STROJNO UČENJE EPITOPOV IZ PEPTIDNIH MIKROMREŽ

Mitja Luštrek
Odsek za inteligentne sisteme
Institut Jožef Stefan
Jamova cesta 39, 1000 Ljubljana, Slovenija
Telefon: +386 1 4773380; telefaks: +386 1 4773131
E-pošta: mitja.lustrek@ijs.si

POVZETEK

Epitopi so delci beljakovin, ki jih prepozna imunski sistem. Peptidne mreže omogočajo določanje velikega števila epitopov z laboratorijskimi poizkusi. Podatke, pridobljene s peptidnimi mrežami, smo uporabili za učenje klasifikatorja za prepoznavanje epitopov. Klasifikator je bil sestavljen iz ansambla osmih osnovnih klasifikatorjev, od katerih je vsak uporabljal svoj nabor atributov, združeval pa jih je meta-klasifikator. Z njim smo dosegli klasifikacijsko točnost 83,7 do 85,9 %, kar je za 0,9 do 2 odstotni točki bolje od metode, ki je sodeč po literaturi trenutno najboljša.

1 UVOD

Antigen je molekula, ki jo imunski sistem prepozna kot škodljivo, epitop pa je del antigena, ki je zaslužen za prepoznavo. Poznavanje epitopov je koristno iz vsej dveh razlogov. Mogoče jih je uporabljati v cepivih, saj izzovejo imunski odziv, na da bi bilo cepljeni osebi treba vbrizgati celoten antigen, ki utegne biti škodljiv. Poleg tega se lahko uporabljajo v diagnostične namene, saj odziv na različne epitope pokaže, katera protitelesa bolnik ima, iz tega pa je razvidno, katero bolezen ima.

Epitope je načeloma mogoče določiti z laboratorijskimi poizkusi, je pa strojno učenje kajpak cenejše; tudi če ne more povsem nadomestiti laboratorijskih poizkusov, lahko vsaj zoža nabor možnih epitopov. V našem delu smo uporabili podatke, pridobljene s peptidnimi mikromrežami, ki so povedali, kateri peptidi (delci beljakovine) vsebujejo epitope. Peptidne mikromreže omogočajo dokaj enostavno ugotavljanje imunskega odziva na veliko število peptidov. Iz teh podatkov smo s strojnim učenjem zgradili klasifikator za prepoznavanje epitopov.

Peptide, ki smo jih klasificirali, smo opisali z osmimi nabori atributov. Za vsak nabor smo izbrali algoritem za strojno učenje, ki je dal najboljši klasifikator na tistem naboru. Te klasifikatorje smo nato uporabili kot ansambel: njihove klasifikacije so služile kot atributi za meta-klasifikator, ki je dal končno odločitev o tem, ali peptid vsebuje epitop. Rezultate ansambla smo primerjali z metodo za prepoznavanje epitopov, ki je sodeč po literaturi trenutno najboljša.

2 PODATKI

Peptid je delec beljakovine, ki ga sestavljajo zaporedno vezane aminokisline. Večina peptidov v naših podatkih je bila sestavljena iz 15 aminokislin. Različnih standardnih aminokislin je 20, kar pomeni, da si vsak peptid lahko predstavljamo kot besedo, sestavljeno iz 15 znakov, pri čemer ima abeceda 20 različnih znakov. Peptidna mikromreža je ploščica, na katero se nanesejo vzorci peptidov. V našem delu je bil na mikromrežo nato nanesen IVIg, ki je mešanica protiteles zdravih darovalcev. Ta protitelesa so se vezala predvsem na peptide, ki so vsebovali epitope. Iz izmerjenega števila protiteles na vzorcu je bilo tako mogoče določiti verjetnost, da peptid vsebuje epitop.

Moč vezave protiteles, izražena s številom med 1 in 65.536, je bila izmerjena za 75.534 epitopov. Za strojno učenje smo uporabili le tiste, za katere smo bili gotovi, da res vsebujejo epitope (moč vezave nad 10.000) ali da jih res ne vsebujejo (moč vezave pod 1.000). Takih je bilo 27.278 in razdelili smo jih na učno in testno množico. Učna množica je bila sestavljena iz 3.420 peptidov z epitopi (pozitivnih) in 10.218 peptidov brez epitopov (negativnih). Testna množica je bila sestavljena iz 3.421 pozitivnih in 10.219 negativnih peptidov.

Učna in testna množica sta vsebovali trikrat več negativnih kot pozitivnih peptidov. Tako neuravnoteženi podatki lahko kvarno vplivajo na strojno učenje, zato smo uporabili dve metodi za uravnoteženje: nad- in podvzorčenje [6]. Prva metoda naredi kopije naključno izbranih primerov manj številčnega razreda (v našem primeru pozitivnih peptidov), druga pa odstrani primere bolj številčnega razreda (negativnih peptidov). Pri učenju končnega klasifikatorja smo uporabili nadvzorčenje (oversampling), ker nismo hoteli zavreči nobenih podatkov. Pri primerjanju algoritmov za strojno učenje in nastavljanju parametrov pa smo uporabili podvzorčenje (undersampling), saj da manj podatkov in je zato učenje hitrejše, poleg tega pa nadvzorčenje povzroča težave pri algoritmih za strojno učenje, ki uporabljajo notranje prečno preverjanje (saj se enak primer lahko znajde v notranje učni in testni množici).

3 POSTOPEK STROJNEGA UČENJA

Učenje ansambla klasifikatorjev za prepoznavanje epitopov je potekalo v dveh korakih. V prvem koraku smo vsak peptid p_i opisali z osmimi različnimi atributnimi vektorji $a_1(p_i)$, ..., $a_8(p_i)$, ki so opisani v podrazdelku 3.2. Na teh vektorjih smo naučili osem osnovnih klasifikatorjev C_1, ..., C_8 (vsakega z najboljšim algoritmom za strojno učenje – več o tem v podrazdelku 3.1), od katerih je vsak vrnil verjetnost, da peptid p_i vsebuje epitop. Te verjetnosti označimo s $P_1(p_i)$, ..., $P_8(p_i)$.

V drugem koraku smo iz verjetnosti, pridobljenih v prvem koraku, tvorili atributni vektor $a_M(p_i) = [P_1(p_i), ..., P_8(p_i)]$. Pomagali smo si s petkratnim prečnim preverjanjem: osnovne klasifikatorje C_1, ..., C_8 smo naučili na štirih petinah učne množice, nakar smo z njimi klasificirali peto petino in tako zanjo dobili a_M. To smo ponovili petkrat, pri čemer smo vsakič klasificirali drugo petino, kar nam je dalo a_M za celotno učno množico. Na koncu smo na atributnih vektorjih a_M naučili meta-klasifikator C_M, ki je vrnil končno verjetnost, da peptid vsebuje epitop.

Testiranje ansambla klasifikatorjev je potekalo podobno kot učenje. Vsak peptid iz testne množice q_i smo predstavili z osmimi atributnimi vektorji $a_1(q_i)$, ..., $a_8(q_i)$ in ga klasificirali z osmimi osnovnimi klasifikatorji C_1, ..., C_8. Njihove klasifikacije smo združili v nov atributni vektor $a_M(q_i)$, ki smo ga klasificirali z meta-klasifikatorjem C_M.

3.1 Izbira algoritmov in nastavljanje parametrov

Naš ansambel klasifikatorjev je uporabljal več algoritmov za strojno učenje, od katerih ima vsak svoje parametre. Poleg tega smo lahko na različne načine izračunali tudi vsakega od atributnih vektorjev, kar je bilo določeno z atributnimi parametri, opisanimi v podrazdelku 3.2. Celoten prostor algoritmov in parametrov je bil prevelik za izčrpno preiskovanje, zato smo jih izbrali in nastavili v šestih zaporednih korakih, izvedenih na učni množici.

V prvem koraku smo preizkusili 41 algoritmov iz orodja Weka [4] na atributnih vektorjih, sestavljenih iz frekvenc aminokislin v peptidu. Pri vseh algoritmih smo uporabili privzete vrednosti parametrov. Izbrali smo osem najboljših algoritmov z metodo podpornih vektorjev (SVM) na čelu. V drugem koraku smo z SVM določili začetne vrednosti atributnih parametrov za vseh osem atributnih vektorjev.

V tretjem koraku smo primerjali osem algoritmov iz prvega koraka na atributnih vektorjih z začetnimi vrednostmi parametrov iz drugega koraka. Pri tem smo nabor najboljših algoritmov zožali na tri: SVM, logistično regresijo in klasifikacijo z regresijo [3]. V četrtem koraku smo te tri algoritme preizkusili z vsemi smiselnimi vrednostmi atributnih parametrov. Algoritme smo uporabili same zase, vse tri združene s skladanjem klasifikatorjev in boljša dva (SVM in logistična regresija) združena s skladanjem. Tako smo dobili najboljši algoritem in najboljše atributne parametre za vsak atributni vektor. V petem koraku smo za vsak atributni vektor določili najbolje parametre za algoritem, izbran v četrtem koraku. Izbrani algoritmi, njihovi parametri in atributni parametri so navedeni pri opisih atributnih vektorjev v podrazdelku 3.2.

V šestem koraku smo izbrali najboljši algoritem za meta-klasifikator in najboljše vrednosti parametrov zanj. Ta algoritem je bil različica linearne regresije PACE [7] s privzetimi vrednostmi parametrov.

3.2 Atributi

V nadaljevanju je opisanih osem atributnih vektorjev, s katerimi smo predstavili vsak peptid.

Frekvence. Ta atributni vektor je sestavljen iz frekvenc aminokislin v peptidu. Peptid je razdeljen na p delov enake dolžine in frekvence se izračunajo za vsak del posebej. Vektor ima obliko:

$[A_1, C_1, ..., Z_1; ...; A_p, C_p, ..., Z_p]$.

A_i je frekvenca aminokisline A v i-tem delu peptida (vsaka aminokislina se označuje z eno črko, A pomeni alanin).

Najboljša vrednost $p = 3$. Najboljši algoritem za strojno učenje je bil SVM s $C = 1$ ter jedrom PUK z $\omega = 0.5$ in $\sigma = 2{,}5$.

Razlike frekvenc. Ta atributni vektor je sestavljen iz razlik med frekvencami aminokislin v peptidu. Te razlike so sicer razvidne že iz prejšnjega vektorja, vendar ni nujno, da jih algoritmi za strojno učenje znajo izkoristiti, če niso podane izrecno. Peptid je spet razdeljen na p delov enake dolžine. Klasifikacija je bila boljša, če smo razlikam v frekvencah dodali tudi same frekvence (velja tudi pri nekaterih naslednjih vektorjih), tako da ima vektor obliko:

$[A–C_1, ..., A–Z_1, ..., Z–Z_1; ...; A–C_p, ..., A–Z_p, ..., X–Z_p; A, ..., Z]$.

$A–C_i$ je razlika med frekvencama aminokislin A in C v i-tem delu peptida. A je frekvenca aminokisline A v celotnem peptidu.

Najboljša vrednost $p = 2$. Najboljši algoritem za strojno učenje je bil logistična regresija z $\lambda = 10^{-8}$.

Frekvence podzaporedij. Ta atributni vektor je sestavljen iz frekvenc podzaporedij v peptidu dolžin do l. Ker se malo podzaporedij ponavlja, podzaporedje definiramo ohlapno: vanje se sme vriniti do g aminokislin, ki podzaporedju ne pripadajo. Podzaporedje z vrinjenimi aminokislinami se ne šteje kot celo podzaporedje, ampak kot w^g podzaporedja, pri čemer $w \leq 1$. Upoštevamo le c najbolj pogostih podzaporedij vsake dolžine. Vektor ima obliko:

$[S_{11}, S_{12}, ..., S_{1c}; S_{21}, S_{22}, ..., S_{2c}; ...; S_{l1}, S_{l2}, ..., S_{lc}]$.

S_{ij} je frekvenca j-tega najbolj pogostega podzaporedja dolžine i.

Najboljše vrednosti parametrov so bile $l = 5$, $g = 0$ in $c = 25$. Najboljši algoritem za strojno učenje je bil SVM s $C = 1$ ter polinomskim jedrom s $p = 2$.

Lastnosti aminokislin. Ta atributni vektor je sestavljen iz povprečnih vrednosti 19 fiziokemičnih lastnosti aminokislin, npr. kislosti, prožnosti in velikosti. Peptid je razdeljen na p delov enake dolžine in lastnosti so povprečene po vsakem delu. Vektor ima obliko:

[$prop_{1,1}$, $prop_{2,1}$, ..., $prop_{19,1}$; ...; $prop_{1,p}$, $prop_{2,p}$, ..., $prop_{19,p}$; A, ..., Z].

Vrednost $prop_{i,j}$ je povprečna vrednost lastnosti i v j-tem delu peptida.

Najboljša vrednost p = 2. Najboljši algoritem za strojno učenje je bil skladanje klasifikatorjev, zgrajenih z SVM in logistično regresijo. SVM je uporabljal C = 1 ter jedro PUK z ω = 1 in σ = 2. Logistična regresija je imela λ = 0.1.

Frekvence razredov. Ta atributni vektor je sestavljen iz frekvenc razredov aminokislin glede na njihove fiziokemične lastnosti. Tak razred je npr. razred kislih aminokislin. Zaradi združitve treh izmed lastnosti, uporabljenih v prejšnjem vektorju, aminokisline razvrstimo v razrede na 17 načinov. Vektor ima obliko:

[*freq* $(\text{prop}_1)_1$, ..., *freq* $(\text{prop}_{17})_1$; ...;
freq $(\text{prop}_1)_p$, ..., *freq* $(\text{prop}_{17})_p$; A, ..., Z]

freq $(\text{prop}_i)_j$ = [$prop_i$-low_j, $prop_i$-med_j, $prop_i$-$high_j$]

Vrednosti $prop_i$-low_j, $prop_i$-med_j in $prop_i$-$high_j$ so frekvence razredov aminokislin, razvrščenih po lastnosti i, v j-tem delu peptida.

Najboljša vrednost p = 3. Najboljši algoritem za strojno učenje je bil skladanje klasifikatorjev, zgrajenih z SVM in logistično regresijo. SVM je uporabljal C = 1 in linearno jedro. Logistična regresija je imela λ = 1.

Frekvence podzaporedij razredov. Ta atributni vektor je podoben frekvencam podzaporedij, le da so podzaporedja namesto iz posamičnih aminokislin sestavljena iz razredov, kakršni nastopajo v prejšnjem vektorju. Tako npr. podzaporedje "EADC" nadomestimo z "anan", kjer "a" pomeni kislo in "n" nevtralno aminokislino. Podzaporedje ima spet lahko dolžino do l, do g vrinjenih aminokislin in ima težo w^g. Upoštevamo le c najbolj pogostih zaporedij vsake dolžine. Vektorju smo dodali tudi frekvence navadnih podzaporedij, tako da ima obliko:

[*subseq* ($prop_1$), ..., *subseq* (prop_{17});
S_{11}, ..., S_{1c}; ...; S_{l1}, S_{l2}, ..., S_{lc}]

subseq ($prop_i$) = [$prop_{i11}$, ..., $prop_{i1c}$; ...; $prop_{i\,l1}$, ..., $prop_{i\,lc}$]

Vrednost $prop_{ijk}$ je frekvenca k-tega najbolj pogostega podzaporedja dolžine j, ko so aminokisline razvrščene v razrede po lastnosti i. S_{ij} je frekvenca j-tega najbolj pogostega podzaporedja aminokislin dolžine i.

Najboljše vrednosti parametrov so bile l = 2, g = 5, w = 0.5 in c = 25. Najboljši algoritem za strojno učenje je bil skladanje klasifikatorjev, zgrajenih z SVM in logistično regresijo. SVM je uporabljal C = 0.5 in linearno jedro. Logistična regresija je imela λ = 50.

Pari. Ta atributni vektor je sestavljen iz frekvenc parov aminokislin z določeno razdaljo med pripadnikoma para. Taka frekvenca je npr. frekvenca para (A, B) z razdaljo 3. Razlog za ta atributni vektor je, da se protitelesa na epitope utegnejo vezati na dveh mestih. Ker je takih atributov zelo veliko, njihovo število zmanjšamo na dva načina. Prvi je, da s sosednjih razdalj združimo v eno. Drugi pa je, da eno ali obe aminokislini v paru nadomestimo z razredom aminokislin. Vektor ima obliko:

[*pair* (A_1, A_2), ..., *pair* (A_1, A_{n2}); ...;
pair (A_{n1}, A_1), ..., *pair* (A_{n1}, A_{n2})]

pair (A_i, A_j) = [(A_i, A_j) z d_1, ..., (A_i, A_j) z $d_{\max}$].

A_i je i-ta aminokislina ali i-ti razred aminokislin, $n1$ in $n2$ sta števili različnih aminokislin ali razredov prvega in drugega pripadnika para, d_k je k-ta razdalja ali skupina razdalj med aminokislinama ali razredoma v paru in $d_{\max}$ je največja razdalja.

Najboljša izbira za prvega pripadnika para so bile aminokisline, za drugega pa razred po aromatičnosti. Najboljša vrednost s = 5. Najboljši algoritem za strojno učenje je bil skladanje klasifikatorjev, zgrajenih z SVM in logistično regresijo. SVM je uporabljal C = 0.1 in polinomsko jedro s p = 2. Logistična regresija je imela λ = 200.

Pričvrščeni pari. Ta atributni vektor je podoben parom, le da je eden izmed pripadnikov para pričvrščen na prvo mesto peptida. Razlog za to je, da je na peptidni mikromreži zgornje (prvo) mesto peptida najlaže dostopno, zato je najbolj verjetno, da se protitelo veže nanj. Spet lahko s sosednjih razdalj združimo v eno in aminokislino nadomestimo z razredom. Vektor ima obliko:

[A_1 na d_1, ..., A_1 na $d_{\max}$; ...; A_n na d_1,..., A_n na $d_{\max}$; *first*].

A_i je i-ta aminokislina ali i-ti razred aminokislin, n je število različnih aminokislin ali razredov, d_j je j-ta razdalja ali skupina razdalj od prvega mesta, $d_{\max}$ je največja razdalja in *first* je prva aminokislina v peptidu.

Najboljša izbira so bile aminokisline (ne razredi) in s = 5. Najboljši algoritem za strojno učenje je bil SVM s C = 5 in linearnim jedrom.

4 REZULTATI

Osnovne klasifikatorje in celoten ansambel klasifikatorjev smo najprej z desetkratnim prečnim preverjanjem testirali na učni množici. Uporabili smo s podvzorčenjem uravnoteženo učno množico, saj smo algoritme za strojno učenje izbrali in jim nastavili parametre na takšni množici. Uspešnost klasifikacije smo primerjali glede na ploščino pod krivuljo ROC (AUC) in klasifikacijsko točnost. Prva odlika AUC je, da je neodvisna od praga verjetnosti, nad katero štejemo, da peptid vsebuje epitop (vsi naši klasifikatorji so namreč vračali verjetnost, ne enega od dveh razredov). Druga odlika pa je, da je pretežno neodvisna tudi od razmerja razredov v učni in testni množici. Točnost

potrebuje prag (uporabili smo 0,5) in je odvisna od razmerja razredov, je pa bolj intuitivna. Rezultati so prikazani v tabeli 1. Osnovni klasifikatorji so se izkazali za podobno dobre, a ker so delali napake na različnih primerih, je njihovo združevanje klasifikacijo opazno izboljšalo.

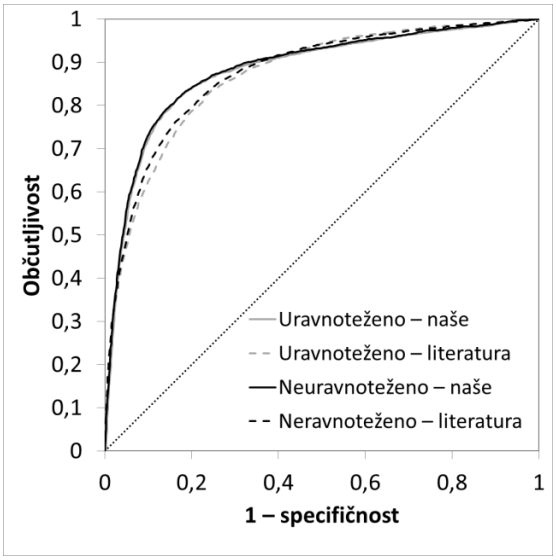
Klasifikator	AUC	Točnost
Frekvence	0,870	80,7 %
Razlike frekvenc	0,868	80,3 %
Frekvence podzaporedij	0,867	80,5 %
Lastnosti aminokislin	0,873	81,2 %
Frekvence razredov	0,866	80,5 %
Frekvence podzaporedij razredov	0,865	80,6 %
Pari	0,873	81,2 %
Pričvrščeni pari	0,863	80,3 %
Celoten ansambel	0,881	83,3 %

Tabela 1. *Klasifikacija na učni množici.*

Celoten ansambel klasifikatorjev smo testirali tudi na testni množici. Pri tem smo meta-klasifikator učili na z nadvzorčenjem uravnoteženi in na izvirni neuravnoteženi učni množici. Uravnotežena učna množica je smiselna, če ne poznamo razmerja razredov v testni množici, neuravnotežena pa, če je razmerje enako kot v učni množici. Rezultate našega ansambla klasifikatorjev smo primerjali z rezultati klasifikatorja, zgrajenega z SVM, ki je uporabljal znakovno jedro. Ta metoda je sodeč po literaturi [1][2] trenutno najboljša znana za prepoznavanje epitopov. Rezultati so prikazani v tabeli 2 in na sliki 1. Vidimo lahko, da se je naš ansambel v vseh pogledih obnesel bolje.

	Uravnoteženo		Neuravnoteženo	
	Naše	Literatura	Naše	Literatura
AUC	0,883	0,868	0,884	0,874
Točnost	83,7 %	82,8 %	85,9 %	83,9 %

Tabela 2. *Klasifikacija na testni množici.*



Slika 1. *Krivulja ROC na testni množici (naši krivulji se skoraj povsem prekrivata).*

5 ZAKLJUČEK

Razvili smo klasifikator za prepoznavanje epitopov, ki prekaša klasifikator, zgrajen z najboljšo znano metodo iz literature. Razlika v klasifikacijski točnosti znaša le 0,9 do 2 odstotni točki, a glede na naravo podatkov to ni malo. Med poizkusi smo namreč ugotovili, da že zelo enostavne metode dosežejo klasifikacijsko točnost nekoliko nad 80 %, vsakršno nadaljnje izboljšanje pa je zelo težavno.

Naš klasifikator je sestavljen iz ansambla osnovnih klasifikatorjev, ki jih – kot pri skladanju klasifikatorjev (stackingu) [5][8] – združuje meta-klasifikator, naučen z linearno regresijo. Naš postopek se od skladanja klasifikatorjev razlikuje po tem, da vsak osnovni klasifikator uporablja ne le svoj algoritem za strojno učenje, ampak tudi svoj atributni vektor. Tako lahko izkoristi različne algoritme za strojno učenje in različne predstavitve podatkov. Predstavili smo tudi temeljit in sistematičen način izbire algoritmov za strojno učenje in nastavljanje parametrov.

Zahvala

Zahvaljujem se prof. Hansu-Jürgenu Thiesnu, ki je priskrbel podatke, ter njemu, prof. Georgu Füllenu in prof. Michaelu Glockerju za nasvete pri delu.

Literatura

[1] EL-Manzalawy, Y., Dobbs, D., in Honavar, V. (2008). Predicting flexible length linear B-cell epitopes. Computational Systems Bioinformatics, str. 121–132.

[2] EL-Manzalawy, Y., Dobbs, D., in Honavar, V. (2008). Predicting linear B-cell epitopes using string kernels. Journal of Molecular Recognition 21 (4), str. 243–255.

[3] Frank, E., Wang., Y., Inglis, S., Holmes, G., in Witten, I. H. (1998). Using model trees for classification. Machine Learning 32, str. 63–76.

[4] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., in Witten, I. H. (2009). The WEKA data mining software: An update. ACM SIGKDD Explorations Newsletter 11(1), str. 10–18.

[5] Seewald, A. (2002). How to make Stacking better and faster while also taking care of an unknown weakness. ICML, 554–561.

[6] van Hulse, J., Khoshgoftaar, T. M., in Napolitano, A. (2007). Experimental perspectives on learning from imbalanced data. ICML, str. 935–942.

[7] Wang, H., in Witten, I. H. (1999) Pace regression. Working paper 99/12.

[8] Wolpert, D. H. (1992). Stacked generalization. Neural Networks 5, str. 241–259.