Znanstveno-raziskovalni prispevek ■

Razvrščanje profilov izražanja genov z metodami strojnega učenja

Tomaž Curk, Blaž Zupan, Gaj Vidmar

Izvleček. Nedavno razvita tehnologija mikromrež DNK omogoča opazovanje časovne aktivnosti (profilov izražanja) večjega števila genov, kar nam lahko pomaga pri določanju funkcije genov. V primeru, da za določen nabor genov njihovo funkcijo že poznamo, lahko iz podatkov gradimo modele, ki napovejo funkcijo genov na podlagi njihovih časovnih aktivnosti. Na dveh zbirkah podatkov smo pokazali, da so metode strojnega učenja primerne za indukcijo napovednih modelov funkcij genov. Navkljub splošnemu prepričanju na področju bioinformatike, da je za obravnavo tovrstnih podatkov najprimernejša metoda podpornih vektorjev, smo pokazali, da dosti bolj preprosta in časovno veliko bolj učinkovita metoda naivnega Bayesa dosega podobne oziroma celo boljše rezultate. Razvili smo tudi novo metodo kvalitativnega modeliranja profilov izražanja genov, ki se je v napovedih izkazala za manj točno, lahko pa uspešno služi za vizualizacijo aktivnosti genov v času.

Classification of gene expression profiles with machine learning

Abstract. The recently developed DNA microarray technology provides a way to measure expression profiles of a large number of genes and assign functions to genes. Given prior knowledge on gene functions and the microarray data, one can build models that predict functions of genes based on their expression profiles. We demonstrate on two genetic data sets that machine learning methods are suitable for induction of such prediction models. Surprisingly, naive Bayesian method proved at least as accurate but much faster than the currently prevailing support vector machines. We also present a new method for qualitative modelling of gene expression profiles, which makes less accurate predictions but it may be very useful for visualization of gene expression profiles.

■ **Infor Med Slov** 2003; 8(1): 69-80

Instituciji avtorjev: Fakulteta za računalništvo in informatiko, Univerza v Ljubljani (TC, BZ), Department of Human and Molecular Genetics, Baylor College of Medicine (BZ), Inštitut za biomedicinsko informatiko, Medicinska fakulteta, Univerza v Ljubljani (GV).

Kontaktna oseba: Tomaž Curk, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, Tržaška 25, 1000 Ljubljana. email: tomaz.curk@fri.uni-lj.si.

Uvod

V zadnjih nekaj letih je na področju molekularne biologije in genetike prišlo do tehnološkega preobrata s pojavom in razvojem tehnologije mikromrež DNK, ki omogoča merjenje nivojev aktivnosti oziroma izražanja več deset tisoč genov hkrati. Analiza tovrstnih podatkov je eden izmed pristopov, ki ga genetiki na področju funkcijske genomike (ang. *functional genomics*) lahko uporabljajo za sklepanje o funkciji genov. Vedenje, kdaj in kje je nek gen izražen, nam namreč lahko olajša nalogo določanja njegove funkcije.[1]

Profil izražanja gena je časovno zaporedje nivoja aktivnosti gena in pove, kako se aktivnost gena spreminja s časom. Profile izražanja genov dobimo tako, da v nekem biološkem eksperimentu opravimo več merjenj izražanja genov ob različnih časih. Slika 1 prikazuje izraze triindvajsetih genov amebe *D. discoideum* v trinajstih točkah časa (razmik med dvema točkama je dve uri).

Velike količine tako zbranih eksperimentalnih podatkov je možno in smotrno obdelati le z uporabo računalnikov in specializiranih metod. Od slednjih so za to področje še posebej zanimive metode za odkrivanje znanja iz podatkov. Med njimi glede na preliminarne študije največ obetajo metode za strojno učenje, ki pa jih je zaradi specifičnosti področja potrebno ustrezno prilagoditi. Strojno učenje se torej ponuja kot ena od tehnik obdelave tovrstnih podatkov in razvoja napovednih modelov, njegova uporaba na tem področju pa je še v povojih. Potrebno je še podrobno preučiti uporabnost posameznih metod strojnega učenja in razviti specializirane metode za reševanje problema določanja funkcije gena iz njegovega profila izražanja.

V pričujočem prispevku smo preučili uporabnost različnih metod strojnega učenja za gradnjo modelov, ki lahko določen gen na osnovi njegovega profila izražanja razvrstijo v neko funkcijsko skupino. Med sabo smo primerjali različne metode strojnega učenja in preučevali njihovo uspešnost na dveh različnih naborih podatkov. Navkljub splošnemu prepričanju, da je za reševanje tovrstnih problemov najbolj primerna metoda podpornih vektorjev, smo pokazali, da preprostejša in časovno veliko bolj učinkovita metoda naivnega Bayesa daje vsaj enako dobre rezultate. V prispevku predstavljamo tudi novo metodo, ki temelji na kvalitativni obravnavi profilov izražanja genov; ta sicer daje slabše rezultate z vidika napovedi, a je zanimiva in uporabna pri iskanju trendov genskih izrazov in vizualizaciji.

Uporabljeni genetski podatki

Za študijo smo uporabili dve zbirki podatkov, ki opisujeta izražanje genov v različnih poskusnih pogojih. Uporabili smo podatke o kvasovki *S. cerevisiae*, ki so jih pripravili Brown in soavtorji,[2,3] in podatke o amebi *D. discoideum*, ki jih je zbrala skupina Gad Shaulsky-ja iz Baylor College of Medicine v Houstonu, Texas. Nivo izražanja gena je navadno definiran kot logaritem razmerja med nivojem izražanja gena v testnem (poskusnem) tkivu in nivojem izražanja istega gena v kontrolnem (normalnem) tkivu.

Podatke o izražanju genov lahko predstavimo s tabelo (primer je tabela 1), kjer so vsi podatki o enem genu zapisani v eni vrstici. Vsak gen, torej vsaka vrstica v tabeli tako predstavlja en učni primer za nadzorovano strojno učenje, kjer so atributi nivoji izražanja gena v različnih točkah časa (profil izražanja), razred pa je funkcijska skupina, v katero je gen razvrščen. Druga možnost predstavitve profilov izražanja genov je graf, kjer abscisna os predstavlja čas meritev, ordinata pa izražanje gena. Uporabna je predvsem za lažji vpogled v podatke in jo genetiki precej uporabljajo pri določanju funkcije genov. Primera takšnih grafov sta sliki 1 in 3.

Podatki o kvasovki *S. cerevisiae*

Podatki o *S. cerevisiae* obsegajo 2467 učnih primerov z 79 atributi (meritvami) iz osmih bioloških poskusov: prehod iz anaerobnega v aerobno dihanje (ang. *diauxic shift*, 7 meritev),

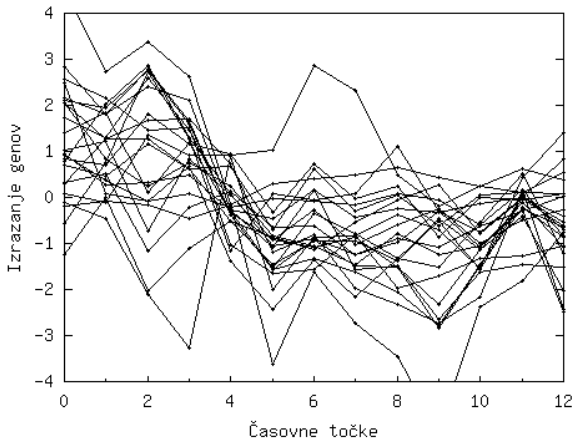
mitoza (ang. *mitotic cell division cycle*, 18+14+15 meritev), sporulacija (ang. *sporulation*, 11 meritev), temperaturni šok (ang. *temperature shock*, 4+6 meritev) in shujševalni šok (ang. *reducing shock*, 4 meritve). Gene so izbrali Eisen in soavtorji[5] glede na razpoložljive in natančne funkcijske oznake (ang. *functional annotations*) in jih razvrstili v šest funkcijskih skupin na podlagi oznak Munich Information Center for Protein Sequences Yeast Genome Database (MYGD): TCA - Tricarboxylic-acid pathway (17 primerov), Resp - Respiration chain complexes (30 primerov), Ribo - Cytoplasmic ribosomal proteins (121 primerov), Proteas – Proteasome (35 primerov), Hist - Histones (11 primerov), HTH - Helix-turn-helix (16 primerov), other – ostalo (2240 primerov). Podatki so bili že v obliki, ki nam je omogočila hitro in enostavno pripravo. Združiti smo morali le dve tabeli. V eni tabeli so bile meritve izražanja genov in imena genov, v drugi pa je bila vsakemu genu določena funkcijska skupina.

Podatki o amebi *D. discoideum*

Podatki o *D. discoideum* obsegajo 126 učnih primerov s 13 atributi (meritvami) iz enega poskusa, kjer je bil organizem podvržen stradanju. Izvirne meritve, dobljene neposredno iz mikromrež DNK, so vsebovale podatke o izražanju 3031 različnih genov, kjer so bili nekateri geni izmerjeni večkrat, in kjer so bile nekatere meritve nepopolne oziroma neuspele. S predhodno obdelavo smo takšne meritve izločili. Gene smo izbrali in razvrstili v skupine tako, da smo združili razpoložljive in natančne funkcijske oznake, ki so dostopne na medmrežju (http://dicty.sdsc.edu/annotationdicty.html), in delne funkcijske oznake, ki nam jih je posredoval Gad Shaulsky. Pri tem smo na njegov predlog uporabili prag *Identity* ≥ 50, ki določa verjetnost pravilnosti funkcijske oznake. Tako smo dobili 126 učnih primerov (genov), ki so razvrščeni v dve funkcijski skupini: sinteza beljakovin (ang. *protein synthesis*, 23 primerov) in ostalo (103 primeri).

Gradnja napovednih modelov s strojnim učenjem

Podatki, kjer so geni eksperimentalno opisani z genskimi izrazi ter označeni s pripadajočo funkcijo, so primerni za obravnavo s strojnim učenjem. Tu je cilj strojnega učenja gradnja modelov, ki lahko za določen gen na podlagi njegovega profila izražanja napovejo pripadnost funkcijski skupini. Na podlagi podatkov o označenih genih iz tabele 1 (prvih deset genov v tabeli) tako lahko zgradimo model, ki za določen profil izražanja gena napove verjetnost pripadnosti funkcijskima skupinama Resp in Ribo. Tak model, zgrajen z metodo *naivnega Bayesa* ($m = 10$), napove, da enajsti (nerazvrščeni) gen iz tabele 1 pripada skupini Ribo z verjetnostjo 0.93.



Slika 1 Izražanje genov funkcijske skupine sinteza beljakovin iz podatkov o D. discoideum.

Tabela 1 Izrazi desetih genov kvasovke S. cerevisiae pri prehodu iz anaerobnega v aerobno dihanje.

gen	X1	X2	X3	X4	X5	X6	X7	funkcijska skupina
YGR207C	0.04	-0.12	0.38	0.14	0.15	0.90	-0.04	Resp
YNL052W	-0.23	-0.23	-0.09	0.08	0.64	1.80	2.30	Resp
YGL187C	0.12	0.29	0.51	0.62	0.94	2.03	2.18	Resp
YGL191W	0.04	-0.07	0.03	0.03	0.89	2.49	2.35	Resp
YLR395C	0.08	0.03	0.42	0.33	0.86	1.32	1.66	Resp
YDL184C	0.19	0.28	0.58	0.30	0.30	-0.62	-1.40	Ribo
YBR048W	0.11	0.20	0.08	-0.42	-1.00	-1.51	-2.47	Ribo
YDR450W	0.44	0.10	0.11	-0.36	-0.09	-1.32	-2.00	Ribo
YHL033C	0.34	0.23	0.19	-0.18	-0.51	-1.56	-2.47	Ribo
YBR189W	-0.03	-0.30	0.03	-0.27	-0.56	-2.00	-2.64	Ribo
YPL198W	0.11	-0.20	0.01	-0.60	-0.64	-1.79	-2.18	?

V ilustracijo problema, s katerim se ukvarja članek, enajsti gen (YPL198W) ni razvrščen: cilj strojnega učenja je zgraditi napovedni model iz podatkov o prvih desetih genih ter za nerazvrščeni gen napovedati, kateri funkcijski skupini pripada.

Uporabnost tovrstnih modelov je potencialno velika, saj je za veliko večino organizmov, na katerih genetiki preučujejo osnovne genetske in biološke zakonitosti, funkcija večine genov še neznana. Modele za razvrščanje genov bi torej gradili iz množice funkcijsko določenih genov, rezultati razvrstitve nerazvrščenih genov pa bi genetikom nudili osnovo za razmišljanje in morebitno načrtovanje dodatnih poskusov, na osnovi katerih bi nedvoumno določili funkcije preostalih genov.

Cilj prispevka je bil raziskati uporabnost metod strojnega učenja na omenjenem področju. Izhajali smo iz zbirk genetskih podatkov, kjer so bili vsi geni razvrščeni v eno od funkcijskih skupin, uspešnost učenja pa smo preverili preko metod, ki so množico genov razbile na učno, t.j. množico za gradnjo modelov, in testno, t.j. množico genov, na kateri smo ovrednotili dobljene modele.

Metode strojnega učenja

Poleg *metode podpornih vektorjev* (SVM, ang. *Support Vector Machines*) smo uporabili še metodo *naivnega Bayesa*, *k-najbližjih sosedov*, *odločitveno drevo* in preizkusili več različic novo razvite metode *QMP*. Metode smo implementirali v sistemu za strojno učenje Orange.[4]

Naivni Bayesov klasifikator

Naivni Bayesov klasifikator predpostavlja pogojno neodvisnost vrednosti različnih atributov pri danem razredu. Osnovna formula Bayesovega pravila je:[8]

$$P(r_k \mid V) = P(r_k)\prod_{i=0}^{a}\frac{P(r_k \mid v_i)}{P(r_k)} \qquad (1)$$

Naloga učnega algoritma je s pomočjo učne množice podatkov oceniti apriorne verjetnosti razredov $P(r_k), k = 1...n_0$ in pogojne verjetnosti razredov $r_k, k = 1...n_0$ pri dani vrednosti v_i atributa $A_i, i = 1...a : P(r_k \mid v_i)$. Za ocenjevanje apriornih verjetnosti se navadno uporablja Laplaceov zakon zaporednosti,[8] za ocenjevanje pogojnih verjetnosti se uporablja *m*-ocena.[8] Vrednost parametra *m*-ocene smo določili z notranjim prečnim preverjanjem.

Za uporabo metode *naivnega Bayesa* smo morali najprej diskretizirati zvezne atribute (meritve izražanja genov v različnih točkah časa). Uporabili smo diskretizacijo Fayyada in Iranija[7], ki s pristopom od zgoraj navzdol (ang. *top-down*) razdeli začetni obseg vrednosti atributa v manjše intervale. Delitev lokalno poveča informacijsko vsebino diskretiziranega atributa (ang.

informativity) in se ustavi, ko je dolžina opisa (ang. *description length*) večja od pridobljene informacije.

Odločitvena drevesa

Odločitveno drevo predstavlja klasifikacijsko funkcijo, ki je hkrati simbolični opis in povzetek zakonitosti v dani problemski domeni. Sestavljeno je iz notranjih vozlišč, ki ustrezajo atributom, vej, ki ustrezajo podmnožicam vrednosti atributov, in listov, ki ustrezajo razredom. Pot v drevesu od korena do lista ustreza enemu odločitvenemu pravilu za klasifikacijo novega primera. Pri tem so pogoji (pari atribut - podmnožica vrednosti), ki jih srečamo na poti, konjunktivno povezani.[8]

Za izbor "najboljšega atributa" smo pri gradnji drevesa uporabili mero razmerje informacijskega prispevka (ang. *gain ratio*). Ker je zanesljivost ocene kvalitete atributa odvisna od števila učnih primerov, ni dobro, da se učna množica prehitro razdeli na majhne podmnožice.[8] Zato smo gradili binarno odločitveno drevo.

k-najbližjih sosedov

Učenje z algoritmom *k*-najbližjih sosedov temelji na shranjenih vseh učnih primerih. Ko želimo napovedati razred r_x novemu primeru u_x, poiščemo med učnimi primeri *k* najbližjih $u_1,...,u_k$ in pri klasifikaciji napovemo večinski razred, t.j. razred, ki mu pripada največ izmed k najbližjih sosedov. Učenja pri tej metodi skorajda ni. Glavnina procesiranja je potrebna pri klasifikaciji novega primera in je zato cena klasifikacije precej večja kot pri drugih metodah učenja.[8] Za računanje razdalj med novim in učnimi primeri smo uporabili evklidsko razdaljo.

Parameter *k* je običajno liho število. S povečevanjem parametra k povprečimo napovedi več bližnjih učnih primerov in s tem zmanjšamo verjetnost, da je vseh k učnih primerov napačnih. Po drugi strani pa z večanjem števila k povečujemo možnost, da h klasifikaciji prispevajo tudi tisti učni primeri, ki niso dovolj podobni novemu primeru. Zato je potrebno za vsak problem posebej eksperimentalno določiti optimalni k.[8]

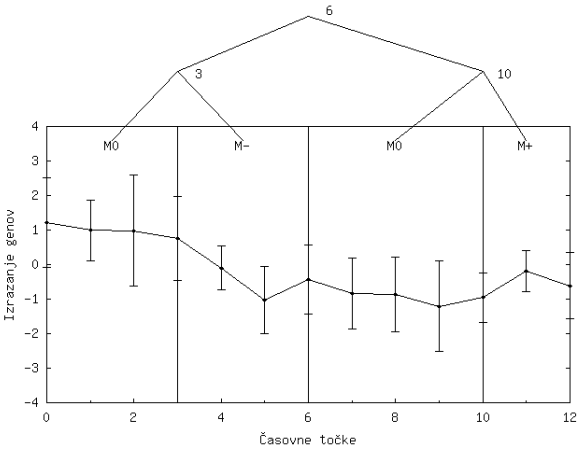
Metoda podpornih vektorjev

Profil izražanja gena si lahko predstavljamo kot točko v m-dimenzionalnem prostoru, kjer je m število meritev v profilu gena. Teoretično bi lahko za vsak razred zgradili binarni klasifikator tako, da bi v prostoru primerov določili hiperravnino, ki bi uspešno ločevala pozitivne od negativnih učnih primerov.

Večina realnih problemov pa vsebuje neločljive podatke (ang. *nonseparable data*), za katere takšna hiperravnina ne obstaja. Problem lahko rešimo z uporabo tako imenovane jedrne funkcije, ki preslika prostor atributov vhodnih podatkov v prostor atributov višje dimenzije (ang. *feature space*), kjer lahko poiščemo ločitveno hiperravnino. Umetno ločevanje učnih primerov na ta način izpostavlja učni sistem nevarnosti generiranja trivialnih rešitev in prevelikemu prileganju podatkom. Metoda podpornih vektorjev (ang. *support vector machines*, SVM)[2,3,10] elegantno rešuje vse te težave. Prevelikemu prileganju podatkom se izogne tako, da vedno izbere hiperravnino z največjo razdaljo do najbližjega učnega primera. Hiperravnino predstavimo kot linearno kombinacijo le tistih učnih točk, ki so ji dovolj blizu. Takšne točke so tipično majhna podmnožica vseh učnih točk, kar dela klasifikacijo učinkovito. Učne točke imenujemo tudi podporni vektorji, ker podpirajo oziroma določajo ločitveno hiperravnino.

Izbira pravega jedra je pomembna, saj določa mero podobnosti dveh vektorjev oziroma točk in tako izraža neko predhodno znanje o pojavu, ki ga modeliramo. Uporabljali smo dve jedrni funkciji: polinomsko jedro $K(\vec{X},\vec{Y}) = (\vec{X}\cdot\vec{Y}+1)^d$, kjer je d stopnja polinoma, in jedro z radialno bazno funkcijo (RBF) $K(\vec{X},\vec{Y}) = \exp\left(\frac{-\left\|\vec{X}\cdot\vec{Y}\right\|^2}{2\sigma^2}\right)$, kjer je

σ širinski parameter Gaussove funkcije. V poskusih Browna in soavtorjev in tudi v naših poskusih σ ustreza mediani vseh evklidskih razdalj v učni množici med vsakim pozitivnim in njemu najbližjim negativnim učnim primerom.[3]



Slika 2 Povprečni profil genov funkcijske skupine sinteza beljakovin amebe in za ta profil zgrajeno drevo kvalitativnih omejitev. Navpične črte so meje med intervali, ki so zapisane v notranjih vozliščih drevesa.

QMP - kvalitativno modeliranje profilov

Poleg uporabe navedenih metod smo razvili in uporabili tudi novo metodo, poimenovano *QMP* (ang. *qualitative modelling of gene profiles*), ki časovne aktivnosti genov modelira kvalitativno. Motivacija za razvoj metode *QMP* je način, na katerega genetiki opisujejo profile izražanja genov. Navadno namreč opisno navajajo časovne intervale, v katerih se izražanje genov ne spreminja, narašča ali pada. Iz takšnih opisov potem sklepajo o funkciji genov. Pri učenju smo želeli modelirati prav te lastnosti profilov in tako upoštevati časovnost, ki je ostale metode ne upoštevajo, saj posamezno meritev (točko na profilu) obravnavajo neodvisno od ostalih. Določiti smo želeli intervale naraščanja, padanja in konstantnega izražanja genov posamezne funkcijske skupine. Tako naučeni model omogoča lažje razumljivo razlago odločanja pri klasifikaciji novih primerov.

Učni algoritem QMP

Osnova metode *QMP* je algoritem *ep-QUIN*,[6] ki se iz numeričnih podatkov nauči binarna drevesa kvalitativnih omejitev. Metoda *QMP* je sestavljena iz *učnega* in *izvajalnega* algoritma. Učni algoritem iz množice podatkov in predznanja tvori model. Izvajalni algoritem pa uporabi naučeni model za reševanje novih problemov.[8] Razvili in preizkusili smo tri različice učnega in tri različice izvajalnega algoritma z dvema možnima vrednostima notranjega parametra (stopnja značilnosti pri statističnih testih). Preizkušali smo torej osemnajst različic ($3\times3\times2$).

Učni algoritem *QMP*(*S*)
Vhod: *S* je profil izražanja funkcijske skupine X. Je polje trojic (X_t, s_{Xt}, n_{Xt}) za vsako točko časa *t*.
Izhod: Drevo kvalitativnih omejitev.

1. naredi vozlišče za Koren drevesa
2. g := **mostConsistentQCF**(*S*)
3. $E(g)$:= cena QCF g ; $E_{best} := E(g)$
4. **for** vsak indeks t polja S **do begin**
5. razdeli profil izražanja gena S v dva dela S_{leq} in S_{grt} glede na *indeks* $\le t$
6. E_{left} := cena **mostConsistentQCF**(S_{leq})
7. E_{right} := cena **mostConsistentQCF**(S_{grt})
8. E_{tree} := cena drevesa (enačba 2)
9. **if** $E_{tree} < E_{best}$ **then begin**
10. $E_{best} := E_{tree}$; $t_{best} := t$
11. **end**
12. **end**
13. **if** $E_{best} < E(g)$ **then begin**
14. v Koren drevesa vstavi indeks t_{best}
15. razdeli S v S_{leq} in S_{grt} glede na *indeks* $\le t_{best}$
16. pod Koren vstavi poddrevesi dobljeni z **QMP**(S_{leq}) in **QMP** (S_{grt})
17. **end else** Koren je list z QCF g
18. **return** Koren

Slika 3 Učni algoritem QMP za učenje drevesa kvalitativnih omejitev.

Vhod v metodo so povprečni profili izražanja genov (vrednosti X_t) vsake funkcijske skupine in njihov standardni odklon (vrednosti s_{Xt}). Učni algoritem uporablja požrešno metodo (podobno

kot algoritmi za učenje odločitvenih dreves) in za vsako funkcijsko skupino zgradi eno drevo kvalitativnih omejitev. Naučeno drevo predstavlja delitev profila izražanja funkcijske skupine na podintervale z enakim kvalitativnim obnašanjem. Notranja vozlišča drevesa so razmejitve profila glede na indeks delitve profila, v listih pa so kvalitativno omejene funkcije QCF, ki opisujejo izražanje na podintervalu v odvisnosti od časa: konstantno (M^0), naraščajoče (M^+) ali padajoče (M^-). Algoritem je prikazan na sliki 3.

Algoritem pri izbiri najprimernejše kvalitativno omejene funkcije (QCF) v listu drevesa uporablja napako, ki temelji na najkrajši dolžini kodiranja.[6] Pri določitvi delitve v notranjem vozlišču izbere vrednost indeksa delitve, ki minimizira napako obeh poddreves. Delitev izvaja, dokler je napaka delitve manjša od napake, kjer za celoten podinterval uporabimo samo eno, najprimernejšo kvalitativno omejeno funkcijo (spremenljivka *g* v zapisu na sliki 3).

$$E_{tree} = E_{left} + E_{right} + SplitCost$$
$$SplitCost = \log_2(Splits_i) \qquad (2)$$

Cena napake (E_{tree}) v vozlišču drevesa (enačba 2) je vsota napak obeh poddreves (E_{left} in E_{right}) in cene kodiranja indeksa delitve (*SplitCost*).[6] $Splits_{\text{i}}$ je število vseh možnih delitev na dva podintervala. Cena napake v listu, za podano kvalitativno omejeno funkcijo, je dolžina kodiranja tistih parov indeksov (pod)intervala, katerih vektor spremembe ne ustreza kvalitativno omejeni funkciji.

Vektor spremembe je smer spremembe vrednosti izražanja med dvema točkama na intervalu, ki sta urejeni po naraščajoči vrednosti indeksa. Možne so tri smeri spremembe, ki sovpadajo s kvalitativno omejenimi funkcijami: pozitivna, negativna in konstantna (brez spremembe).

Razvili smo tri načine izbiranja najprimernejše kvalitativno omejene funkcije (mostConsistentQCF(*S*)) na intervalu *S*: *majority*, *votingTH* in *variance*. Metoda *majority* izbere tisto kvalitativno omejeno funkcijo, katere vektorji sprememb na intervalu so najbolj številčni. Metoda *votingTH* dela podobno kot majority. Med funkcijama M^+ in M^- izbira le, če sta podprti z dovolj velikim deležem vektorjev sprememb, sicer izbere funkcijo M^0. Metoda *variance* s statističnim testom enakosti varianc najprej ugotovi, ali interval ustreza modelu M^0. Sicer izbere med funkcijama M^+ in M^- tisto, ki je podprta z večjim številom vektorjev sprememb.

Statistično značilen vektor spremembe $v_s(i,j)$ med točkama *i* in *j* v intervalu določimo odvisno od izbranega načina računanja najprimernejše QCF. Pri metodah *majority* in *votingTH* uporabimo t-test, pri metodi *variance* pa uporabimo t-test z *Bonferronijevim popravkom*, ki je izrazito konzervativen. Notranji parameter je tako stopnja značilnosti $\alpha \in \{0.05, 0.01\}$, ki naj se uporabi za statistični test.

$$v_s(i,j) = \begin{cases} poz., \text{razlika znacilna in } X_i < X_j \\ neg., \text{razlika znacilna in } X_j > X_i \\ konst., \text{sicer} \end{cases} \qquad (3)$$

Razvrščanje z modelom QMP

Vhod v izvajalni algoritem je profil izražanja gena z neznano funkcijsko skupino. Izvajalni algoritem izračuna napako modela posamezne funkcijske skupine na vhodnem profilu. Gen klasificira v funkcijsko skupino z najmanjšo napako modela. Zaupanje (verjetnost) v odločitev za posamezni razred je nasprotno premosorazmerna z normalizirano vrednostjo izračunane napake.

Izračun napake modela oziroma kvalitativnega drevesa na vhodnem profilu poteka podobno kot v učnem algoritmu. Rekurzivno razdeli profil na podintervale glede na indekse v notranjih vozliščih drevesa, dokler ne pride do listov, kjer so zapisane kvalitativno omejene funkcije. Za vsak list izračuna ceno napake, ki jih nato skozi vozlišča sešteva proti korenu drevesa, ki predstavlja napako celotnega drevesa. Težava nastopi pri računanju napake v listih. Na prvi pogled ne moremo uporabiti statističnih testov za izračun vektorjev sprememb kot smo to počeli v učnem algoritmu

(enačba 3), ker imamo samo en, vhodni profil izražanja gena. Težavo smo poskusili odpraviti na tri načine: *THzero*, *meanDiffKeep* in *meanDiffZero*. Metoda *THzero* uporabi od uporabnika podano minimalno spremembo vrednosti T_{zero}, ki se še obravnava kot konstantna (enačba 4). Privzeta vrednost T_{zero} je 1% razlike med maksimalno in minimalno vrednostjo[6] izražanja profilov genov učne množice.

$$v_s(i,j)=\begin{cases} poz., \text{razlika znacilna in } \mathrm{X_i} + T_{zero} < X_j \\ neg., \text{razlika znacilna in } \mathrm{X_i} - T_{zero} > X_j \\ konst., \text{sicer} \end{cases} \quad (4)$$

Drugi dve metodi, *meanDiffKeep* in *meanDiffZero*, pa uporabita statistični *z*-test za določitev vektorja spremembe med točkama *i* in *j* v intervalu (3). Metoda *meanDiffZero* v primeru, da testiramo model M^0, privzame ničelno razliko med točkama. Povprečja populacij in standardni odkloni pri *z*-testu so vrednosti v točkah profila funkcijske skupine, ki smo jih izračunali v učnem algoritmu.

Obravnavanje podatkov o več poskusih

Kadar podatki o genu vsebujejo meritve iz več neodvisnih poskusov, kot je to v našem primeru pri organizmu *S. cerevisiae*, jih je potrebno obravnavati ločeno. V takšnem primeru za vsak poskus zgradimo en model. Pri razvrščanju novega gena je potrebno upoštevati napovedi vseh modelov. To smo storili na dva načina. Izbrali smo razred, ki je prejel največjo verjetnost pri enem ali več modelih, ali pa smo za vsak razred med seboj zmnožili izračunane verjetnosti modelov in izbrali razred z najvišjo tako izračunano verjetnostjo.

Ocenjevanje uspešnosti učenja

Kvaliteto dobljenih modelov oziroma uspešnost metod strojnega učenja smo ocenjevali z 10-kratnim prečnim preverjanjem, s katerim smo množico razpoložljivih učnih primerov razdelili na 10 približno enako močnih podmnožic. Učenje in ocenjevanje smo tako izvajali desetkrat. V *i*-tem izvajanju smo za ocenjevanje vzeli *i*-to podmnožico, za učenje pa preostalih devet. Za vse metode strojnega učenja smo uporabili isto razbitje na učno in testno množico.

Poleg klasifikacijske točnosti (v rezultatih označena s KT) smo uspešnost učenja merili še s ceno napake oziroma prihrankom cene (označena s prihranek) in površino pod krivuljo ROC (ang. *Receiver Operating Characteristics*, označena z AUC).[9] Statistično značilnost razlike med klasifikatorji smo določili z McNemarovim testom s stopnjo značilnosti $\alpha = 0.005$.

Za prihranek cene napak smo vzeli funkcijo, ki so jo predlagali Brown in soavtorji.[2] Cena uporabe dobljenega modela (klasifikatorja) z metodo strojnega učenja M je definirana kot $C(M) = fp(M) + 2\,fn(M)$, kjer je $fp(M)$ število negativnih primerov, ki jih klasifikator napačno klasificira za pozitivne (ang. *false positives*), in $fn(M)$ število pozitivnih primerov, ki jih napačno klasificira za negativne (ang. *false negatives*). Ceno klasifikatorja metode M primerjajo s ceno klasifikatorja ničelne metode strojnega učenja *N*, ki vse primere klasificira za negativne. Prihranek cene (ang. *cost savings*) pri uporabi metode M je tako $S(M) = C(N) - C(M)$. Večji kot je prihranek cene $S(M)$, bolj je metoda M uspešna.

Poskusi in rezultati

Poskuse smo izvajali na obeh zbirkah podatkov in primerjali uspešnost metod strojnega učenja. Ocenili smo metode *večinski klasifikator* (majority), *odločitveno drevo* (TDIDT), k-najbližjih sosedov (*k-NN*), *naivni Bayes* (nb), štiri različice *SVM* in osemnajst različic *QMP*. Gradili smo klasifikator *k-NN* z vrednostjo parametra $k = 11$, en klasifikator *SVM* z jedrom RBF (SVM RBF) in tri s polinomskimi jedri stopenj $d = 1,2,3$ (SVM p1, SMV p2 in SVM p3). Najprej smo izmerili vpliv vrednosti parametra *m* na uspešnost učenja metode *naivnega Bayesa*. Nato smo med seboj primerjali osemnajst različic metode *QMP* in za končno primerjavo izbrali najboljšo. Nazadnje smo

izvedli primerjavo med izbranimi in vsemi ostalimi metodami strojnega učenja.

Večinski klasifikator smo pri primerjavah uporabili kot referenco za najslabši in najenostavnejši možni model, ki določa spodnjo mejo uspešnosti učenja. Klasifikator prešteje učne primere, ki pripadajo posameznemu razredu, s čimer določi verjetnost vsakega razreda. Rezultat učenja je klasifikator, ki vedno vrača najbolj verjetni (večinski) razred in v fazi učenja ocenjene verjetnosti vseh razredov.[4]

Parameter *m* metode *naivnega Bayesa* smo določili z notranjim prečnim preverjanjem. V okviru (glavnega) 10-kratnega prečnega preverjanja smo z (notranjim) 5-kratnim prečnim preverjanjem na učni množici izbrali vrednost parametra $m \in \{0.1,0.2,0.5,1.0,2.0,5.0,...,10000.0\}$, pri kateri je uspešnost učenja največja (za vse tri mere). To vrednost smo uporabili za učenje na celotni učni (pod)množici trenutnega koraka glavnega prečnega preverjanja. Z dvosmernim t-testom s stopnjo značilnosti $\alpha = 0.05$ smo testirali, ali je uspešnost s tako izbranimi vrednostmi parametra statistično značilno različna od uspešnosti pri vrednosti $m = 0$. V končno primerjavo smo vključili boljšega.

Uporabljena imena klasifikatorjev *QMP* (npr. QMP.majority.THzero.0.05) so sestavljena iz štirih, s piko ločenih delov, ki opisujejo značilnosti posameznega klasifikatorja. Ime se vedno začne s *QMP*. Drugi del opisuje uporabljeno metodo pri gradnji modela, tretji del opisuje uporabljeno metodo pri klasifikaciji novih primerov, četrti del pa je lahko 0.05 ali 0.01 in opisuje uporabljeno stopnjo značilnosti za določanje statistično značilnih razlik med gradnjo modela in klasifikacijo novih primerov. Deli imen so zaradi preglednosti v tabelah smiselno okrajšani. Posamezni deli so podrobno obrazloženi v začetnem razdelku opisa metode QMP .

Kvasovka S. cerevisiae

Merili smo klasifikacijsko točnost in prihranek cene napak. Ker v podatkih nastopa več razredov, nismo merili površine pod krivuljo ROC.

Najprej smo izmerili vpliv vrednosti parametra *m* metode *naivnega Bayesa* na uspešnost učenja tako, da smo vrednost parametra *m* izbrali vnaprej in ta isti *m* uporabili pri vseh delih učenja v prečnem preverjanju. To smo ponavljali za različne vrednosti *m* in tako dobili odvisnost uspešnosti učenja od vrednosti parametra *m*. Uspešnost za obe meri je bila največja pri vrednosti $m = 2000$.

Nato smo za obe meri uspešnosti določili povprečje in standardni odklon vrednosti parametra *m* metode *naivnega Bayesa*, pri kateri je bila uspešnost učenja z uporabo notranjega prečnega preverjanja največja (tabela 2).

Tabela 2 Najboljše vrednosti parametra m.

Mera uspešnosti učenja	$\overline{m}$	s_m
Klasifikacijska točnost	1375.000	176.777
prihranek cene napak	1375.000	176.777

Povprečna vrednost in standardni odklon najboljše vrednosti parametra *m*, določene z notranjim prečnim preverjanjem za podatke o *S. cerevisiae*.

Notranje prečno preverjanje uspešno določi vrednost parametra *m* ($m = 1375.0$), saj je ta blizu točke ($m = 2000.0$), kjer sta klasifikacijska točnost in prihranek cene napak največji. Kljub temu pa razlika uspešnosti v primerjavi z uspešnostjo pri vrednosti $m = 0$ ni statistično značilna za obe meri.

Primerjava različic QMP

Primerjali smo osemnajst različic *QMP*. Vsi klasifikatorji *QMP* so slabši od *večinskega klasifikatorja*. Za obe meri dobimo enaki lestvici. McNemarov test označi vse klasifikatorje *QMP* za statistično značilno slabše od *večinskega klasifikatorja*. Najboljši klasifikator *QMP.variance.meanDiffKeep.0.05*, ki smo ga tudi izbrali za primerjavo z ostalimi metodami, je statistično značilno boljši od vseh klasifikatorjev

QMP razen dveh (*QMP.majority.meanDiffKeep.0.05* in *QMP.votingTH.meanDiffKeep.0.05*).

Primerjava metod strojnega učenja

Tabela 3 prikazuje uspešnost učenja klasifikatorjev glede na obe meri uspešnosti. Za obe meri uspešnosti dobimo isto lestvico najboljših klasifikatorjev. Statistično značilno različni pari, ki jih določi McNemarov test, so prikazani v tabeli 4. Tabela je simetrična preko diagonale. Metode so razvrščene po naraščajoči uspešnosti učenja od leve proti desni v vrstici oziroma od zgoraj navzdol v stolpcu. Najboljši klasifikator *k- NN* je statistično značilno boljši od vseh klasifikatorjev *SVM* razen dveh.

Tabela 3 Uspešnost metod strojnega učenja na podatkih o S. cerevisiae.

Mera uspešnosti	$\bar{x}_{prihranek}$	$s_{prihranek}$	$\bar{x}_{KT}$	s_{KT}
majority	425.0	0.0	0.907	0.000
SVM p1	436.7	12.6	0.923	0.017
SVM p2	464.9	9.2	0.961	0.012
SVM p3	464.6	6.8	0.960	0.009
SVM RBF	459.8	8.4	0.954	0.011
nb prihranek	459.8	8.5	0.954	0.011
k-NN	**468.5**	4.7	**0.966**	0.006
TDIDT	425.0	0.0	0.907	0.000
QMP*	384.5	39.7	0.852	0.054

*QMP.var.mDK.0.05

Ameba D. discoideum

Merili smo klasifikacijsko točnost in prihranek cene napak. Ker v podatkih nastopata samo dva razreda, smo lahko merili tudi površino pod krivuljo ROC.

Tudi tu smo na enak način kot pri kvasovki najprej izmerili vpliv vrednosti parametra *m* metode *naivnega Bayesa* na uspešnost učenja in dobili primerljive rezultate. Tabela 5 prikazuje povprečje in standardni odklon vrednosti parametra *m*, pri kateri je bila uspešnost učenja z uporabo notranjega prečnega preverjanja največja.

Tudi tu notranje prečno preverjanje uspešno določi vrednost parametra *m*. Razlika uspešnosti v primerjavi z uspešnostjo pri vrednosti $m = 0$ ni statistično značilna za vse tri mere. Uspešnost, merjena s površino pod krivuljo ROC, je skorajda konstantna in ne kaže vpliva vrednosti *m* na uspešnost učenja.

Tabela 4 Najboljše vrednosti parametra m.

Mera uspešnosti učenja	$\bar{m}$	s_m
klasifikacijska točnost	9.280	15.610
prihranek cene napak	4.290	6.414
površina pod krivuljo ROC	2.600	3.459

Povprečna vrednost in standardni odklon najboljše vrednosti parametra *m*, določene z notranjim prečnim preverjanjem za podatke o D. discoideum.

Primerjava različič QMP

Tudi tu so vsi klasifikatorji *QMP* statistično značilno slabši od *večinskega klasifikatorja*, če jih primerjamo na podlagi klasifikacijske točnosti in prihranka cene napak. Za obe meri dobimo enaki lestvici. Najboljši klasifikator *QMP* je *QMP.votingTH.THzero.0.01*. Lestvica najbolj uspešnih klasifikatorjev se spremeni, če uporabimo za oceno uspešnosti učenja površino pod krivuljo ROC. *Večinski klasifikator* v tem primeru pade na predzadnje mesto, najboljši klasifikator pa je *QMP.majority.meanDiffKeep.0.01*. Za primerjavo z ostalimi metodami smo izbrali *QMP.votingTH.THzero.0.01*.

Primerjava metod strojnega učenja

Tabela 6 prikazuje uspešnost učenja klasifikatorjev za vse tri mere uspešnosti. McNemarov test pokaže statistično značilne razlike le med klasifikatorjem *QMP* in ostalimi. Za meri klasifikacijske točnosti in cene napak je *naivni Bayes* najboljši, sledijo mu klasifikatorji *SVM* in *k-NN*.

Primerjava na podlagi površine pod krivuljo ROC nam da drugačne rezultate. *Naivni Bayes* je še vedno najboljši, sledijo mu pa *k-NN*, *SVM* in *QMP*. Metodi *TDIDT* in *majority* si delita zadnje mesto.

Zaključek

Strojno učenje se je izkazalo za uporabno na obravnavani problemski domeni. Kot bolj uspešni od metode podpornih vektorjev, čeprav ne statistično značilno, sta se izkazali metodi *naivnega Bayesa* in *k-NN*. Metoda *QMP* je manj uspešna, nam pa namesto kompleksnih matematičnih modelov daje zelo razumljive simbolne modele, ki dobro opisujejo profile izražanja genov.

Poleg razvoja nove metode *QMP*, ki lahko v prikazani inačici služi predvsem za pomoč pri vizualizaciji časovnih odzivov genov, je osnovni prispevek pričujočega dela ugotovitev, da se enostavna metoda *naivnega Bayesa* pri napovedi funkcije genov obnese vsaj tako dobro kot metoda *SVM*. Ta ugotovitev je presenetljiva, saj velja *SVM* za *de-facto* standard za tovrstno obdelavo podatkov, in je v nasprotju z nekaterimi objavljenimi rezultati.[2] Poleg preprostosti sta prednosti *naivnega Bayesa* tudi časovna učinkovitost (gradnja napovednega modela za podatke o *S. cerevisiae* traja približno 4 sekunde namesto 5 minut, kolikor jih potrebuje SVM) in možnost razlage odločitve.

Zahvala

Dr. Gad Shauslky in dr. Chad Shaw iz Baylor College of Medicine sta nam posredovala genetske podatke o *D. discoideum* ter gene iz podatkov v namene opisane analize razvrstila v funkcionalne skupine. Hvala tudi dr. Dorianu Šucu iz Fakultete za računalništvo in informatiko Univerze v Ljubljani za nasvete pri razvoju metode QMP. Delo je nastalo v okviru MŠZŠ projekta Metode odkrivanja znanj za funkcionalno genomiko (J2-3387, BZ in TC).

Tabela 5 Statistično načilne razlike med klasifikatorji na podatkih o S. cerevisiae.

	QMP	majority	TDIDT	SVM p1	SVM RBF	nb prih.	SVM p3	SVM p2	*k*-NN
QMP	*	*	*	*	*	*	*	*	*
majority	*				*	*	*	*	*
TDIDT	*				*	*	*	*	*
SVM p1	*				*	*	*	*	*
SVM RBF	*	*	*	*					*
nb prih.	*	*	*	*					*
SVM p3	*	*	*	*					
SVM p2	*	*	*	*					
k-NN	*	*	*	*	*	*			

Statistično značilne razlike (označene z zvezdico) med klasifikatorji, določene z McNemarovim testom, na podatkih o *S. cerevisiae*. Klasifikator *QMP.variance.meanDiffKeep.0.05* je poimenovan *QMP*.

Tabela 6 Uspešnost metod strojnega učenja na podatkih o D. discoideum.

Mera uspešnosti učenja	$\bar{x}_{prih.}$	$s_{prih.}$	$\bar{x}_{KT}$	s_{KT}	$\bar{x}_{AUC}$	s_{AUC}
majority	18.3	1.3	0.818	0.034	0.500	0.000
SVM p1	19.2	5.3	0.842	0.139	0.763	0.231
SVM p2	19.5	4.9	0.850	0.128	0.760	0.191
SVM p3	19.2	5.4	0.843	0.146	0.695	0.227
SVM RBF	19.8	4.5	0.859	0.124	0.810	0.205
nb prihranek	**20.7**	3.9	**0.882**	0.104	**0.858**	0.198
k-NN	19.5	4.9	0.849	0.123	0.831	0.202
TDIDT	18.3	1.3	0.818	0.034	0.500	0.000
QMP.var.mDK.0.05	-3.3	7.6	0.249	0.208	0.702	0.257

Literatura

1. Dennis C, Gallagher R (urednika): The Human Genome. Nature Publishing Group, 2001.
2. Brown MPS, Grundy WN, Lin D, Cristianini N, et al.: Knowledge-based Analysis of Microarray Gene Expression Data Using Support Vector Machines, PNAS, vol. 97(1), pp. 262-267, 2000.
3. Brown MPS, Grundy WN, Lin D, et al.: Support Vector Machine Classification of Microarray Gene Expression Data, University of California, Santa Cruz and University of Bristol, Tec. Rep. UCSC-CRL-99-09, June 1999.
4. Demšar J, Zupan B: Programski paket za strojno učenje Orange. http://magix.fri.uni-lj.si/orange, 2002.
5. Eisen M, Spellman P, Brown P, Botstein D: Cluster analysis and display of genome-wide expression patterns. PNAS, vol. 95, pp. 14863-14868, Dec. 1998.
6. Šuc D: Machine reconstruction of human control strategies, Doktorska disertacija, Fakulteta za računalništvo in informatiko, Ljubljana, 2001.
7. Fayyad UM, Irani KB: The Attribute Selection Problem in Decision Tree Generation. Proc. of AAAI-92, pp. 104-110, San Jose, CA, 1992.
8. Kononenko I: Strojno učenje, Založba FE in FRI, Ljubljana, 1997.
9. Provost F, Fawcett T:, Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions, Proc. of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), 1977.
10. Jakulin A: Knjižica orngExtn-1.0. http://ai.fri.uni-lj.si/~aleks, 2002.