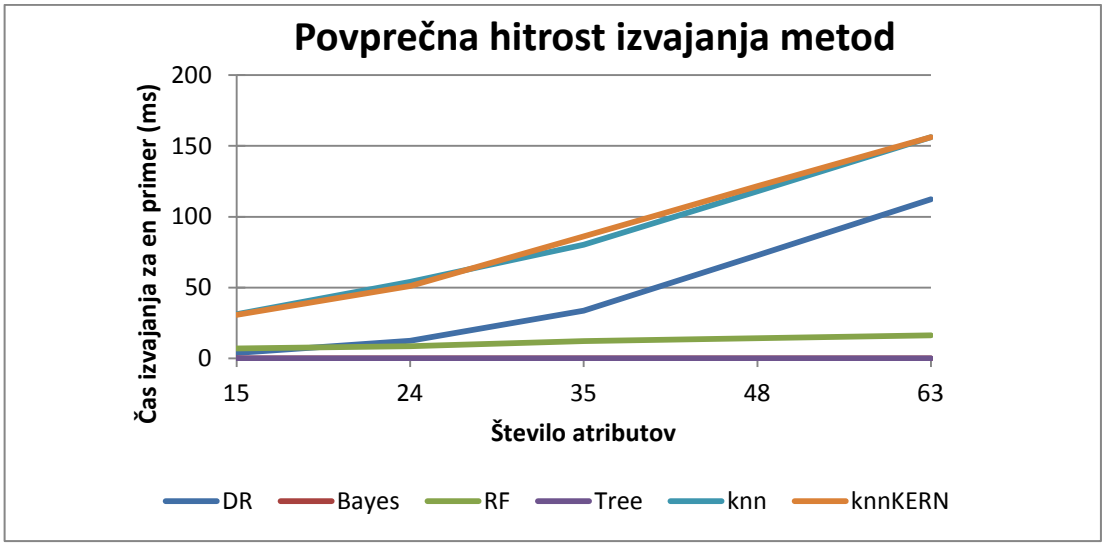
Slika 9. Primerjava hitrosti učenja in testiranja; N = 35.

Pri največjem številu učnih primerov in N = 15 je hitrost klasifikacije DR 3 ms, pri knn metodah pa 56 ms, pri N = 63 je hitrost klasifikacije DR 175 ms, pri knn metodah pa 290 ms.



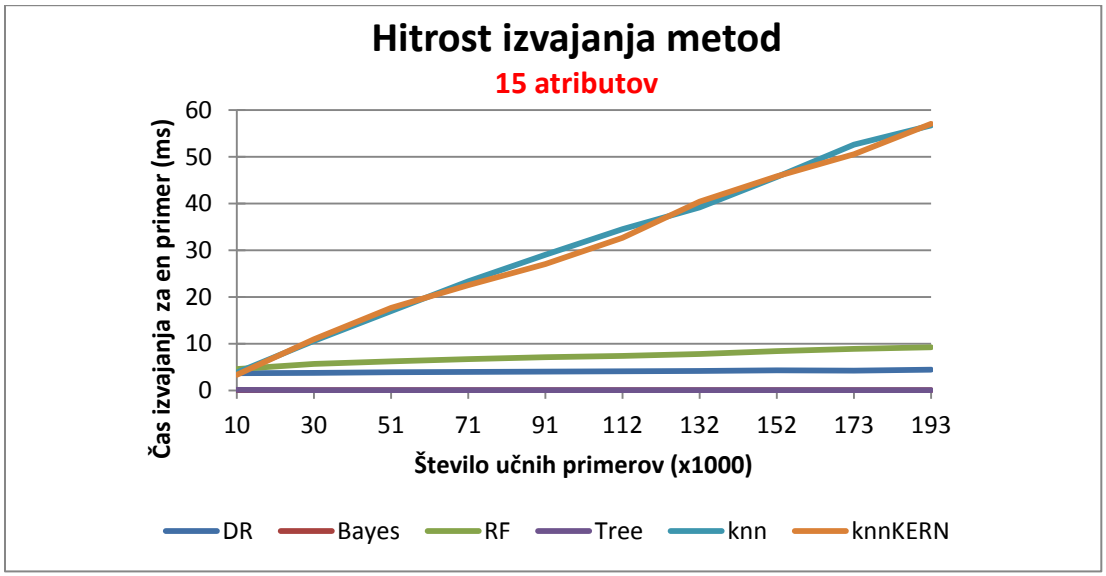
Slika 10. Primerjava hitrosti učenja in testiranja; N = 63.

Pri TREE in BAYES se hitrost klasifikacije giblje med 4 in 30 µs, pri čemer je BAYES skoraj dvakrat hitrejši, kar je praktično zanemarljivo glede na ostale metode.

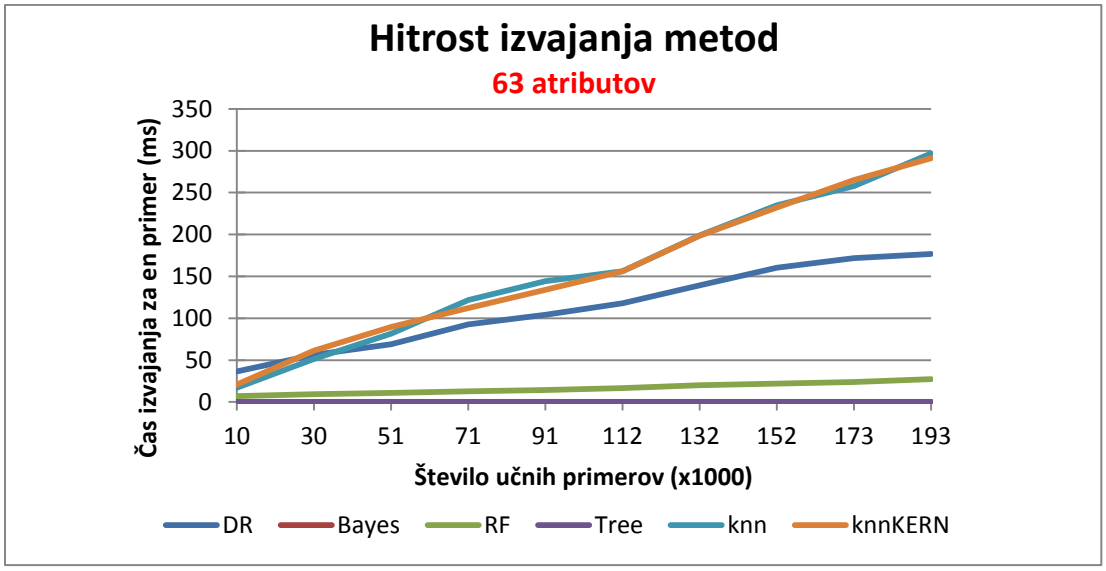


Slika 11. Povprečna hitrost izvajanja metod čez vse učne množice; učenje + testiranje.

Ali torej drži, da metoda DR ohranja konstantno hitrost klasifikacije ne glede na velikost učne množice? Sodeč po Grafu 12, to pri manjših dimenzijah drži. Pri večjih dimenzijah pa to ni res. V primerjavi s knn, pri katerem čas raste premosorazmerno z velikostjo učne množice, čas klasifikacije raste počasneje. Bolj kot povečujemo število učnih primerov, bolj se ta rast upočasnjuje.



Slika 12. Hitrost izvajanja metod; učenje + testiranje; N = 15.



Slika 13. Hitrost izvajanja metod; učenje + testiranje; N = 63.

5. Zaključek

Ko govorimo o trendih pri podatkovnem rudarjenju in strojnem učenju, ne moremo mimo pojma "big data". Količina podatkov, ki jo je treba pregledati, postaja gromozanska in s tem je treba poiskati algoritme, ki te podatke lahko obvladajo z razumno porabo virov.

Metoda razpršenega kodiranja naslavlja ta problem. Če pogledamo teoretično osnovo, nedvomno drži, da prostorska in časovna zahtevnost ostajata konstantni ne glede na število učnih primerov. Kar pa ni takoj očitno, je, da je prostorska zahtevnost ogromna. Če želimo hraniti vektor uteži, potrebujemo najmanj 2^N bitov prostora, kjer je N dimenzija problema. Pri dimenziji N = 15 ni to nič posebnega, pri N = 35 pa potrebujemo najmanj 4 GB prostora za vsak razred. Glede na to, da smo srečali tudi problem z N = 100, bi potrebovali vse pomnilniške kapacitete tega sveta in bi zadovoljili milijardinko potreb tega problema.

S prehodom na razpršeno tabelo in hranjenjem neničelnih uteži smo zahteve spravili v obvladljivo območje, in tako uspeli testirati in primerjati metodo z drugimi. Če so izpolnjeni določeni pogoji, metoda dobro opravi z nalogo velikega števila učnih primerov pri porabi prostora, časa, kot tudi klasifikacijske točnosti.

Pri nizkih dimenzijah je prostorska poraba majhna in metoda deluje hitro, žal pa ni dovolj dobra klasifikacijska točnost, če ne gre za problem, ki je primeren za to metodo. Slednje predstavlja največjo težavo. Veliko število problemov nima popolnoma neodvisnih atributov ali pa je pomembna njihova vrednost. Problemi, kjer je vrstni red pomemben, so zato bolj primerni za to metodo. Seveda pa obstajajo tudi popolnoma neprimerni, in kljub temu, da nekatere lahko prevedemo na bolj primerno obliko z uporabo skaliranja atributov, metoda vseeno ni dovolj splošna.

Če imamo problem, za katerega je smiselno uporabiti metodo DR in smo pripravljeni sprejeti nekoliko slabšo klasifikacijsko točnost, lahko pridobimo pri prostorski porabi v primerjavi z naključnimi drevesi. Precej prihranimo pri prostorski in časovni porabi tudi glede na metodo najbližjih sosedov. Tam velja, da več kot je učnih primerov, večji je ta prihranek.

Morebitne izboljšave so možne na tri načine.

Nismo veterani programiranja v okolju R in verjetno bi bolj izkušen uporabnik znal metodo sprogramirati bolje. Morda bi bilo smiselno implementacijo izvesti v jeziku C in s tem dodatno pridobiti pri hitrosti izvajanja metode.

Ne vidimo, kako bi zmanjšali prostorsko porabo, a bi bila ena od možnosti za izboljšavo časovne zahtevnosti optimizacija seznamov v razpršenih tabelah. Tu mislimo predvsem na to, da bi jih uredili in bi s tem učinkoviteje iskali v njih. Učinek je odvisen od implementacije v internih R-jevih funkcijah.

Glede klasifikacijske točnosti bi iskali boljše načine za uporabo uteži. Naša različica je v določenih primerih dala dobre rezultate, drugod pa slabe. Omenili smo iskanje unikatnih indeksov, ki sami po sebi žal ne zadostujejo za klasifikacijo, če pa bi jih spremenili zgolj v bolj vplivne uteži, bi morda dobili boljše rezultate. Po kakšnem principu določiti uteži za te indekse, ni trivialno. Lahko bi bili odločujoči, lahko pa bi bili bolj uravnovešeni glede na velikost seznama, število razredov in tudi raznolikosti seznamov med tabelami.

Ideja metode preprečuje, da bi bila uporabna za vse probleme, na katerih bi jo radi uporabili, in izboljšave tega ne bodo spremenile. Toda tam, kjer DR deluje dobro, lahko uspešno nadomesti druge metode.

Literatura

[1] BIRDS character pattern database 2007, BIRDS Systems Research Insutitute, Inc. Dosopno na: http://www.geocities.jp/onex_lab/birdsdb/birdsdb_eng.html [doseg januar 2013]

[2] J. Demšar, "Statistically correct comparison of classifiers over multiple datasets", *Journal of Machine Learning Research 7*, januar 2006, str. 1–30

[3] A. Frank, A. Asuncion (2010). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. Dostopno na: http://archive.ics.uci.edu/ml [doseg januar 2013]

[4] T. Kobayashi, M. Nakagawa, "Pattern recognition by distributed coding: test and analysis of the power space similarity method", v zborniku *9th International Workshop on Frontiers of Handwriting Recognition*, oktober 2004, str. 389–394

[5] T. Kobayashi, I. Shimizu, "A Linear Classification Method in a Very High Dimensional Space Using Distributed Representation", v zborniku *6th International Conference on Machine Learning and Data Mining 2009*, julij 2009, str. 137–147

[6] I. Kononeko, M. Robnik Šikonja, *Inteligentni sistemi*, Ljubljana: Založba FE in FRI, 2010, pogl. 1.

[7] M. Robnik-Šikonja, "Improving Random Forests", v zborniku Boulicaut et al.(eds): *Machine Learning, Proceedings of European Conference on Machine Learning 2004*, Springer, Berlin, 2004, str. 359–370

[8] W. H. Wolberg, Breast Cancer Wisconsin (Original) Data Set, University of Wisconsin Hospitals, Madison, Wisconsin, USA, 1991. Dostopno na: http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29 [doseg januar 2013]