Univerza v Ljubljani
Fakulteta za računalništvo in informatiko
Fakulteta za matematiko in fiziko

Vito Janko

AKTIVNO UČENJE

diplomsko delo
univeritetni študijski program prve stopnje
računalništvo in matematika

Mentor: prof. dr. Igor Kononenko

Ljubljana, 2012

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

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: 00014/2012
Datum: 12.04.2012

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

Kandidat: **VITO JANKO**

Naslov: **AKTIVNO UČENJE**
ACTIVE LEARNING

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

Naloga je narediti pregled področja aktivnega učenja (active learning). Aktivno učenje je podpodročje strojnega učenja (machine learning), kjer algoritem aktivno izbira učne primere, ki jih potrebuje, da bi bilo njegovo učenje hitrejše in/ali uspešnejše. Kandidat(inja) naj pregleda novejšo literaturo na tem področju in naj povzame najpomembnejše pristope k aktivnemu učenju. Opiše naj osnovne probleme aktivnega učenja in pristope k njihovemu reševanju. Opiše naj tudi trenutno odprte raziskovalne probleme na tem področju s poudarkom na ocenjevanju zanesljivosti predikcij in njihovi razlagi. V praktičnem delu naj preizkusi enega od pristopov na umetno generiranih podatkih oziroma na izbranih realnih domenah.

Mentor:

prof. dr. Igor Kononenko

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan Fakultete za računalništvo in informatiko:

prof. dr. Nikolaj Zimic

Dekan Fakultete za matematiko in fiziko:

akad. prof. dr. Franc Forstnarič

UNIVERZA V LJUBLJANI • Fakulteta za matematiko in fiziko • 6

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Vito Janko z vpisno številko **63090013**, sem avtor diplomskega dela z naslovom:

Aktivno učenje

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Igorja Kononenka,
- 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 12. september Podpis avtorja:

Kazalo

Povzetek

V nasprotju s standardnim nadzorovanim učenjem, kjer učenec dobi naključne učne primere, je ideja aktivnega učenja v tem, da si jih učenec sam iterativno izbere. Izbrani učni primeri so lahko za učenca bolj "informativni"in pokažemo, da jih za enako dober model v primerjavi s standardnim učenjem pogosto potrebujemo precej manj. V tem delu so predstavljeni različni načini, na katere si učenec lahko izbira učne primere ter različne hevristike, na podlagi katerih se odloča o "informativnosti"posameznega učnega primera. Prikazana je medsebojna primerjava uspešnosti različnih pristopov in izboljšava, ki jo prinese aktivno učenje. Predstavljena je teoretična podlaga za aktivno učenje ter njene omejitve; podanih je tudi nekaj kriterijov, ki zagotavljajo uspeh aktivnega učenja. Na koncu predstavljene metode tudi sami preizkusimo.

Abstract

In contrast with standard supervised learning where learner gets random training examples, an active learner can pick training examples itself. Examples picked this way can be more "informative" and we show that for the same model we often need less of them. In this work we present ways in which a learner can choose examples and criteria on how to evaluate their "informativness". We compare different approaches and show that active learning can surpass the standard one. We show some theoretical foundations of active learning and give some criteria that guarantee its success. At the end we present results of our own tests.

Poglavje 1

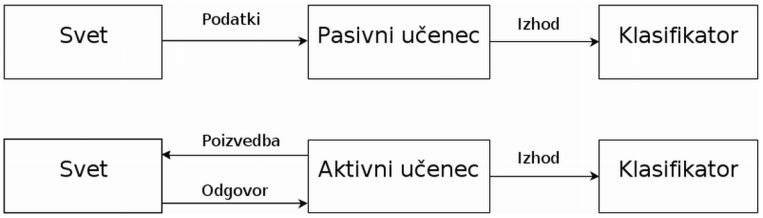
Uvod

Strojno učenje je veja v umetni inteligenci, ki se v splošnem ukvarja z avtomatičnim prepoznavanjem vzorcev v podatkih [15]. Denimo, da imamo bazo podatkov o prodajnih oglasih nepremičnin; o vsaki nepremičnini poznamo velikost in ceno ter vemo, ali se je prodala ali ne. Nato želimo za nek nov primer, ki ni v bazi, na podlagi velikosti in cene vnaprej predvideti, ali se bo prodal ali ne. Takšnemu problemu pravimo klasifikacija, danim spremenljivkam (v našem primeru so to velikost in cena) pravimo atributi, prodaja in neprodaja nepremičnine sta pa razred, v katerega klasificiramo. Navadno se takega problema lotimo tako, da algoritem v danih podatkih prepozna zakonitosti in na njihovi podlagi zgradi klasifikator – funkcijo, ki nadaljnje primere klasificira v enega izmed razredov.

Na takšen algoritem lahko gledamo kot na pasivnega učenca, ki le posluša in se iz danega poskuša čim več naučiti, sam pa ne sodeluje v izbiri učnih primerov. V nasprotju s tem se ta naloga ukvarja z bolj aktivnim učencem, ki sam tudi poskuša izbrati učne primere tako, da bi se iz njih lahko čim več naučil. Primerjamo ga lahko z učencem, ki pri predavanjih sprašuje in tako usmeri učitelja v svoje neznanje in se posledično tako hitreje nauči. Takemu učenju rečemo aktivno učenje.

Formalno, aktivno učenje poteka tako: učenec na podlagi trenutnega klasifikatorja od učitelja zahteva nov, neviden učni primer z danimi vrednostmi

atributov, učitelj ta primer ročno klasificira, nato učenec s pomočjo prejšnjih primerov in novega (opremljenega s pravilnim razredom) zgradi nov klasifikator in iterativno ponavlja postopek. Na takšen način upa, da bo za enako dober model, kot bi ga zgradil pasiven učenec, porabil bistveno manj učnih primerov. Shema je prikazana na sliki 1.1.

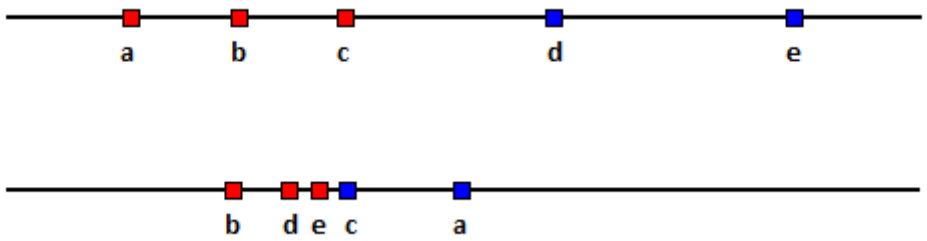


Slika 1.1: Slika prikazuje razliko med shemo učenja pasivnega učenca (zgoraj) in aktivnega učenca (spodaj).

Največja težava je pogosto ravno število učnih primerov, ki so nam na voljo. Več podatkov o problemu imamo, boljše in z večjo zanesljivostjo lahko učenec naredi svojo napoved, vendar je zbiranje večje količine podatkov o domeni navadno zamudno, težko ali drago. Pogosto se zgodi, da je pridobivanje neoznačenih primerov (učni primeri brez podanega razreda) samo po sebi enostavno, drago je le njihovo označevanje. Za primer lahko vzamemo prepoznavanje govora, kjer lahko vedno najdemo večje količine naključnih posnetkov s govorom. Ročno označevanje, katera beseda je bila izrečena v katerem trenutku posnetka, je pa lahko v večji bazi podatkov zelo zamudno. Podobno situacijo dobimo pri klasifikaciji dokumentov, spletnih strani, slik itd. V takšnih situacijah bi želeli označiti samo neko podmnožico vseh učnih primerov, ki jih imamo na voljo. V standardnem (pasivnem) učenju navadno izberemo popolnoma naključno podmnožico, na aktivno učenje pa lahko gledamo kot na nadgradnjo tega koncepta in pustimo, da učenec sam izbere potrebne učne primere in zgradi dober model z manj potrebnimi oznakami.

1.1 Motivacija

Naslednji enostavni primer predstavi, kako lahko s pametno izbiro učnih primerov dosežemo vidne izboljšave v izdelavi modela. Poenostavimo primer iz uvoda in denimo, da na prodajo nepremičnine vpliva le njena cena – vse neprimičnine cenejše od neke vrednosti P se bodo prodale, vse dražje pa ne. Naloga učenca je torej le na podlagi podatkov določiti neznan prag P.



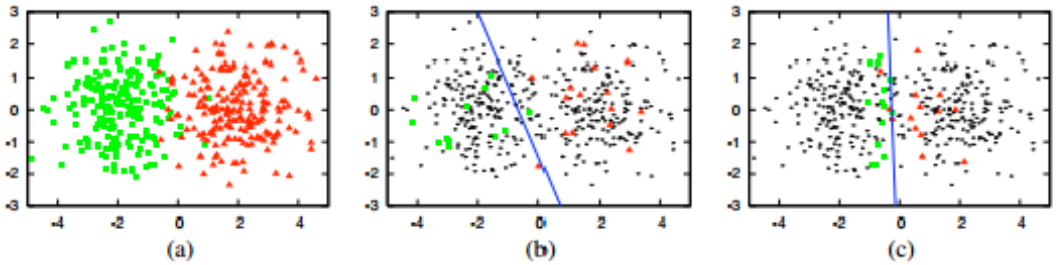
Slika 1.2: Označeni učni primeri v primeru pasivenega (zgoraj) in aktivnega učenja (spodaj).

Na sliki 1.2 zgoraj je os, ki predstavlja možne cene, primeri označeni z zeleno so klasificirani kot prodani, rdeči kot neprodani. Območje na premici med primeroma c in d predstavlja možne kandidate za dejansko vrednost P, med katerimi bo učenec izbral enega za izdelavo svojega klasifikatorja. Manjše kot je to območje, večje zaupanje lahko imamo, da tako izdelan klasifikator dobro opisuje dano domeno. Primeri na zgornji skici so bili izbrani naključno in lahko se pokaže, da je pri taki izbiri primerov pričakovana velikost območja $O(\frac{l}{n})$, kjer je os velikosti l in je n število učnih primerov.

Iz podobnih problemov vemo, da bi bilo dan problem najboljše reševati z metodo bisekcije. Naslednji učni primer, ki ga zahtevamo, ni naključen ampak izbran tako, da prepolovi možno območje praga P. Gre torej za aktivnega učenca, ki izbira naslednji učni primer tako, da se kar se da veliko nauči.

Na sliki 1.2 spodaj vidimo stanje po aktivnem učenju. Učni primeri so

poimenovani po vrstnem redu, po kateremu so bili izbrani. Vidimo da je območje, kjer lahko leži prag P, bistveno manjše (v splošnem velikosti $\frac{l}{2^n}$), kljub uporabi istega števila učnih primerov. Ta sicer enostavni primer nam lahko služi kot motivacija za izdelavo in uporabo metod aktivnega učenja tudi na težavnejših domenah z bolj kompleksnimi klasifikatorji.



Slika 1.3: Na sliki *a* so vsi primeri, ki jih lahko označimo, vzorčeni iz dveh Gaussovih porazdelitev. Naloga klasifikatorja je ločiti razreda z linerno premico. Na sliki *b* je premica, dobljena iz učenja na naključnih učnih primerih, na sliki *c* premica, dobljena iz primerov, ki jih je učenec iterativno izbral. Vidimo, da je meja na tretji sliki bolj ustrezna kot ta na drugi, kar je posledica boljših učnih primerov – primeri na levem in desnem robu slike ne prinesejo veliko informacije klasifikatorju. Slika je vzeta iz [23].

Slika 1.3 je dodaten vizualni zgled učinkovitosti metod aktivnega učenja na kompleksnejšem vendar podobnem problemu.

1.2 Scenariji

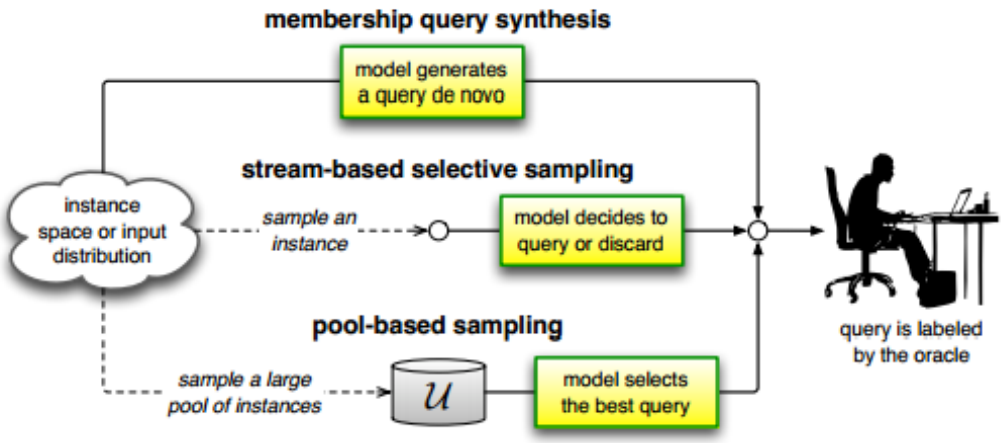
Glavna značilnost aktivnega učenja je izražanje želje po naslednjem učnem primeru – poizvedba (*query*). V tem razdelku prikažemo tri načine, na katere dovolimo učencu tvoriti poizvedbo - poimenujemo jih sestavljene poizvedbe, tokovne poizvedbe ter izbirne poizvedbe [23]. Izbrani tip poizvedb je pogosto določen s samimi okoliščinami učnega problema.

Prvi raziskovani tip poizvedb so bile sestavljene poizvedbe (*Membership query synthesis*). Pri sestavljenih poizvedbah damo učnemu algoritmu povsem proste roke, poizvedbo lahko sestavi "*de novo*"iz katerekoli veljavne kombinacije atributov. Motivacijski primer z bisekcijo spada pod to kategorijo, saj si je učenec izbral poljubno točko na osi in zahteval oznako. Nepričakovana težava nastopi, kadar niti učitelj ne zna klasificirati takega umetno sestavljenega primera [19]. Vzemimo za primer razpoznavanje črk v pisavi. Če učenec vpraša, kateri črki je najbolj podobna neka umetno sestavljena krivulja, obstaja verjetnost, da niti učitelj ne pozna pravega odgovora. Na ta problem pogosto naletimo pri tej vrsti poizvedb, saj učenec pogosto sprašuje ravno zelo robne primere, ki so najtežji za klasifikacijo. Omenimo še, da z umetno generacijo učnih primerov le ti niso usklajeni z dejansko porazdelitvijo učnih primerov v domeni in tako lahko tvorijo slab vzorec za celotno domeno. To težavo ponovno omenimo v kasnejšem poglavju. Ta tip poizvedb se je izkazal za primernega in uporabenega v primeru, da moramo za označevanje opraviti eksperiment in učenec določi okoliščine, pod katerimi se bo izvedel.

Tokovne poizvedbe (*stream-based selective sampling*) so v primerjavi s prejšnjimi vidno preprostejše. Učencu ponudimo učni primer, ta ga sprejme ali zavrne. Če je primer sprejet, ga označimo, v nasprotnem mu ponudimo novega. Tako učenec dobiva "tok"učnih primerov in iz njega vzame samo tiste, ki se mu zdijo primerni. Za uporabnost tega pristopa je seveda treba predpostaviti, da je pridobivanje neoznačenega primera zastonj oziroma vsaj relativno poceni. Učni primeri so vzeti iz dejanske porazdelitve le teh v domeni in so zato bolj smiselni za označevalca, kot so bili v primeru sestavljenih poizvedb.

Pogosto imamo vnaprej dane vse možne učne primere, ki jih lahko označimo in to množico ponudimo učnemu algoritmu za izbor. Temu rečemo izbirne poizvedbe (*pool-based sampling*); v vsaki iteraciji učenec izbere iz množice en primer, učitelj ga označi, nato pa s novim pridobljenim znanjem izbere iz preostale množice nov primer itd. Tudi tukaj moramo predpostaviti nizko

ceno pridobivanja neoznačenih primerov. Z vidika računske kompleksnosti je ta način težji od prejšnjega, saj je prej imel samo en primer, za katerega se je odločal, tukaj pa mora analizirati vse dane primere in izmed njih najti najboljšega.



Slika 1.4: Na sliki so omenjeni tri možni scenariji. Slika je vzeta iz [23].

Poglavje 2

Izbira učnega primera

Kadar se odločimo dani metodi strojnega učenja dodati aktivno komponento, je najpomembnejše vprašanje, na podlagi kakšnega kriterija izbrati naslednji učni primer. Ta razdelek opisuje nekatere pogosteje uporabljene pristope k problemu. Odgovor na vprašanje, kateri izmed v nadaljevanju omenjenih pristopov je najboljši, je zelo odvisen od same metode učenja in učnega problema. Zelo na splošno rečeno moramo določiti funkcijo $f(x)$, ki jo lahko interpretiramo kot izboljšavo trenutnega klasifikatorja, če znanim učnim primerom dodamo nov učni primer x. V primeru sestavljene poizvedbe moramo poiskati $\max f(x)$ (in tako najboljši primer), v primeru tokovne poizvedbe gledamo ali je $f(x) > t$ (kjer je t nek prag, nad katerim so ustrezni primeri) in nazadnje v primeru izbirne poizvedbe iščemo $\max(f(x_1), f(x_2), \ldots . f(x_n))$.

2.1 Pristop najmanjše zanesljivosti

Mnogi klasifikatorji poleg same klasifikacije znajo podati tudi svoje prepričanje o klasifikaciji, to je, s kakšno verjetnostjo je ta klasifikacija pravilna. Naravno si je želeti, da bo klasifikator pri vsaki klasifikaciji imel močno prepričanje. Prepričanje ranga okoli 0.5 (če imamo samo dva razreda) po drugi strani pove, da o tem primeru ne ve dovolj in je zanesljivost take napovedi blizu zanesljivosti ugibanja. Kadar je zanesljivost napovedi znana, jo lahko

uporabimo kot izhodišče za našo aktivno komponento. Za poizvedbo zahtevamo tak primer, kjer je zanesljivost napovedi kar se da nizka, saj bo tako odgovor (označen razred) prinesel koristno vsebino za težji/neraziskani del domene [14]. V nasprotju, če zahtevamo primer za katerega že imamo visoko zanesljivost klasifikacije, nam odgovor verjetno ne bo dal veliko koristne informacije, saj bo le potrdil, kar je učenec že vedel.

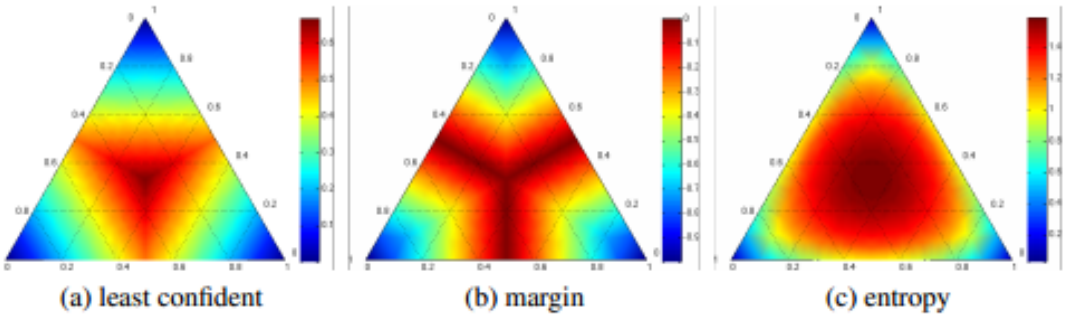
Poskusimo najti dober kriterij za motivacijski primer iz uvoda, če izhajamo iz prej opisanega izhodišča. Recimo, da je A najdražja nepremičnina, ki se je prodala, ter B najcenejša, ki se ni. Območje (A, B) torej predstavlja množico možnih izbir za prag P. Če moramo klasificirati primer, ki je le malo odmaknjen od A (glede na ceno), potem smo relativno prepričani, da se je tudi ta nepremičnina prodala. Če pa po drugi strani dobimo primer s ceno $\frac{(A+B)}{2}$, potem o tem primeru težko kaj povemo. Hiter razmislek pokaže, da je to ravno primer, za katerega smo najmanj prepričani in zato ustrezna izbira za naslednjo poizvedbo. Tako smo s pristopom najmanjše zanesljivosti dobili enak algoritem kot bi ga z metodo bisekcije. Dobro je poudariti, da smo lahko ocenili učni primer z namanjšo zanesljivostjo kjub temu, da samih natančnih verjetnosti nismo imeli.

V primeru, da ima naš klasifikacijski problem več kot 2 možna razreda, ima prejšnji pristop najnižje zanesljivosti pomankljivost: da nam informacijo prepričanosti samo za najbolj verjeten razred tega učnega primera. Imejmo dva učna primera (x_1, x_2), ki imata tri možne razrede (a, b, c). Podane so verjetnosti za posamezen razred, kot jih poda klasifikator: $x_1 - (a : 0.5, b : 0.49, c : 0.01)$, $x_2 - (a : 0.46, b : 0.22, c : 0.22)$. Učenec s pristopom najnižje zanesljivosti bi naredil poizvedbo s učnim primerom x_2. Opazimo lahko, da je za primer x_1 precej neodločen med razredi a in b, primer x_2 bi pa gotovo klasificiral v razred a; iz tega vidika je primer x_1 težji in bolj primeren za naslednjo poizvedbo. Tako smo dobili pristop najmanjšega roba, kjer za naslednjo poizvedbo izberemo primer z najmanjšo razliko med vodilnima razredoma (razreda s prvo in drugo največjo verjetnostjo). Tretji možni pristop je meriti entropijo razredov [27]. Gre za mero iz teorije informacije,

ki meri čistost razredov in se uporablja tudi v mnogih drugih vejah strojnega učenja.

$$f(x) = -\sum_i P_\theta(y_i|x) \log_2 P_\theta(y_i|x)$$

Kjer P_θ v zgornji enačbi predstavlja trenutno oceno porazdelitve razredov in y_i predstavljajo možne razrede učnega prmera x.



Slika 2.1: Na sliki imamo primer s tremi razredi. Od leve proti desni so predstavljeni pristopi najmanjše zanesljivosti, pristop najmanjšega roba in pristop z entropijo razredov. Rdeča barva predstavlja območja, ki bi jih ustrezen pristop najraje izbral. Slika vzeta iz [23].

2.2 Velikost prostora konsistentnih hipotez

Pri motivacijskem primeru smo poudarjali velikost intervala možnih pragov P. To velikost v splošnem imenujemo velikost prostora konsistentnih hipotez[1]. Prostor konsistentnih hipotez (*v*ersion space) je množica vseh možnih hipotez, ki ustrezajo danim primerom (vse znane primere pravilno klasificirajo). Učenec za klasifikacijo uporabi eno izmed njih; manjša množica hipotez seveda pomeni večjo verjetnost, da bo učenec izbral pravo izmed njih. Prav tako lahko pogosto upamo, da bo v primeru manjše množice razlika med

[1]Izrazi hipoteza, klasifikator in model predstavljajo isti pojem

hipotezami manjša. Tako bo tudi v primeru, da učenec izbere napačno hipotezo, ta bolj podobna pravi. Takšna definicija prostora konsistentnih hipotez je smiselna le, če je dana domena ločljiva (*separable*), kar pomeni da obstaja taka hipoteza, ki pravilno klasificira vse učne primere. V realnih učnih problemih, ki vsebujejo šumne učne primere, je prostor konsistentnih hipotez navadno prazen.

Pri tem pristopu aktivni učenec izbere učni primer, ki kar se da zmanjša velikost prostora hipotez. Voditi natančno velikost tega prostora je navadno iz računskega vidika pretežko, prav tako je težko najti najboljši primer, zato se za oboje navadno uporabijo približki. Uspešno uporabo te metode lahko najdemo v [28] in je nakratko opisana v dodatku. Omenimo še, da so tudi drugi pristopi opisani v tem poglavju v resnici dobre hevristike za zmanjševanje prostora konsistentnih hipotez, kar bo utemeljeno v poglavju s teoretičnimi evalvacijami.

Ponovno se vrnimo k izhodiščnemu primeru s trenutnim prostorom hipotez (A, B). Če izberemo x na tem intervalu, ne moremo vnaprej vedeti, kakšen bo nov prostor konsistentnih hipotez (označimo ga z V), saj ne vemo pravega razreda tega primera: če učitelj označi primer x kot 'prodan', je nov prostor (x, B), v nasprotnem pa (A, x). Odločimo se, da želimo izbrati tak x, da bo tudi v najslabšem primeru prostor hipotez karseda majhen - računamo torej $\min\max(velikost(V))$. Ni težko zaključiti da je iskani x ponovno $\frac{(A+B)}{2}$. Računanje izrazov tipa min max je pri aktivnem učenju pogosto, saj želimo narediti dobro odločitev kljub temu, da razreda ne poznamo.

2.3 Komisijski izbor

Naslednji pristop poimenujemo komisijski izbor (*Q*uery by comitee) [26]. Namesto da z danimi podatki tvorimo en klasifikator, kot je v navadi, jih učenec na različne načine naredi več. Nato poskusi klasificirati neoznačeni učni primer z vsakim izmed njih. Če vsi klasificirajo primer v isti razred, je ta primer nezanimiv, nasprotno, kadar se klasifikatorji med seboj ne strinjajo, ta primer

izberemo in označimo. Nato z novim znanjem ponovno generiramo nove klasifikatorje. Za implementacijo tega pristopa potrebujemo način, kako narediti množico klasifikatorjev ter se odločiti za mero nestrinjanja. Klasifikatorji so navadno istega tipa (naivni Bayes, odločitvena drevesa...) le z drugačnimi parametri. Število potrebnih klasifikatorjev je odvisno od danega problema, vendar lahko ta pristop uspešno uporabimo tudi s samo dvemi ali tremi [23].

V primeru ločljivega učnega problema lahko različne klasifikatorje vzorčimo iz prostora konsistentih hipotez in pri tem upoštevamo njihovo porazdelitev, če je le ta poznana. Učni primeri, za katere se ti klasifikatorji ne strinjajo, predstavljajo še neraziskani del domene.

Za pridobivanje več klasifikatorjev si lahko pomagamo z metodami *bagging* [4] in *boosting* [11]. Bagging je metoda, kjer iz originalne množice označenih primerov zaporedoma naključno izbiramo primere in jih damo v novo učno množico. Dovolimo tudi večkratno izbiro istega učnega primera. Nato klasifikator naučimo na novi učni množici. Postopek večkrat ponovimo in tako dobimo klasifikatorje, vse naučene na nekoliko drugačni učni množici. *Boosting* je metodi bagging podobna z vidika, da klasifikatorje pridobivamo s spreminjanjem učne množice. V vsaki iteraciji iz učne množice generiramo klasifikator; težavnejše primere – take, ki bi jih klasifikator napačno klasificiral, dodatno utežimo in nato iteracijo ponovimo na uteženi množici. V praksi lahko primer utežimo kot bolj pomemben tako, da v učno množico postavimo več njegovih kopij.

Drugačen način reševanja tega problema je metoda z različnimi pogledi (*views*) [20]. Predpostavimo, da lahko atribute, ki jih imamo na voljo, razbijemo na disjunktne podmnožice, pri čemer je vsaka od njih dovolj, da lahko iz nje generiramo dober klasifikator. Vsaka podmnožica predstavlja drugačen pogled na dan problem. Na primer: stran na internetu lahko klasificiramo na podlagi njene vsebine ali na podlagi povezav, ki kažejo nanjo. Različne klasifikatorje pridobilmo tako, da se vsak uči le iz atributov, ki pripadajo danemu pogledu. Tak pristop je seveda mogoč le, če naravna delitev na poglede obstaja, oziroma če jo drugače uspemo najti v podatkih.

Pri tem pristopu aktivnega učenja za poizvedbo izberemo učni primer, pri katerem pride do spora – klasifikatorji se o njem ne strinjajo, toda kaj narediti kadar, je teh sporov več? Potrebujemo način, da določimo „najbolj sporen“ učni primer, mero spora. V primeru binarne klasifikacije lahko za mero enostavno vzamemo razliko pozitivnih in negativnih glasov klasifikatorjev. Za posplošitev mere na več razredov je v literaturi uporabljenih veliko kriterijev, omenimo nekatere pomembnejše [20] med njimi.

Mera roba (*Margin-based disagreement*): spor kvantificiramo kot razliko med prvo in drugo največjo verjetnostjo, s katero klasifikatorji določijo različne razrede. Primer: klasifikator 1 klasificira učni primer v razred A z verjetnostjo 0.7, klasifikator 2 v razred B z 0.5 in klasifikator 3 v razred C z 0.8. Rob je v tem primeru $0.8 - 0.7 = 0.1$. Največji spor se torej zgodi, kadar je rob najmanjši, torej kadar sta dva klasifikatorja močno prepričana o drugačnem razredu. Ta mera se je v empiričnih poskusih [16] izkazala za najboljšo.

Mera negotovosti (*Uncertanity sampling-based disagreement*): izberemo učni primer z najnižjo verjetnostjo razreda, to je največja verjetnost, s katero je bil ta primer klasificiran. Izberemo torej učni primer, ki ga noben klasifikator ne zna dobro klasificirati.

Mera entropije (*Entropy-based disagreement*): entropija spremenljivke X je definirana kot

$$H(X) = -\sum_i p(x_i) \log_2 p(x_i)$$

V tem primeru je $p(x_i)$ verjetnosti razreda, kot jo določi i-ti klasifikator.

Körner in Wrobel-ova mera:

$$R = M + 0.5 * \frac{1}{(|C| * P)^3}$$

Ta mera je kombinacija mere roba M in največje vejretnosti razreda P nad množico razredov C.

Kullback-Leibler divergence: v splošnem je *Kullback-Leibler divergence* (v nadaljevanju KL) [17] mera za razliko med porazdelitvima dveh slučajnih

spremennljivk p, q.

$$KL(p||q) = \sum_i p(x_i) \log \frac{p(x_i)}{q(x_i)}$$

kjer so x_i vsi možni dogodki. Visoka mera KL pomeni veliko razliko v porazdelitvah p, q. Spor med klasifikatorji merimo s

$$KLmean = \frac{1}{k} \sum_i KL(p_i(x)||pmean(x)) \tag{2.1}$$

kjer je k število klasifikatorjev, $p_i(x)$ porazdelitev razredov, kot jih določa i-ti klasifikator in $pmean(x)$ je povprečna porazdelitev preko vseh klasifikatorjev.

Entropija razredov:

$$VE(x) = -\frac{1}{\log k} \sum_i \frac{V(l_i, x)}{k} logV(l_i, x)/k \tag{2.2}$$

kjer je k število klasifikatorjev in $V(l_i, x)$ število klasifikatorjev, ki primeru x dajo oznako l_i. Za razliko od ostalih, za to mero spora od klasifikatorja ne potrebujemo gotovosti napovedi.

2.4 Pričakovano zmanjšanje napake

Želimo si, da bi naš klasifikator, posobljen z vrnjeno in označeno poizvedbo, imel v prihodnje kar se da majhno napako [22]. Te napake ne moremo vnaprej vedeti, vendar lahko dobimo oceno z uporabo množice neoznačenih primerov kot reprezentativno podmnožico vseh primerov. Ta pristop lahko torej uporabimo le v primeru izbirnih poizvedb. Zanima nas verjetnost, s katero bo nov klasifikator znal klasificirati ostale primere v neoznačeni množici. Nov klasifikator se spremeni v skladu z oznako, ki nam ni poznana, zato to verjetnost ocenimo z matematičnim upanjem izraza.

$$Error_{0/1} = \sum_i P_\theta(y_i|x)(\sum_u 1 - P_{\theta+(x,y_i)}(y^*|x^{(u)}))$$

kjer je $P_\theta(y_i|x)$ verjetnost i-tega razreda glede na trenutni klasifikator, y^* je razred z največjo verjetnostjo, $P_{\theta+(x,y_i)}(y^*|x)$ verjetnost glede na nov klasifikator, pridobljen iz prejšnje množice učnih primerov in novega para (x, y_i),

$\sum_u$ je vsota preko vseh neoznačenih učnih primerov, ki jih imamo na voljo in $\sum_i$ je vsota preko vseh možnih razredov učnega primera x.

Podobno dobimo, če poskušamo zmanjšati pričakovano entropijo po množici preostalih primerov.

$$Error_{log} = \sum_i P_\theta(y_i|x)(-\sum_u \sum_j P_{\theta+(x,y_i)}(y_j, x) * \log P_{\theta+(x,y_i)}(y_j, x)) \quad (2.3)$$

Težava tega pristopa je v računski zahtevnosti: za izbiro naslednje poizvedbe moramo za vsak možen razred vsake poizvedbe posebej izračunati nov klasifikator in ga nato uporabiti za klasificiranje vsakega preostalega učnega primera.

2.5 Mešani pristop

Metode, kot so izbira učnega primera z najmanjšo zanesljivostjo, prioritizirajo primere, ki so zelo mejni, različni od ostalih v trenutni označeni množici. Taki primeri so lahko drugačni le zaradi šuma v podatkih ali so samo zelo redki v originalni porazdelitvi primerov in tako mogoče niso reprezentativni za ostale učne primere. Tako se lahko učenec uči le na osamelcih (*outlier*), kar ne poveča klasifikacijske točnosti. Metoda najmanjše prihodnje pričakovane napake se temu izogne tako, da ne gleda le na individualni primer, temveč upošteva tudi celotno neoznačeno množico.

To idejo lahko uporabimo tudi tako [24], da računamo

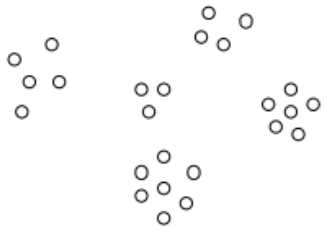
$$f(x) = \phi(x) * (\frac{1}{n} \sum_u sim(x, x^{(u)}))^\beta \quad (2.4)$$

kjer je $\phi(x)$ ocena "informativnosti" za x glede na katerikoli že opisan pristop, $\frac{1}{n} \sum_u sim(x, x^{(u)})$ povprečje podobnosti tega učnega primera z ostalimi v neoznačeni množici in β parameter, ki določa pomembnost člena gostote.

S tem učne primere, ki so bolj reprezentativni, tudi močneje utežimo in jim tako pripišemo večji pomen. Ta metoda združuje navidezno nasprotne interese: poizvedba naj bo kar se da različna od ostalih v označeni množici in kar se da podobna ostalim v neoznačeni množici.

2.6 Struktura

Teoretično gledano obstajata dva razloga, zakaj bi lahko aktivno učenje pomagalo. Prvi je v tem, da bolj učinkovito preizkujemo prostor hipotez in na podlagi tega izbiramo zanimive primere. Ta pristop je uporabljen v vseh do sedaj omenjenih metodah in je od teh dveh bolj raziskan. Drugi pristop je v izkoriščanju strukture podatkov. Vzemimo za primer učne primere na sliki 2.2.



Slika 2.2: Strukturirani podatki. Slika je vzeta iz [6].

Na sliki 2.2 vidimo 5 gruč in morda imajo vsi v isti gruči enako oznako in bi tako lahko naredili poizvedbe le na petih učnih primerih. Tako upanje je sicer preveč optimistično, saj v splošnem lahko da ni v podatkih nobenih očitnih gruč, oziroma le te obstajajo na različnih nivojih. Lahko da so celo gruče nekorelirane z njihovimi oznakami. Področje, ki bi se ukvarjalo z razvojem algoritmov, ki so sposobni izkoristiti strukturo podatkov brez predpostavk o njegovi porazdelitvi, je še relativno nerazvito. Vseeno na hitro opišimo dva algoritma, ki to počneta.

Enostavna shema

Prva enostavna shema [6] poteka tako:

Na podlagi sosednosti sestavi graf iz neoznačene množice učnih primerov
Zahtevaj oznake za nekaj naključnih učnih primerov[2]

[2]Neoznačene primere imamo že na začetku na voljo, zato gre tu za izbirne poizvedbe

while imamo na voljo poizvedbe **do**
    Propagiraj oznake iz označenih primerov na njegove sosede
    Zahtevaj oznako za neraziskan del grafa
end while

Algoritem DH

Naslednji algoritem je po avtorjih poimenovan DH [7]. Vzemimo množico vseh učnih primerov na voljo S. Algoritem bo v naslednji iteraciji na podlagi trenutne grupiranosti izbral gručo, v kateri bo naredil naslednjo poizvedbo. Da se izogne pristranosti vzorčenja (*sampling bias*)[3], bo v izbrani gruči izbral popolnoma naključen učni primer in zanj zahteval oznako.

Ko bo porabil vse poizvedbe, ki so mu na voljo, bo v vsaki gruči vse učne primere označil glede na večinski razred v tej gruči. Za omejitev napake tega procesa je pomembno, da imajo posamezne gruče čim bolj homogeno oznako. V primeru, da je posamezna gruča preveč nehomogena, jo algoritem razcepi na dve manjši gruči. Napako, ki jo nehomogenost povzroči, lahko ocenimo z uporabo binomske distribucije.

Da bi se izognili komplikacijam pri evalvaciji algoritma, je način deljenja gruč na manjše neodvisen od do sedaj videnih oznak; z drugimi besedami: hirarhično deljenje gruč na manjše in manjše je že v naprej določeno pred samim učenjem. Hirarhično grupiranje podatkov lahko pridobimo s katerimkoli za to namenjenim postopkom.

Na koncu se moramo odločiti še za strategijo, na podlagi katere bo algoritem izbral naslednjo gručo za poizvedbo. Dve izmed možnosti so izbiranje naključne gruče in izbiranje najbolj nehomogene gruče.

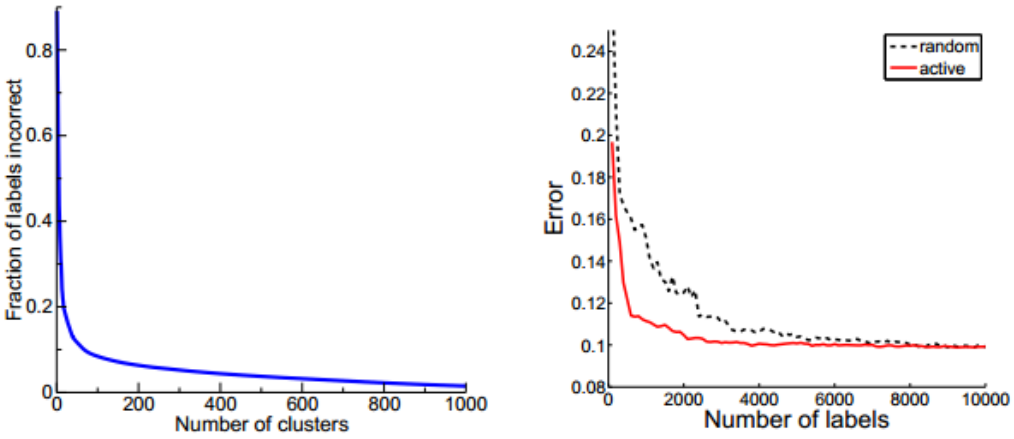
Evalvacija algoritma

Za praktične evalvacije različnih pristopov je v nadaljevanju namenjena celotno poglavje, vendar za ta specifičen algoritem rezultat navedimo kar tukaj. V [6] je avtor algoritem DH preizkusil na učnem problemu prepoznavanja

[3]Pojasnjeno bo v teoretičnem razdelku

ročno napisanih števk. Na začetku je algoritem imel na voljo 10000 neoznačenih slik, ki so bile predstavljene kot vektorji v $\mathbb{R}^{784}$.

Hirarhično drevo gruč bi v tem učnem problemu lahko imelo od potencialno 10 (vsaka števka predstavlja svojo gručo) do 10000 različnih listov. Izkazalo se je, da je bilo 50 listov dovolj za klasifikacijo z napako, manjšo od 12%. Slika 2.3 prikazuje odvisnosti napake od števila poizvedb in števila gruč.



Slika 2.3: Prva slika prikazuje odvisnost napake od števila različnih gruč, druga pa v odvisnosti od opravljenih poizvedb. Na drugi sliki je za primerjavo s črtkano črto prikazan uspeh pasivnega učenca. Slika je vzeta iz [6].

2.7 Ostalo

Glede na učni problem in algoritem učenja lahko seveda izberemo tudi poljubni drugi kriterij, ki dani situaciji ustreza. Če na primer namesto ene same hipoteze učenec vodi verjetnostno porazdelitev čez vse možne hipoteze (tak primer imamo naprimer pri učenju Bayesovih mrež), je na primer smiselno iskati učni primer, ki zmanjša disperzijo take porazdalitve. Če gradimo klasifikator z lokalno optimizacijo: vsak učni primer nekoliko spremeni trenutni

model, lahko izberemo učni primer z največjim gradientom (primer uporabe v [25]). Ta pristop lahko posplošimo tako, da vedno izberemo učni primer, ki najbolj spremeni trenutni učni model. Sprememba modela je seveda odvisna od oznake tega učnega primera, zato izberemo takega, kjer je sprememba v najslabšem primeru največja.

Poglavje 3

Praktične evalvacije

V prejšnjem poglavju smo našteli možne pristope (različne hevristike pri izbiri novega učnega primera) za aktivno učenje; zanima nas, ali bomo z uporabo le teh res kaj pridobili in če jih bomo lahko uporabili v praksi. Odgovor je vsaj deloma pritrdilen, saj se pojavlja vedno več člankov s pozitivnimi rezultati na realnih problemih. Prav tako so te metode začela uporabljati večja podjetja, kot so CiteSeer, Google, IBM, Microsoft, in Siemens [23]. Obstajajo tudi rezultati, ki pričajo nasprotno: na nekaterih učnih problemih je za isto natančnost aktivni učenec potreboval več primerov kot pasivni. Eden izmed njimi je delo [21], v katerem avtor med seboj primerja različne hevristike za izbor naslednjega primera in naključno izbiranje primerov. Rezultati so obnovljeni v naslednji sekciji.

3.1 Primerjava pristopov

Namen tega razdelka je primerjava različnih pristopov, bazirana na večrazrednem klasifikatorju *logistic regression* in izbirnih poizvedbah. Opomnimo, da bi se lahko z drugim klasifikatorjem pristopi drugače obnesli. Pri tem klasifikatorju so verjetnosti razredov učnega primera x modelirane kot:

$$P(y = c|x) = \frac{\exp(w_c x)}{\sum_c \exp(w_c x)}$$

kjer s c označumo možne razrede in so w_c parametri klasifikatorja.

Uporabljeni pristopi so bili sledeči:

- Pristop najmanjše zanesljivosti, kjer se zanesljivost kvantificira po principu entropije
- Pristop najmanjše zanesljivosti, kjer se zanesljivost kvantificira po principu najmanjšega roba. Ta in prejšnji pristop sta izmed vseh najlažje izračunljiva.
- Komisijski izbor, pri čemer različne klasifikatorje pridobimo z metodo *bagging* in spor kvantificiramo z (2.1). (V nadaljevanju po avtorjih označimo z QBBMN)
- Komisijski izbor, pri čemer različne klasifikatorje pridobimo z metodo *bagging* in spor kvantificiramo po principu najmanjšega roba.(V nadaljevanju po avtorjih označimo s QBBAM)
- Pristop zmanjšanja pričakovane entropije (2.3)
- *Variance reducing*
- *Log loss reduction*

Zadnja dva pristopa sta bolj teoretične narave in njuna implementacija je specifična za klasikator *logistic regression*, za več podrobnosti glej [21]. Velja omeniti, da sta izmed vseh naštetih najtežje izračunljiva - celotno učenje z *Log loss reduction* je vzporedno teklo na desetih računalnikih, in kjub temu trajalo 3 tedne.

Poleg algoritmov z aktivnim učenjem za primerjavo v poizkusu nastopa še pasiven učenec, ki izbira naključne primere ter pasiven učenec, ki za izdelavo klasifikatorja uporablja metodo *bagging*.

Poizkus je bil izveden sledeče: učni algoritem je imel na začetku na voljo začetno množico učnih primerov (*seed set*) velikosti 20, nato je iterativno pridobil iz neoznačene množice 10 naključnih učnih primerov, izmed katerih je

glede na hevristiko izbral najboljšega. Postopek se ponovlja, dokler algoritem ne uporabi vseh učnih primerov, ki so mu na voljo. Testiranje je potekalo z 10-kratnim prečnim preverjanjem.

V testiranju so bile uporabljene tako realne kot umetno generirane učne množice. Realne množice so bile poimenovane *Comp2a, Comp2b, LetterDB, NewsGroups, OptDigits, TIMIT, WebKB*; gre za večje množice podatkov (do 20 000 učnih primerov) z do 26 različnimi razredi. Rezultati poizkusa so prikazani na sliki 3.1.

Nekatere izmed učnih množic so bile umetno generirane, da preizkusijo vpliv šuma na metode aktivnega učenja. Prvi tip šuma povečuje klasifikacijsko napako ne glede na velikost množice učnih primerov. V namen testiranja tega šuma sta bili na voljo množici *Art* in *ArtNoisy*. Prva je bila brez šuma in je služila kot osnova za primerjavo, druga je imela nakljućen Gaussov šum. Drug tip šuma je namenjen umetnemu ustvarjanju spornih regij, da bi zavedel hevristike najmanjše zanesljivosti. Množica *ArtConfig* je ustvarjena tako, da atributi učnih primerov izhajajo iz dveh distinktnih regij; Učni primeri iz prve regije so klasificirani z isto verjetnostjo v razreda 1 ali 2, učni primeri iz drugih regij so klasificirani kot v originalnem *Art* primeru, brez šuma v preostalih 18 razredov. Učni primeri iz prve regije so sicer najtežji za klasifikacijo, vendar poizvedovanje le teh nima nobene učne vrednosti.

Da bi ugotovili, ali je uspešnost hevristik odvisna od zgoraj naštetih okoliščin poizkusa, je bil poizkus ponovljen z drugačnimi razmerami. Izkazalo se je, da s spreminjanjem začetne množice, večje množice nakljućnih primerov, ki jih ima algoritem v vsaki iteraciji na izbiro in večjimi *bag sizes* pri metodah QBB relativna uspešnost različnih hevristik ostane enaka.

Nekoliko presenetljivo je, da nobenemu pristopu ni uspelo biti boljši od nakljućnega izbiranja v vseh primerih ter da so se ti pristopi včasih izkazali celo za slabše.

Učni algoritem, ki je uporabljal samo *bagging*, se je odrezal slabo, kar je lahko posledice manjše variance v teh učnih problemih (metoda *bagging* se navadno uporablja v učnih problemih z visoko pričakovano varianco).

Data Set	random	bagging	variance	log loss
Art	NA	-	+	+
ArtNoisy	NA	-	+	+
ArtConf	NA			
Comp2a	NA	-		
Comp2b	NA			
LetterDB	NA	-	+	+
NewsGroups	NA	-	NA	NA
OptDigits	NA		+	+
TIMIT	NA	-		
WebKB	NA	-	NA	NA

	CC	QBB-MN	QBB-AM	entropy	margin
Art	+	+	+	+	+
ArtNoisy		+		-	+
ArtConf				-	-
Comp2a	-				
Comp2b					
LetterDB	+	-	+	-	+
NewsGroups	NA	-	-	-	-
OptDigits	+	+	+	+	+
TIMIT	-		+	-	+
WebKB	NA	+	+	+	+

Slika 3.1: Prikazani so rezulati poizkusa. Znak + označuje statistično signifikante razlike v prid dane metode v primerjavi z naključnim izbiranjem primerov, nasprotno − pomeni, da se je ta metoda odrezala signifikantno slabše. Nekateri pristopi zaradi zapletov z izračunljivostjo niso bili testirani na vseh množicah, taki primeri so označeni z NA. Slika je vzeta iz [21].

Najboljše sta se odrezala *Variance reducing* in *Log loss reduction*, ki sta bila v vseh primerih enaka ali boljša od naključnega izbiranja primerov. Na drugi strani se je najslabše odrezal pristop najmanjše prepričanosti z entropijo, kar avtor navaja kot posledico šuma (učinkovitost tega pristopa je bila obratno proporcionalna s količino šuma v podatkih). Glede na njeno preprostost se je pristop najmanjše zanesljivosti z robom izkazal za kar uspešnega; spodletelo mu je le dveh učnih problemih, eden izmed katerih je bil umetno generiran tako, da temu pristopu škoduje.

3.2 Dodatna primerjava

V tem razdelku navedimo dodatne rezultate primerjave različnih pristopov, tokrat vzete iz [24]. Aktivno učenje je bilo tu uporabljeno na problemu klasifikacije besed v danem nizu v naravnem jeziku. Primer takega problema je glede na kontekst ugotoviti, ali je dana beseda ime kraja ali ime podjetja. Zaradi velikega števila prosto dostopnih dokumentov je pridobivanje neoznačenih učnih primerov poceni in tako aktivno učenje postane zelo atraktivno. Avtor uporablja izbirne poizvedbe in model naredi tako, da le ta maksimizira *log likelihood* trenutnih označenih primerov.

Prvi uporabljeni pristopi so iz družine pristopov najmanjše zanesljivosti. Testiranih je več: navadni, kjer gledamo le verjetnost razreda z največjo zanesljivostjo (označimo z LC), pristop z robom (M), pristop po entropiji razredov (TTE), njegovo normalizirano verzijo glede na dolžino niza (TE), pristop z entropijo, kjer namesto razreda posamezne besede opazujemo vse različne sekvence razredov v nizu (SE) in lažje izračunljiv približek prejšnji (NSE).

Sledi družina pristopov po principu komisijskega izbora. Klasifikatorji so v vseh primerih narejeni po principu *bagging*, razlikujejo se v kvantificiranju nestrinjanja. Testirana sta bila kriterija entropija razredov (TVE) in *KL-divergence* (TKL) ter njuni normalizirani verziji glede na dolžino niza (VE, KL). Vključena sta tudi kriterija, ki merita nestrinjanje na nivoju celotnega niza in ne na nivoju besed (SVE in SKL).

Nazadnje so testirani še trije pristopi: pričakovana dolžina gradienta (LGL), mešani pristop (2.4), pri katerem je uporabljen koeficient $\beta = 1$ (ID) in *fisher information* (FIR), ki tukaj ni opisan (bralec si lahko pogleda [24]). Za primerjavo sta dodana še dva učenca; prvi uporablja naključne učne primere, drugi pa poizveduje najdaljše (glede na število besed) učne primere.

Učenje se v vseh primerih začne z začetno množico velikosti 5 in nato dovolimo 150 poizvedb. Rezultati so bili povprečeni na petkratnemu prečnemu preverjanju.

Kakor pri prejšnji primerjavi, tudi tukaj ni bilo čistega zmagovalca. Naj-

boljše se je odrezal pristop ID, saj v nobenem primeru ne deluje slabo in ima najvišje povprečje. Naslednji močen kandidat je SVE, ki konsistentno zaseda mesto med prvimi tremi. Izmed vseh pristopov najmanjše zanesljivosti pa se je najbolj izkazal LC. Pristop pričakovanega najdaljšega gradienta in *fischer information* sta bila po eni strani računsko najbolj zahtevna (potrebovala sta do 30 min na poizvedbo v daljših besedilih), po drugi ne dosegata rezultatov, da bi utemeljila njuno zamudnost. Pripomnimo, da je bila večina metod v povprečju boljša od naključnega izbiranja (izjema so bili pristopi TE, VE in KL), vendar so redki bili boljši na vseh učnih množicah.

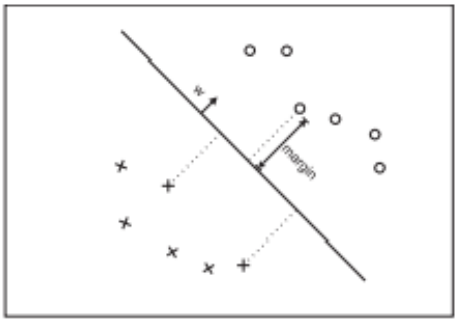
3.3 Testiranje z metodo SVM

V tem razdelku podamo rezulate za aktivno učenje z uporabo metodo podpornih vektorjev (SVM – Support Vector Machine) [29], povzete po [28]. Ideja te metode je med seboj ločiti učne primere iz dveh razredov s hiperravnino. V primeru dveh dimenzij so učni primeri ločeni s premico (glej sliko 3.2).

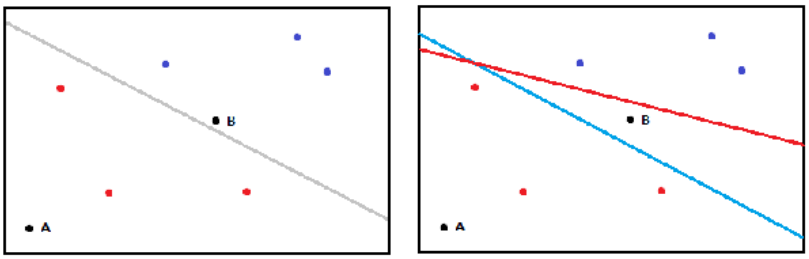
Za delovanje metode SVM obstaja nazorna vizualna interpretacija, prav tako si lahko vizualno predstavljamo izboljšave učenja, ki jih prinese aktivna komponenta (glej sliko 3.3). V testiranju uporabljeni pristopi aktivnega učenja so razviti posebej za metodo SVM, skico njihove izpeljave in razlage lahko najdemo v dodatku.

3.3.1 Klasifikacija besedila

V [28] avtor preizkusi delovanje aktivne komponente na problemu klasfikacije besedila. Vsako besedilo ima temo, ki jo klasifikator poskuša poiskati. Uporabljena je bila realna domena: *Reuters-21578*. Vsako posamezno besedilo je bilo predobdelano in predstavljeno kot množica atributov, kjer je vsaka različna beseda predstavljala svoj atribut. Zelo pogoste besede so bile izpuščene, besede z istim korenom pa združene v eno. Vrednost i-tega atributa w_i je bila: $TF(w_i) * IDF(w_i)$, kjer je $TF(w_i)$ število pojavitev te besede



Slika 3.2: Učni primeri ločeni s premico tako, da je rob (*margin*) kar se da velik. Slika je vzeta iz SVM [28].



Slika 3.3: Na levi sliki imamo tri primere iz enega razreda in tri iz drugega (označeni s rdečo in modro barvo), ter sivo premico, ki označuje trenutno hipotezo z največjim robom. Na voljo imamo še dva neoznačena primera A in B. Primer A skoraj gotovo pripada 'rdečemu' razredu (še posebej če smo predpostavili, da linearna meja obstaja) in oznaka tega primera v 'rdeče' nič ne spremeni trenutne meje. Po drugi strani bi označen primer B bistveno spremenil obstoječo mejo. Desna slika prikazuje meji po oznaki primera B z 'modro' in mejo po oznaki primera B z 'rdečo'. Relativno velika razlika v meji govori o veliki informativnosti primera B. Vidimo, kako se s pravilno izbiro primerov model bolj izboljša, kot če uporabimo naključno izbrano podmnožico neoznačene množice.

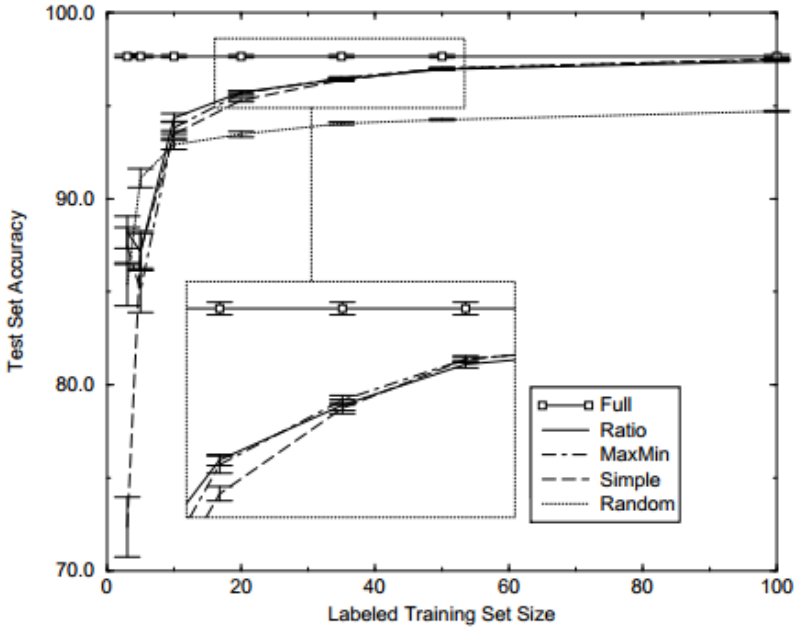
v besedilu, $IDF(w_i) = \log \frac{n}{N_i}$ in N_i število dokumentov, kjer se ta beseda pojavi. Intuitivno nam visoka vrednost atributa pove, da je v tem besedilu ta beseda pogosta in v drugih redka.

V tej domeni je bilo 3300 besedil, besedila so imela 10 različnih tem, vektor atributov je bil približno 10000 dimenzionalen. Testiranje je potekalo tako: izmed vseh učnih primerov jih je 1000 bilo naključno izbranih za neoznačeno učno množico. Učenec je v začetku dobil 2 označena primera, nato je po omejenem številu poizvedb moral vrniti klasifikator, ki pove ali, določeno besedilo pripada izbrani temi ali ne - gre za binarno klasifikacijo. Postopek je bil tridesetkrat ponovljen za vsako temo in dobljeni rezulat je povprečje vseh testiranj. Učenec je poizkusil tri pristope: Enostavni rob, MaxMin rob in Razmerni rob (pristopi so opisani v dodatku, na tem mestu samo omenimo, da sta zadnja dva precej bolj računsko zahtevna od prvega). Poleg tega je rezultate primerjal z naključnim izbiranjem učnih primerov.

Rezultate lahko vidimo na sliki 3.4. Razlika med različnimi pristopi je bila majhna, z Enostavnim robom malo v ozadju. Po drugi strani je izboljšava, ki jo prinese aktivna komponenta, tukaj pri vseh temah očitna. Lahko opazimo, da se aktivni učenec že po stotih poizvedbah nauči skoraj toliko, kot če bi na voljo imel vseh 1000 označenih učnih primerov.

Rezultati preizkusa so bili ocenjeni tudi po meri priklica in preciznosti in tudi v tem primeru je bil aktivni učenec ocenjen precej boljše kot pasivni.

Aktivni učenec se je v primerjavi s pasivnim še posebno izkazal pri klasificiranju redkih tem. Opaziti je bilo možno, da je med poizvedbami približno polovica takih besedil, ki so relevantna glede na zahtevano temo in polovica takih, ki niso. Uravnoteženo izbiranje učnih primerov je torej prispevalo h gradnji boljšega klasifikatorja. Da bi preizkusil, ali je ravno to uravnoteženo izbiranje poglavitnega pomena za uspeh metod aktivnega učenja, je avtor na domeni testiral učenca, ki sicer izbira naključne primere, vendar jih pol izbere izmed relevantnih besedil in pol med nerelevantnimi. Izkazalo se je, da je tak učenec sicer boljši od popolnoma naključnega, vendar še vedno vidno slabši od aktivnega.



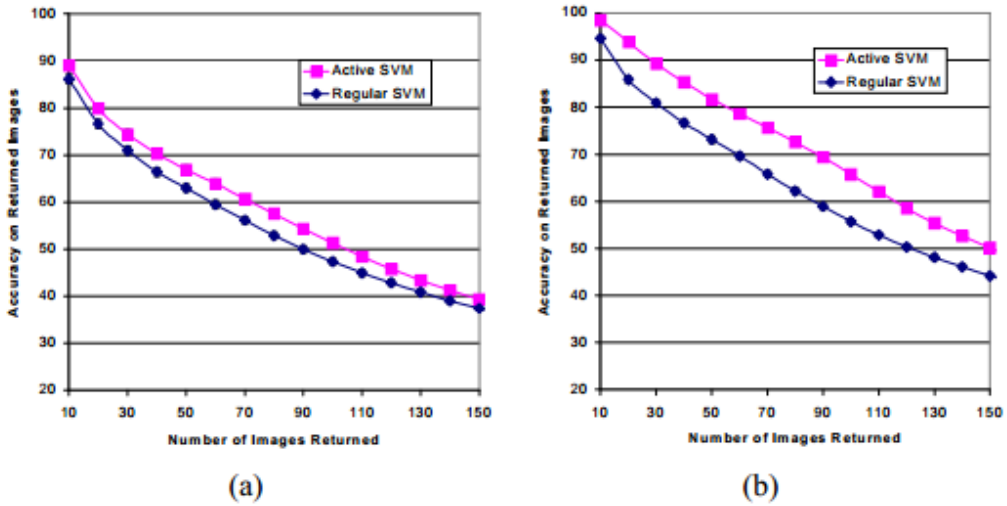
Slika 3.4: Klasifikacijska točnost treh pristopov glede na število zahtevanih poizvedb. Slika je vzeta iz [28].

Iskanje slik

Omenimo še eno testiranje istega avtorja, tokrat na domeni določanja relevantnosti slik. Predstavljamo si uporabnika, ki v dani podatkovni bazi slik želi najti zanj pomembne. To lahko z aktivnim učenjem dosežemo tako: uporabniku ponudimo nekaj slik, ta med njimi izbere tiste, ki mu ustrezajo, nato mu ponudimo novo množico slik in ponovimo. Slike v množici izberemo tako, da se bo učni algoritem pomembnosti slik čim prej naučil - torej z metodami aktivnega učenja. Po nekaj iteracijah uporabniku vrnemo vse slike, za katere je učni algoritem močno prepričan, da so za uporabnika pomembne. Primer je zanimiv, ker je tukaj označevalec nek uporabnik aplikacije in imamo zaradi njegove omejene potrpežljivosti tudi omejeno (in majhno) množico označenih učnih primerov.

Brez omenjanja vseh podrobnosti testiranja pokažimo rezult na sliki 3.5.

Vidimo, da je tudi za to nalogo aktivno učenje uspešnejše od običajnega.



Slika 3.5: Po učenju je učni algoritem vrnil najboljših nekaj slik glede na svoj trenutni klasifikator. Sliki prikazujeta klasifikacijsko točnost vrnjenih slik glede na njihovo število. Slika *a* prikazuje stanje po treh iteracijah (v vsaki iteraciji je bilo 20 slik), slika *b* po petih. Slika je vzeta iz [28].

Poglavje 4

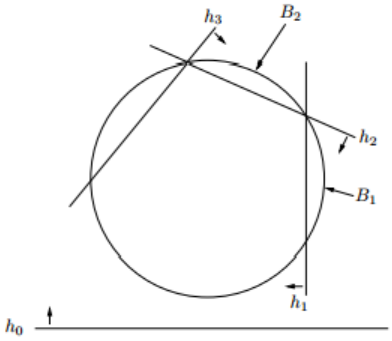
Teoretične evalvacije

To poglavje je namenjeno teoretičnem obravnavanju metod aktivnega učenja. To področje je trenutno zelo dejavno in hitro napreduje, vendar je večina analize narejene na zelo močnih predpostavkah. Navadno predpostavljamo, da je problem klasifikacije binaren (imamo samo dva klasifikacijska razreda) in ločljiv.

Z uporabo pasivnega učenja lahko dosežemo klasifikacijsko točnost boljšo od ϵ, če imamo približno $\frac{d}{\epsilon}$ učnih primerov [3], kjer je d dimenzija VC učnega problema. Več o njej si lahko bralec prebere v [30]. Videli smo, da je bilo v motivacijskem primeru možno zasnovati algoritem tako, da je bilo število potrebnih primerov eksponentno manj – $O(\log \frac{1}{\epsilon})$. Naravno vprašanje je, ali je mogoče ta fenomen generalizirati na druge, kompleksnejše učne probleme. Na žalost lahko hitro najdemo naravni učni problem (opisan spodaj), pri katerem si z aktivnim učenjem ne moremo vedno pomagati.

4.1 Primer linearnih klasifikatorjev v $\mathbb{R}^2$

Za prostor hipotez vzemimo linearne klasifikatorje v $\mathbb{R}^2$, učni primeri naj ležijo na enotski krožnici. Krožnico razdelimo na odseke tako, da je verjetnost, da učni primer leži na vsakem izmed njih, enaka ϵ. Hipotezo, ki loči i-ti odsek B_i od ostalih, poimenujemo h_i. Učni algoritem mora na podlagi



Slika 4.1: Na sliki je predstavljen učni problem z označenimi odseki in hipotezami. Slika je vzeta iz [9].

označenih primerov biti zmožen ločiti med hipotezami h_i , za vsak i, ter trivialno hipotezo h_0, ki vse primere klasificira kot negativne (glej sliko 4.1). V nasprotnem bo napaka algoritma za te hipoteze večja ali enaka ϵ.

Da bi te hipoteze lahko med seboj razlikovali, moramo označiti vsaj en primer na vsakem odseku, torej za to potrebujemo $O(\frac{1}{\epsilon})$ označenih učnih primerov. To ni nič boljše, kot ponuja pasivno učenje.

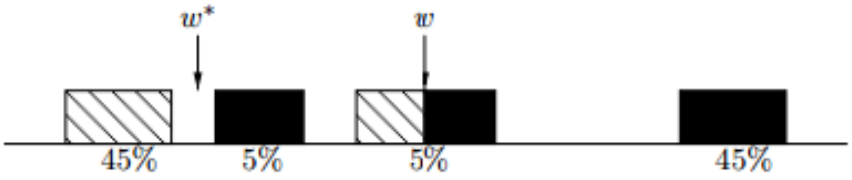
Problematične so v tem problemu bile le hipoteze z zelo neenakomerno porazdelitvijo pozitivnih in negativnih učnih primerov. Recimo, da smo našli na krožnici en negativen in en pozitiven primer, potem lahko ciljno hipotezo poiščemo z bisekcijo na eni in drugi strani. Torej lahko učni problem razbijemo na potencialno počasno iskanje pozitivnega in negativnega primera in nadaljnjo hitro bisekcijo.

Aktivnega učenja torej ne moremo v splošnem uporabiti ne glede na učni problem. V tem razdelku predstavimo nekatere možne kriterije za učne probleme, ki nam zagotavljajo, da si pri njih lahko pomagamo z aktivnim učenjem in dosežemo do eksponentno manjše število potrebnih učnih problemov kot pri pasivnem učenju.

4.2 Premisleki proti aktivnemu učenju

Navajamo še dva teoretična premisleka proti aktivnemu učenju. V svojem delu Eisenberg in Rivest [10] trdita, da je za veliko naravnih učnih problemov aktivno učenje precej neuporabno, saj učenec, ki vidi le malo označenih problemov, ne more biti občutljiv na majhne spremembe ciljne hipoteze in na porazdelitev učnih primerov. Pokazala sta, da je mogoče to dvoje rahlo spreminjati tako, da se hipoteza učenca ne bi spremenila, klasifikacijska napaka pa bi se občutno povečala.

Velika težava metod aktivnega učenja je tudi v reprezentativnosti označenega vzorca[6]. Naključna in dovolj velika množica učnih primerov je dober vzorec za pravo porazdelitev učnih primerov. Težava nastane, saj v aktivnem učenju iz te množice po nekem kriteriju izberemo in označimo manjšo podmnožico učnih primerov. Porazdelitev teh primerov je lahko zelo oddaljena od prvotne porazdelitve in zato ta vzorec ni reprezentativen. Klasifikator, z dobro točnostjo na tem vzorcu, ni nujno dober za učne primere pridobljene iz prvotne distribucije. Tej težavi v splošnem rečemo pristranost vzorčenja (*sampling bias*). Težava je dodatno podkrepljena, če poizvedbe delamo na umetno ustvarjenih učnih primerih.



Slika 4.2: Učni problem je najti prag, ki bo ločil bele primere od črnih. Učni primeri naj bodo porazdeljeni v označenih gručah, s pripadajočimi verjetnostmi pripisanimi pod njimi. Idealen prag (z najmanjšo pričakovano napako) je označen z w^*. Slika je vzeta iz [6].

Za ilustracijo zgornjega problema vzemimo učni problem, prikazan na

sliki 4.2. Recimo da učni algoritem začne z nekaj učnimi primeri iz skrajno levega in nekaj iz skrajno desnega sklopa. Nato bo naredil hipotezo, ki je blizu w, ter začel povpraševati po učnih primerih, ki so blizu te meje. Z nadaljnjimi označenimi primeri bo pozicija meje vedno bolj blizu w in vanjo bo imel vedno večje prepričanje. Tako nikoli ne bo našel idealne meje w^* ne glede na to, koliko oznak bo zahteval. Del prostora bo v tem primeru vedno ostal neraziskan.

4.3 Spodbudni rezultat

Prikažimo algoritem, po avtorjih poimenovan CAL [5], ki uporablja tokovne poizvedbe in lahko ob določenih predpostavkah doseže eksponentno boljši čas, kot bi ga s pasivnim učenjem. Algoritem predpostavlja ločljivost učnega problema ter končni koeficient nestrinjanja (*disagreement koeficient*) [13], ki bo definiran kasneje v tem razdelku. Ta koeficient se izkaže za dober kriterij za uspešno uporabo metod aktivnega učenja in ga zato zasledimo tudi pri drugih avtorjih.

CAL

Shema delovanja algoritma:

function CAL(št. poizvedb n)
  pridobi prostor konsistentnih hipotez V
  for $i = 1, 2, .., n$ **do**
    pridobi neoznačen primer x
    if vse hipoteze v V se strinjajo glede oznake primera x **then**
      zavrni x
    else
      zahtevaj oznako za x
      posodobi V
    end if
  end for

end function

Na tem mestu uvedimo območje nestrinjanja (*disagreement region*) kot množico vseh učnih primerov, za katere se $h \in V$ ne strinjajo. Označimo ga z $DIS(V)$. Učni primer x je torej sprejet (zahtevamo njegovo oznako) natanko tedaj, kadar leži v področju nestrinjanja. Področje nestrinjanja se tako v vsaki iteraciji zmanjša (verjetnostna gostota prostora je manjša), hitrost zmanjševanja pa odloča o uspešnosti algoritma. Videli bomo, da se ob ugodnih predpostavkah področje nestrinjanja v vsakem koraku prepolovi.

V praksi lahko prostor V implicitno vodimo tako, da si zapomnemo vse označene učne primere. Kadar dobimo nov učni primer x ga poskusimo označiti z 1 in skupaj z ostalimi označenimi podatki izdelamo klasifikator. Nato to ponovimo, le da x tokrat označimo z 0. Če oba klasifikatorja obstajata, potem x leži v območju nestrinjanja in zato zahtevamo njegovo oznako.

Definirajmo razdaljo med dvema hipotezama h in h' glede na verjetnostni prostor učnih primerov z

$$d(h, h') = P(h(x) \neq h'(x)) \tag{4.1}$$

in pripadajočo kroglo s polmerom r okoli hipoteze h

$$B(h, r) = \{h' \in H : d(h, h') \leq r\} \tag{4.2}$$

in končno koeficient nestrinjanja θ

$$\theta = \sup_{r>0} \frac{P(DIS(B(h^*, r)))}{r}$$

Vzemimo ciljno hipotezo h^*. Po nekaj poizvedbah upamo, da bo prostor V vsebovan v krogli okoli h^* s čim manjšim polmerom r. V tem primeru je verjetnost, da bomo zahtevali oznako naključnega primera manjša od $P(DIS(B(h^*, r))$. Koeficient nestrinjanja nam pove, kako se ta verjetnost spreminja glede na r.

Za pregled novih pojmov analizirajmo naš motivacijski primer. Krogla $B(h^*, r)$ v tem primeru predstavlja vse hipoteze, ki postavljajo mejo na intervalu $(h^* - r, h^* + r)$. Območje $DIS(B(h^*, r))$ so ravno vsi učni primeri na

tem istem intervalu. Zaradi uniformne porazdelitve je $P(x \in (h^* - r, h^* + r))$ enak $2r$. Koeficient nestrinjanja je v tem primeru torej $\frac{2r}{r} = 2$.

Omenimo še en razred problemov s končnim koeficientom nestrinjanja: Linearni separatorji v $\mathbb{R}^d$, ki grejo skozi izhodišče in kjer je porazdelitev učnih primerov uniformna na enotski krožnici. V tem primeru je $\theta \leq \sqrt{d}$ [6].

Izrek 4.1 *Recimo, da je naš učni problem ločljiv ima dimenzijo VC d in koeficient nestrinjanja θ. Potem*

$$L_{CAL}(\epsilon) \leq O(\theta d \log \frac{1}{\epsilon})$$

kjer je $L_{CAL}(\epsilon)$ število potrebnih poizvedb, da lahko zagotovimo, da ima naša hipoteza napako manjšo od ϵ z verjetnostjo večjo od $1 - \delta$. Pojavitev člena δ je v izrazu L izpuščena, saj nastopa kot največ $\log \frac{1}{\delta}$. Izrek je nekoliko presenetljiv, saj zagotavlja občutno izboljšanje v številu potrebnih primerov, kjub temu da ne izvajamo poizvedb na najbolj informativnih učnih primerih - zahtevamo, le da so informativni. Dokaz izreka lahko najdemo v [13].

DHM

Velika pomankljivost prejšnjega algoritma je v njegovi predpostavki ločljivosti učnega problema. V tem odseku bomo skicirali izboljšan algoritem, po avtorjih poimenovan DHM [8], za katerega velja podoben rezultat, kot je za algoritem CAL.

Da bi se spopadli z neločljivostjo problema, moramo drugače definirati prostor konsistentnih hipotez V, saj je po prejšnji definiciji najverjetneje prazen. V t-ti iteraciji bomo imeli t označenih primerov. Označimo z $err_t(h)$ empirično napako hipoteze, t.j delež izmed vseh t učnih primerov, ki jih h klasificira napačno. Z h_t označimo hipotezo z najmanjšo napako in nato lahko definiramo novi prostor V_i.

$$V_{t+1} = h \in V_t : err_t(h) \leq err_t(h_t) + \Delta_t$$

kjer Δ_t izhaja iz *standard generalization bound*. Za podrobnosti glej [8].

Upoštevajoč novo definicijo V nadaljujemo kakor v prejšnjem primeru: če se za nov dobljen učni primer x vsi $h \in V$ strinjajo, privzamemo njegovo oznako y' in ga vključimo k ostalim označenim učnim primerom, v nasprotnem zahtevamo njegovo oznako. Zaradi neločljivosti problema oznaka y' ni vedno pravilna, vendar s tem postopkom tako ali drugače označimo vse dane učne primere. Z označitvijo vseh učnih primerov se izognemo omenjeni težavi pristranosti vzorčenja.

Izrek 4.2 *Recimo, da je naš učni problem ločljiv ima dimenzijo VC d, koeficient nestrinjanja θ ter $\nu = inf_{h \in H} error(h)$. Potem*

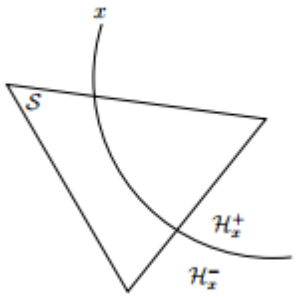
$$L_{DHM}(\epsilon) \leq O(\theta(d \log^2 \frac{1}{\epsilon} + \frac{d\nu^2}{\epsilon^2}))$$

Vidimo, da če je ν majhen v primerjavi z ϵ, potem je izboljšava tudi tukaj eksponentna. V nasprotnem imamo izboljšavo za faktor ν. Dokaz izreka lahko najdemo v [8].

4.4 Drugačni kriterij

Namen te sekcije je predstavitev novega kriterija, ki prav tako kot koeficient nestrinjanja, vpliva na uspeh aktivnega učenja. Vse definicije, izreki in dokazi v tej sekciji so povzete po [9], delu ki prvo uvede ta kriterij. Vzamimo v prejšnji sekciji definirano metriko (4.1) in pripadajočo kroglo (4.2). Da bi dobili hipotezo z napako manjšo od ϵ, je dovolj, da zmanjšamo prostor konsistentnih hipotez V do te mere, da je vsebovana v v $B(h^*, \epsilon)$, in nato vrnemo poljuben $h \in V$. V tem primeru rečemo, da ima V polmer manjši od ϵ. Nasprotno, v primeru da je polmer V večji od ϵ, iz V ne moremo z gotovostjo izbrati hipoteze, ki bo imela primerno majhno napako. Vidimo, da je glavni cilj učenja zmanjšati polmer prostora V.

Vsak učni primer x razdeli prostor V na dva dela: na $V_x^+ = \{h \in V : h(x) = 1\}$ in na simetrično definiran V_x^-. Vendar vsak tak rez ne zmanjša polmera V, kot kaže slika 4.4.



Slika 4.3: Učni primer x je očitno informativen, vendar ne zmanjša polmera prostora V. Slika je vzeta iz [9].

Definirajmo $Q \subset \binom{V}{2}$. Predstavljamo si lako graf, pri kateremu so pari $\{h, h'\} \in Q$ povezave med vozlišči h in h'. Te povezave bodo izbrane tako, da bodo predstavljale pare hipotez, ki jih želimo med seboj ločiti. Iskali bomo take x, da bodo odrezali kar se da velik del teh povezav. Za x pravimo, da ρ-loči Q, če ne glede na njegovo oznako odstrani $\rho|Q|$ povezav t.j če:

$$max\{|Q \cap \binom{V_x^+}{2}|, |Q \cap \binom{V_x^-}{2}|\} \leq (1-\rho)|Q|$$

Za primer glej sliko 4.4.

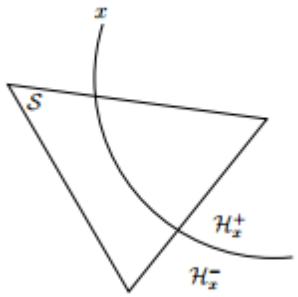
Zanimajo nas hipoteze, ki so med seboj oddaljene za več kot ϵ. Če takih hipotez ne bi bilo, potem bi bila tudi razdalja poljubne hipoteze h do ciljne hipoteze h^* manjša ali enaka ϵ in učenje bi bilo uspešno. V ta namen definirajmo

$$Q_\epsilon = \{\{h, h'\} \in Q : d(h, h') \leq \epsilon\}$$

Končno rečemo, da je neka podmnožica hipotez $S \subset V$, (ρ, ϵ, τ)-ločljiva, če za vsaj končen $Q \subset \binom{S}{2}$ velja:

$$P(\text{naključen učni primer } \rho\text{-loči } Q_\epsilon) \geq \tau$$

Delež τ nam tako pove delež učnih primerov uporabnih za ločitev S. Opazimo, da če lahko sestavimo povezavo dolžine $l \geq \epsilon$ in si za Q izberemo samo to



Slika 4.4: Na sliki so narisane povezave v Q in rez, ki ga naredi učni primer x. V tem primeru x 3/5-loči Q. Slika je vzeta iz [9].

povezavo, potem je $\tau \leq l$. Tako tipično pričakujemo, da bo τ manjši od ϵ (lahko je veliko manjši). Majhen τ pomeni, da dobri učni primeri sicer obstajajo, vendar moramo na voljo imeti veliko neoznačenih učnih primerov.

V nadaljevanju pokažemo, da je v primeru ločljivosti množice, število potrebnih oznak zgoraj omejeno z $\frac{1}{\rho}$ in število potrebovanih neoznačenih primerov zgoraj omejeno z $\frac{1}{\tau}$.

Lema 4.1 *Vzemimo poljuben* $0 < \alpha, \epsilon < 1$ *in poljubno množico* S. *Tedaj je* S $((1-\alpha)\epsilon, \epsilon, \alpha\epsilon)$*-ločljiv.*

Dokaz. Označimo z Z število povezav v Q_ϵ, ki jih izloči naključno izbran primer x. Ker imajo povezave dolžino vsaj ϵ, x z verjetnostjo vsaj ϵ preseka vsako izmed njih. Tako je matematično upanje $E(Z) \geq \epsilon|Q_\epsilon|$.

$$\epsilon|Q_\epsilon| \leq E(Z) \leq P(Z \geq (1-\alpha)\epsilon|Q_\epsilon|)|Q_\epsilon| + (1-\alpha)\epsilon|Q_\epsilon|$$

iz česar po preurejanju sledi

$$P(Z \geq (1-\alpha)\epsilon|Q_\epsilon|) \geq \alpha\epsilon$$

kar dokaže lemo. □

Iz zadnje leme sledi, da je ρ vedno $\Omega(\epsilon)$. Za uspešno učenje seveda pričakujemo precej večjo vrednost.

Izrek 4.3 *Vzamimo poljuben prostor hipotez H in poljubno porazdelitev učnih primerov P. Recimo da pri neki $0 < \rho, \epsilon < 1$ in pri nekem $0 < \tau < 1/2$ množica $S \subset H$ ni (ρ, ϵ, τ)-ločljiva. Potem vsaka strategija aktivnega učenja, ki z verjetnostjo večjo od 0.75 doseže klasifikacijsko točnost $\frac{\epsilon}{2}$ na vsaki hipotezi h v S, mora imeti na voljo ali več kot $\frac{1}{\tau}$ neoznačenih primerov ali več kot $\frac{1}{\rho}$ oznak.*

Dokaz. Vzemimo nek končni $Q_\epsilon \in \binom{S}{2}$ za katerega ločljivost ne velja in naj bodo $V \subset S$ njegova vozlišča:

$$V = \{h : \{h, h'\} \in Q_\epsilon, h' \in H\}$$

Pokazali bomo, da če želimo hipoteze v V ločiti med sabo, potrebujemo ali $\frac{1}{\tau}$ neoznačenih primerov ali $\frac{1}{\rho}$ oznak.

Recimo, da imamo manj kot $\frac{1}{\tau}$ naoznačenih učnih primerov. Po definiciji vsak učni primer z verjetnosto vsaj $(1-\tau)$ ne ρ-loči Q_ϵ. Torej, z verjetnostjo vsaj $(1-\tau)^{(1-\tau)} \geq 1/4$, nobena izmed teh točk ne ρ-loči Q_ϵ. V tem primeru ima vsak učni primer "slabo oznako", ki odstrani manj kot $\rho|Q_\epsilon|$ povezav. Da bi vse hipoteze ločili med seboj moramo odstraniti vse povezave v Q_ϵ, za kar potrebujemo več kot $\frac{1}{\rho}$ oznak. □

Z drugimi besedami trditev pove, da če ima neko območje v prostoru hipotez nizek indeks ločljivosti, potem v tem območju obstaja hipoteza, ki je ni lahko identificirati z metodami aktivnega učenja.

Sledi opis algoritma, ki v vsaki iteraciji prepolovi polmer prostora V. Algoritem sam je sicer računsko prezahteven in bo služil samo za postavitev zgornje meje za število potrebnih neoznačenih učnih primerov ter oznak v primeru dobrega indeksa ločljivosti.

Algoritem

Shema algoritma:

izberi $\epsilon_0 < \epsilon$

$S_0 = \epsilon_0$-pokritje prostora H, t.j taka množica, da so vse hipoteze v H oddaljene za manj kot ϵ_0 od S_0. Pomembno je poudariti, da vedno obstaja

končno tako pokritje; tako S_0 predstavlja končni nadomestek celotnega prostora H.

for $t \in \{0, 1, ..., T = \lg \frac{2}{\epsilon}\}$ **do**

S_t=loči($S_{t-1}, 1/2^t$)

end for

return poljuben $h \in S_T$

Sledi shema procedure *loči*(S, Δ). Namen te procedure je iterativno pridobivanje novih neoznačenih primerov in nekaterih oznak, dokler se polmer prostora V ne prepolovi.

function LOČI(S, Δ)

$Q_0 = \{\{h, h'\} \in \binom{S}{2} : d(h, h') > \Delta\}$

for $t \in \{0, 1, 2...\}$, dokler Q_t ni prazen **do**

Pridobi m neoznačenih primerov $x_{t1}, x_{t2}, ..., x_{tm}$

Poišči x_{ti}, ki maksimalno loči Q_t

Pridobi njegovo oznako

Q_{t+1} = preostale povezave

end for

return prostale hipoteze v S

end function

Naslednji izrek povzame uspeh zgornjega algoritma.

Izrek 4.4 *Naj bo h^* ciljna hipoteza. Vzamimo poljubni $\epsilon > 0$ in $\delta > 0$. Predpostavimo, da je $B(h^*, 4\Delta)$ (ρ, Δ, τ)-ločljiv za vsak Delta $\geq \frac{\epsilon}{2}$. Potem obstaja ϵ_0 in m tako da bo, z verjetnostjo $1 - \delta$ algoritem vrnil hipotezo s klasifikacijsko točnostjo manjšo od ϵ in pri tem potreboval:*

$$\text{št. neoznačenih primerov } \leq O(\frac{d}{\rho\tau} \log \frac{1}{\epsilon} \log \frac{1}{\epsilon\tau})$$

$$\text{št. poizvedb } \leq O(\frac{d}{\rho} \log \frac{1}{\epsilon} \log \frac{1}{\epsilon\tau})$$

Dokaz izreka je naveden v dodatku.

Zahtevati, da so vse okolice h^* ločljive, navadno ni mogoče, vendar lahko včasih pokažemo, da so ločljive vse manjše okolice h^*. Tako iskanje poteka v dveh korakih, prvem in počasnejšem, v katerem se V zmanjša, tako da je vsebovan v $B(h^*, r)$ za nek majhen r in drugem, kjer zgornji izrek drži in je preiskovanje učinkovito.

Brez dokaza omenimo, da so linearni separatorji, ki grejo skozi izhodišče in kjer je porazdelitev učnih primerov uniformna na enotski krožnici, $(\frac{1}{4}, \epsilon, \Omega(\epsilon))$-ločljivi za vsak ϵ. [9]

4.5 Bayesova predpostavka

Sledi povzetek analize dela Y.Freund [12]. Analiziral je binarno klasifikacijo s tokovnim modelom poizvedb in pristopom komisijskega izbora. Za učni problem predpostavlja deterministočnost in separabilnost ter Bayesovsko porazdelitev hipotez, t.j. možne hipoteze so razporejene po neki poznani apriori porazdelitvi.

Prvo naravno merilo za napredek učenja je hitrost zmanjševanja prostora konsistentnih hipotez. Naj bo V_i prostor konsistentnih hipotez v i-ti iteraciji učnega algoritma. Definirajmo takojšnji informacijski dobitek (*Instantaneous information gain*)

$$Ig = -\log \frac{P(V_i)}{P(V_{i-1})}$$

Ter kumulativni informacijski dobitek (*Cumulative information gain*)

$$Kg = -\sum_i \log \frac{P(V_i)}{P(V_{i-1})}$$

Želimo ugotoviti pričakovani takojšnji informacijski dobitek za poizvedbo nekega učnega primera x - označimo ga z $G(x|V_{i-1})$.

$$G(x|V_{i-1}) = -p_0 \log \frac{P(V_{i_0})}{P(V_{i-1})} - p_1 \log \frac{P(V_{i_1})}{P(V_{i-1})}$$

kjer je V_{i_1} prostor konsistentnih hipotez v primeru, da je vrnjena oznaka primera x enaka 1 in je p_1 verjetnost te oznake. Podobno za V_{i_0} in p_0.

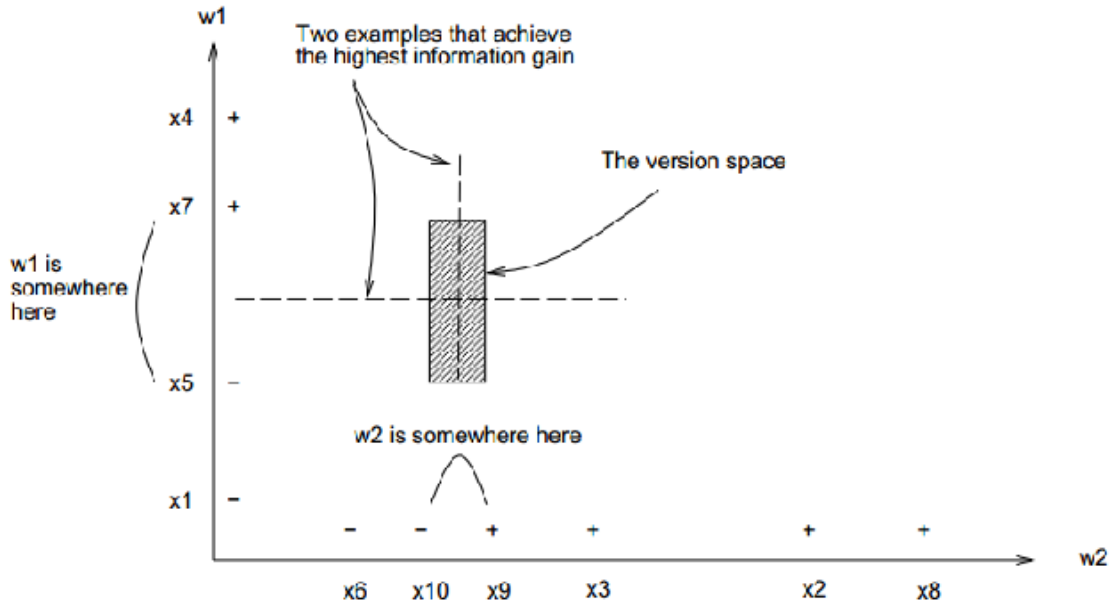
Velja: $P(V_{i_1}) = P(V_{i-1}, x = 1) = P(V_{i-1}) * P(x = 1|V_{i-1}) = p_1 * P(V_{i-1})$
Zato se zgornji izraz okrajša v

$$G(x|V_{i-1}) = -p_0 \log p_0 - (1 - p_0) \log(1 - p_0) = H(x)$$

kar je ravno informacijska vrednost spremenljivke in ima maksimum pri $p_0 = p_1 = 0.5$. Vidimo torej, da se ob Bayesovski predpostavki metoda najmanjše zanesljivosti ter metoda zmanjšanja prostora hipotez ujemata.

4.5.1 Protiprimer

To merilo je atraktivno, vendar ni zadostno da bi zagotovilo uspeh aktivnega učenja. To pokažimo na preprostem protiprimeru.



Slika 4.5: Prikaz učnega problema. Slika je vzeta iz [12].

Imejmo prostor učnih primerov $X = 1, 2 \times [0, 1]$ in hipoteze oblike $h_w(i, z) = 1$, če $w_i > z$ in 0 v nasprotnem. Prva komponenta i je izbrana naključno

izmed $1, 2$, druga poimenovana z uniformo iz intervala. Gre za nekakšno posplošitev motivacijskega primera, kjer se mora učenec naučiti dveh pragov. Prostor hipotez si lahko predstavljamo kot pravokotnik (glej sliko 4.5). Ni težko videti, da imata največji informacijski dobitek učna primera, ki prepolovita pravokotnik po eni ali po drugi strani.

Vzemimo, da učni algoritem za poizvedbo vedno izbere učni primer na isti izmed osi. Tedaj bo ta pravokotnik po tej osi v vsakem koraku prepolovil in tako v vsakem koraku zmanjšal prostor V za faktor 2. Vidimo, da bi se velikost prostrora V hitro spustila proti 0.

Po drugi strani je pričakovana napaka končnega prostora hipotez proporcionalna obsegu pravokotnika. Pričakovano napako lahko ocenimo tako da računamo pričakovano napako naključno izbrane konsistentne hipoteze glede na njihovo porazdelitev. Velja da je ta napaka največ dvakrat večja od prave.

$$P(napaka) = 0.5 * P(napaka|i = 1) + 0.5 * P(napaka|i = 2)$$

Označimo s a in b krajišča pravokotnika na eni osi, s d in c na drugi. Označimo s h_1 neznani pravi prag na prvi osi, vemo da je izbran uniformno iz intervala.

$$\begin{aligned} P(napaka|i = 1) &= \int_a^b P(w_1 = x)P(napaka|i = 1, w_1 = x))\,\mathrm{d}x \\ &= \frac{1}{(b-a)} \int_a^b \int_a^b P(h_1 = y)P(napaka|i = 1, w_1 = x, h_1 = y)\,\mathrm{d}y\,\mathrm{d}x \\ &= \frac{1}{(b-a)^2} \int_a^b \int_a^b |x - y|\,\mathrm{d}y\,\mathrm{d}x \\ &= \frac{b-a}{3} \end{aligned}$$

Simetrično je $P(napaka|i = 0) = \frac{d-c}{3}$, in je tako celotna napaka enaka $\frac{((b-a)+(d-c))}{6}$, kar je kot omenjeno proporcionalno obsegu.

Kljub hitremu zmanjševanju velikosti prostora hipotez bo obseg (v omenjenem primeru da poizvedujemo le na eni osi) in s tem pričakovana napaka vedno ostal nad neko konstanto ne glede na število poizvedb. Primer toka

takih poizvedb bi bil $(1, 0.5)$, $(1, 0.25)$, $(1, 0.125)$... Vidimo, da je množica tako označenih primerov nereprezentativna glede na celoten prostor učnih primerov, saj ne vsebuje nebenega primera oblike $(2, x)$ in posledično nemore dosežti visoke klasifikacijske točnosti ne glede na število poizvedb.

Vidimo torej, da tudi hitro zmanševanje prostora hipotez in velik informacijski dobiček poizvedbe ne zagotavljata uspešnosti učenja.

4.5.2 Algoritem

Naslednji algoritem uporablja tokovne poizvedbe in pristop komisijskega izbora.

pridobi prostor konsistentnih hipotez V
for $t \in \{0, 1, 2...\}$, dokler ne zaporedoma zavrnemo zadostno število učnih primerov **do**
    zahtevaj neoznačen učni primer x
    vzorči dve hipotezi v V glede na njihovo porazdelitev
    primerjaj njune predikcije
    if predikciji sta enaki **then**
        zavrni x
    else
        zahtevaj oznako tega primera
        posodobi V
    end if
end for

Zadostno število zaporednih zavrnjenih primerov je enako

$$t_n = \frac{1}{\epsilon} \ln(\pi^2 \frac{(n+1)^2}{3\delta})$$

kjer je n število do sedaj opravljenih poizvedb. V tem primeru velja, da če se algoritem ustavi, potem je pričakovana napaka manjša od ϵ z verjetnostjo večjo od $1 - \delta$. Večja težava je ugotoviti, ali se res kdaj ustavi. Pokazali bomo, da obstaja razred problemov, za katere se ta algoritem ustavi z veliko

verjetnostjo ter da pri tem porabi $O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon\delta}))$ neoznačenih učnih primerov ter $O(log(\frac{1}{\epsilon}))$ oznak.

Intuitivno lahko vidimo, da tudi QBC poskuša delati poizvedbe na učnih primerih, ki kar se da razpolovijo prostor hipotez, saj če se glede tega učnega primera strinja večina hipotez, je možnost zavrnitve velika. Vzemimo, da nek učni primer x razdeli prostor hipotez na dva dela z verjetnostmi F in $1-F$. Verjetnost sprejetja tega učnega primera je enaka $2F(1-F)$. Maksimum tega izraza sovpada z maksimumom informacijskega dobitka $H(F)$. Tako imajo učni primeri, izbrani z metodo QBC, večji pričakovan informacijski dobitek kot nakjučni primeri. Ne velja pa vedno, da je pričakovani informacijski dobitek izbranih učnih primerov vedno večji od neke konstante.

Navedimo nekaj oznak. Naj bo $I = \{i_1, i_2, i_3...\}$ množica indeksov sprejetih učnih primerov, naj bo $X_I = \{x_{i_1}, x_{i_2}, x_{i_3}...\}$ množica sprejetih učnih primerov, $X_M = \{x_1, x_2...x_M\}$ prvih M učnih primerov, naj bo $I_N = \{i_1, i_2, ...i_N\}$ prvih N indeksov sprejetih učnih primerov,$X_{i_n} = \{x_{i_1}, x_{i_2}, ...x_{i_n}\}$ prvih N sprejetih učnih primerov in nazadnje $X_{I\cap M}$ izbrani učni primeri izmed prvih M zaporednih primerov

Imamo prostor trojice $\Omega(c, X, I)$ in porazdelitevΔ nad njo. Porazdelitev Δ upošteva porazdelitev po prostoru hipotez P, porazdelitev po prostoru učnih primerov D in porazdelitev čez vse možne I.

Definicija 4.1 *Pričakovan informacijski dobitek poizved narejenih s QBC je uniformno spodaj omejen, če je pričakovan informacijski dobitek za $n+1$-vo poizvedbo v prostoru $\Omega(c, X, I)$ večji od q za vsak n in za vsako zaporedje poizvedb. Drugače:*

$$P_\Delta(E(G(x_{i_{n+1}}|V(X_{I_n}, c(X_{I_n}))))|X_{I_n}, c(X_{I_n})) > q) = 1$$

Intuitivno: v vsakem prostoru hipotez, ki ga lahko dosežemo s neničelno verjetnostjo, bo naslednja poizvedba imela takojšnji informacijski dobitek večji od q.

Izrek 4.5 *Imejmo razred hipotez C, ki ima končno dimenzijo VC d in njegov pričakovan informacijski dobitek je spodaj omejen z $q > 0$. Potem se bo z*

verjetnostno $1-\delta$ *algoritem ustavil in vrnil hipotezo s pričakovano napako, ki je manjša od* ϵ *in pri tem porabil* m_0 *neoznačenih učnih primerov ter* n_0 *označenih.*

$$m_0 = \max(\frac{4d}{e\delta}, \frac{160(d+1)}{q\epsilon}\max(6, \ln\frac{80(d+1)}{q\epsilon\delta^2})^2)$$

$$n_0 = \frac{10(d+1)}{q}\ ln(\frac{4m_0}{\delta})$$

Skica dokaza te trditve je navedena v dodatku.

Vidimo, da če pogoj o uniformni spodnji meji informacijskega dobička drži, potem je učni problem učinkovito učljiv s samo logaritemsko malo oznakami. Poudarimo, da algoritem še vedno potrebuje $O(\frac{1}{\epsilon})$ neoznačenih učnih primerov, s katerimi si intuitivno gledano pomaga oceniti distribucijo primerov iz X. Brez dokaza omenimo dva razreda učnih problemov, kjer je pogoj izpolnjen.

Primer paralelnih ravnin

X predstavljajo vsi pari oblike (x,t), kjer je x normaliziran vektor v $\mathbb{R}^d$ in t je realno število v $[-1,1]$. Hipoteze so parametrizirane s normaliziranimi vektorji w v $\mathbb{R}^d$ in oblike $h_w(x,t)=1$, če $wx > t$ in 0 sicer. Apriori distribucija v prostoru hipotez je uniformna na enotski krožnici v $\mathbb{R}^d$.

Ta učni problem se lahko nato razširi na naslednjega.

Perceptroni

$c_w(x)=1$, če $wx >$t in 0 sicer. V primerjavi s prejšnjim problemom, je tukaj t del hipoteze, ne učnega primera.

Definicija 4.2 *Distribucija* D' *je od* D *oddaljena za* λ*, če za vsako merljivo množico* A *velja:* $\lambda \leq \frac{P_D(A)}{P'_D(A)} \leq \frac{1}{\lambda}$

Izrek 4.6 *Za vsak* $\alpha > 0$ *je* H_α *prostor hipotez, ki vsebuje samo tiste* w*, za katere velja* $ww_0 > \alpha$*, za nek vektor* w_0*. Naj bo porazdelitev* P *za* λ_p *oddaljena*

od uniformne, ter porazdelitev D za λ_d oddaljena od uniformne. Tedaj je uniformna spodnja meja informacijskega dobitka vetja od $0.672\alpha^{5d}\lambda_p^4\lambda_d$

Poglavje 5

Naši preizkusi

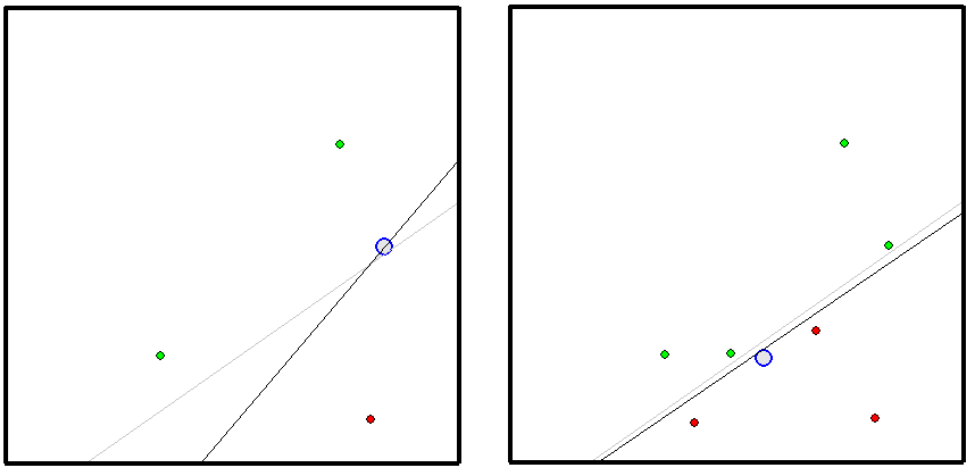
V tem poglavju predstavimo rezultate poizkusov, ki smo jih sami izvedli. Testirali smo različne pristope in primerjali njihovo učinkovitost.

Vsa programska koda je bila napisana v programskem jeziku *Python* (http://www.python.org/). Za izdelavo klasifikatorjev je bil uporabljen paket *scikit-learn* (http://scikit-learn.org/). V tem paketu so bile vključene tudi zbirke podatkov, na katerih smo testirali. Grafi z rezultati so bili narisani s paketom *pylab* (http://www.scipy.org/PyLab).

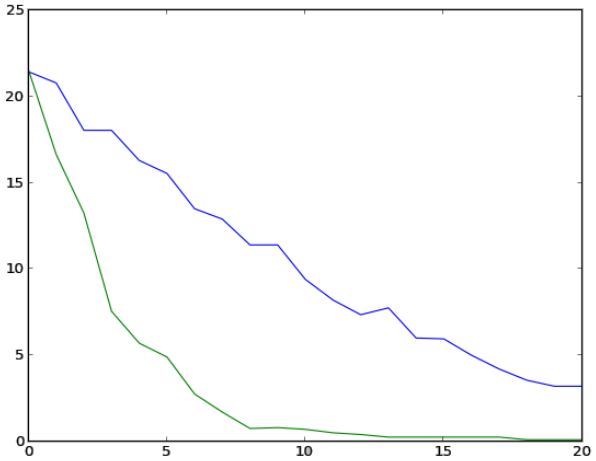
5.1 Enostaven primer

Testiranja smo se najprej lotili na preprosti umetno generirani binarni domeni: točke na ravnini, ki so linearno ločljive. Generirali smo množico točk, nato smo z uporabo metode SVM in principa najmanjše zanesljivosti poskušali dobiti dober klasifikator. Izkazalo se je, da za to nalogo potrebujemo presenetljivo malo točk. Stanje po nekaj učnih primerih je prikazano na sliki 5.1.

Da bi potrdili učinkovitost aktivnega izbiranja primerov, smo ga primerjali z naključnim izbrianjem primerov. Test je bil dvajsetkrat izveden na različnih generiranih podatkih in ciljni premici. Rezultat je bil nato povprečen in prikazan na sliki 5.2. Vidimo, da je razlika med aktivnim in pasiv-



Slika 5.1: Na sliki so z zeleno in rdečo narisani izbrani učni primeri. Dejanska meja je označena s sivo, trenutna meja pa s črno barvo. Moder krog prikazuje učni primer, ki si ga učenec želi v trenutni iteraciji. Leva slika prikazuje stanje po treh poizvedbah, na desni jih je bilo opravljeno sedem. Lahko vidimo, da so učni primeri dobro izbrani in je dejanska meja v zadnjem primeru že zelo natančna.



Slika 5.2: Na tej sliki je predstavljen graf napake (v procentih) v odvisnosti od števila izbranih učnih primerov. Uspeh pasivnega učenca je označen z modro, uspeh aktivnega učenca z zeleno barvo.

nim učencem v tej domeni zelo velika.

5.2 Prepoznavanje števk

Prejšnji primer je bil preprost, nizko dimenzionalen in brez šuma ter je bolj kot ne služil vizualizaciji učenja, zato smo se lotili nekoliko težje domene. Iz paketa *scikit-learn* smo vzeli zbirko podatkov *digits*, kjer so bili podatki (slike velikosti 8x8) klasificirani v enega izmed desetih razredov (števke od 0 do 9). Na tej domeni smo testirali vrsto pristopov:

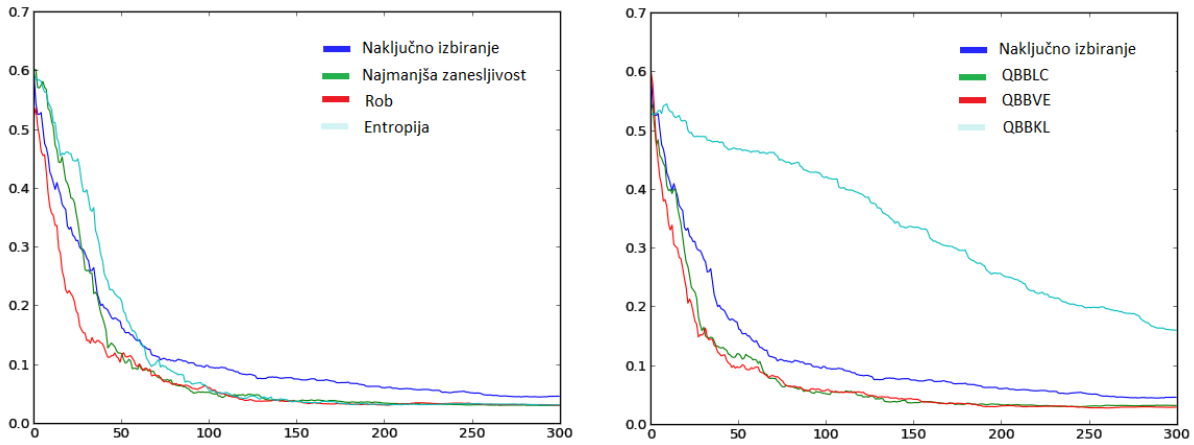
- pristop najmanjše zanesljivosti,
- pristop najmanjše zanesljivost z mero najmanjšega roba,
- pristop najmanjše zanesljivosti z mero entropije,
- komisijski izbor z mero entropije glasovanja (2.2) - označimo QBBVE,
- komisijski izbor z mero *KL divergence* (2.1) - označimo QBBKL,
- komisijski izbor z mero najmanjšega prepričanja - označimo QBBLC,
- naključno izbiranje primerov, ki naj služi za primerjavo.

Pri vseh primerih s komisijskim izborom smo klasifikatorje ustvarili z metodo *bagging*. Nekateri drugi pristopi, omenjeni v prejšnjih poglavjih, so izpuščeni zaradi njihove računske kompleksnosti.

Na sliki 5.3 so prikazani rezultati pri uporabi metode SVM. Testiranje je bilo izvedeno z začetno množico primerov velikosti 20 in 5 kratnim prečnim preverjanjem.

Pokazalo se je, da je v tej domeni aktivno učenje precej uspešno, saj se vsi pristopi, z izjemo *KL divergence*, obnesejo precej bolje kot naključno izbiranje primerov. Zanimivo je tudi, da je uspeh večine pristopov presenetljivo podoben, nekoliko se razlikujejo samo na začetku učenja. Na začetku učenja

napaka pri pasivnem učencu pada hitreje kot pri nekaterih aktivnih pristopih, kar bi lahko interpretirali, da učenec na začetku domene ne pozna dovolj dobro in so zato njegove poizvedbe slabše.



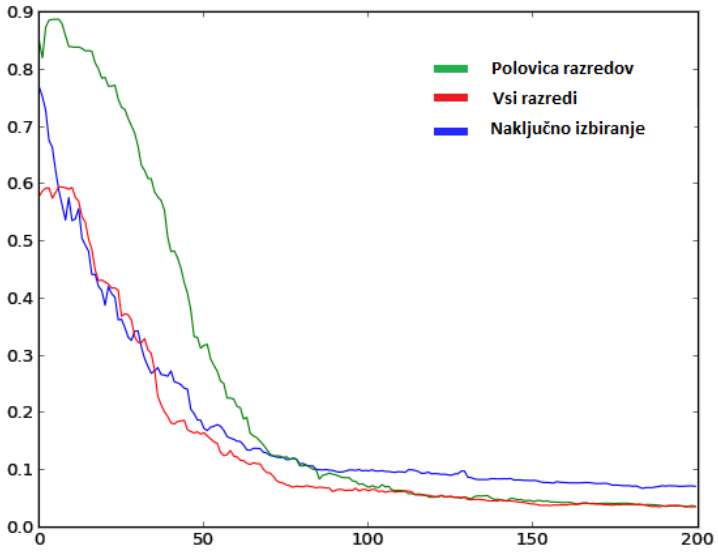
Slika 5.3: Na tej sliki je predstavljen graf napake v odvisnosti od števila izvedenih poizvedb. Na obeh slikah je pasivni učenec prikazan z modro barvo in lahko vidimo, da je njegova napaka večja od napak aktivnih učencev.

Zaradi podobnosti uspeha pristopov smo v nadaljnjih testiranjih primerjali le nakjučno izbiranje in pristop najmanjše zanesljivosti.

Začetna množica

Začetna množica učnih primerov je bila tu izbrana naključno in brez zagotovila, da bo vsebovala vse možne razrede. Lahko bi domnevali, da bodo zaradi mankajočih razredov poizvedbe slabše in bo rezultat učenja, še posebej na začetku, slabši. Domnevo smo testirali tako, da smo umetno nastavili dve začetni množici: ena je vsebovala vseh deset razredov, druga le pet. Rezultati so prikazani na sliki 5.4.

Vidimo, da je učenje v drugem primeru očitno slabše in učenec potrebuje veliko učnih primerov, da začenja dohitevati uspeh učenja v prvem primeru. Učenec v drugem primeru je v poprečju potreboval okoli 30 poizvedb, da si je



Slika 5.4: Na sliki sta predstavljena uspeha aktivnega učenja po metodi najmanjše zanesljivosti pri začetni množici z vsemi razredi in pri začetni množici z le polovico razredov. Za primerjavo je dodan še uspeh pasivnega učenca z začetno množico z vsemi razredi.

zagotovil vsaj en učni primer za vsak razred. Ta rezultat sicer ni presenetljiv, vendar pokaže, da je pomembno izbrati dobro začetno množico.

Porazdelitev razredov

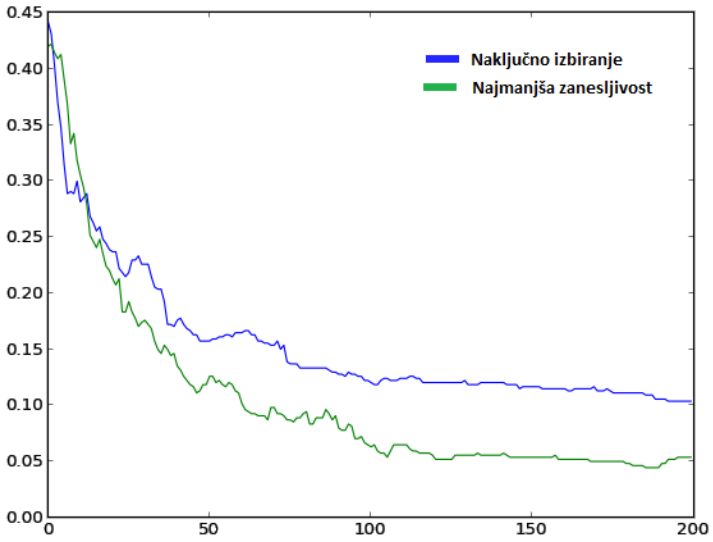
V tej množici podatkov so bili vsi razredi zastopani z enako verjetnostjo. Opazili smo, da se ta porazdelitev v končni množici izbranih primerov ohrani ne glede na izbran pristop. Zanimalo nas je, ali bi tudi iz bolj neenakomerne začetne množice (glede na porazdelitev razredov) aktivni učenec izbral približno isto število učnih primerov, ki pripadajo vsakemu razredu, in ali bi mu to dalo dodatno prednost pred pasivnim učencem. V ta namen smo iz originalnih podatkov *digits* izbrisali delež učnih primerov tako, da je bil razred *1* najbolj zastopan, razred *2* drugi najbolj zastopan, do razreda *10*, ki je bil najmanj zastopan. Nato smo aktivnega učenca testirali na tej spremenjeni

množici.

V tabeli 5.1 je prikazano število izbranih učnih primerov iz vsakega razreda. V primeru aktivnega učenca so posamezni razredi veliko enakomernejše zastopani. Testiranje smo ponovili tudi z drugimi pristopi in izkazalo se je, da vsi zelo enakomerno izbirajo posamezne razrede. Razlika v klasifikacijski točnosti je bila v tem primeru še nekoliko večja, kot v primeru z enakomerno razporejenimi razredi, in je prikazana na sliki 5.5.

Naključni primeri	50	29	30	26	27	23	14	14	5	2
Najmanjše zaupanje	22	33	24	29	23	23	17	19	19	11

Tabela 5.1: Število izbranih primerov iz vsakega razreda.



Slika 5.5: Na sliki je predstavljen graf napake v odvisnosti od števila izvedenih poizvedb. Testiranje je bilo izvedeno na spremenjeni množici *digits*, ki ima neenakomerno zastopane razrede.

Izbiranje večih učnih primerov hkrati

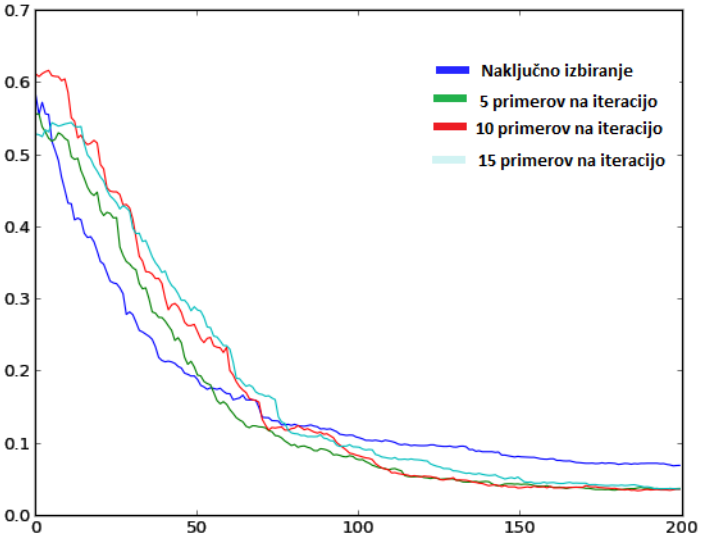
Aktivno učenje prinese celotnemu postopku občutno dodatno časovno kompleksnost. V primerjavi s pasivnim učenjem, kjer se moramo naučiti le en klasifikator, moramo pri aktivnem učenju klasifikator izdelati za vsak nov prejet učni primer, poleg tega moramo tudi v vsakem koraku poskušati klasificirati vse ostale učne primere. Tako je tudi z enostavnimi pristopi učenje lahko zelo dolgo. Temu problemu se lahko poskušamo izogniti tako, da v vsaki iteraciji izberemo več kot en učni primer. Naivni pristop k temu problemu je k označevalcu poslati n najboljših učnih primerov glede na enak kriterij, kot če bi izbirali le enega. V zameno za enostavnost ima pristop težave: ne upošteva, koliko se informativnost teh n primerov prekriva. Lahko so si vsi zelo podobni in zato vsi informativni, vendar bi bil eden izmed njih dovolj, da se učenec nauči nekaj o tistem delu domene. Za dober algoritem bi moral učenec poleg posamezne informativnosti upoštevati tudi njihovo medsebojno prekrivanje. Kljub temu nas je zanimalo, ali pri naivni metodi zgornji pomislek drži in koliko (v tej specifični domeni) pada uspešnost učenja glede na število hkrati izbranih primerov. Rezulat lahko vidimo na sliki 5.6.

5.3 Dodatni primeri

Da bi se prepričali, ali ni bil uspeh aktivnega učenja vezan samo na prejšnjo specifično kombinacijo učnih podatkov in medote učenja, smo klasifikacijsko točnost poskušali testirati še na drugače postavljenih učnih problemih.

Najprej na enaki množici podatkov različne pristope preizkusimo z drugačno metodo strojnega učenja: naivni Bayes. Čeprav smo s to metodo dosegli slabše klasifikacijske točnosti, kot smo jih z metodo SVM, so bili rezultati aktivnega učenja še vedno boljši od pasivnega. Prikazani so na sliki 5.7 .

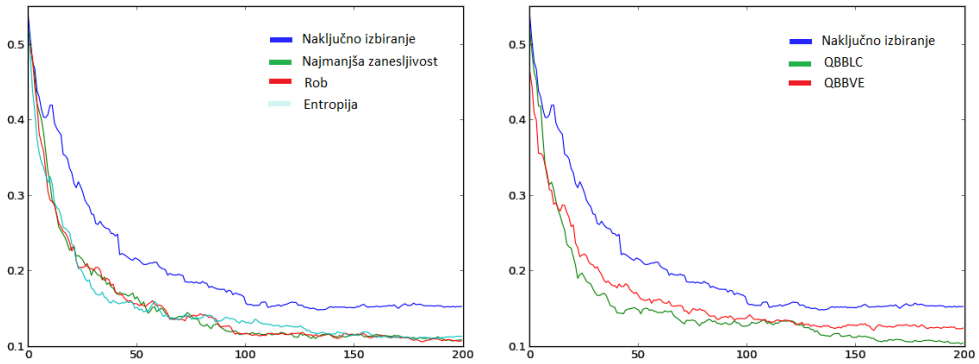
Sledi rezultat iz množice podatkov poimenovane *iris*. Gre za manjšo množico učnih primerov, ki so razvrščeni v tri razrede. Uporabili smo metodo *logistic regression*, saj se je pri pasivnem učenju na tej množici najbolj izkazala. Zaradi manjše množice smo dovoli samo 30 poizvedb. Uspeh je



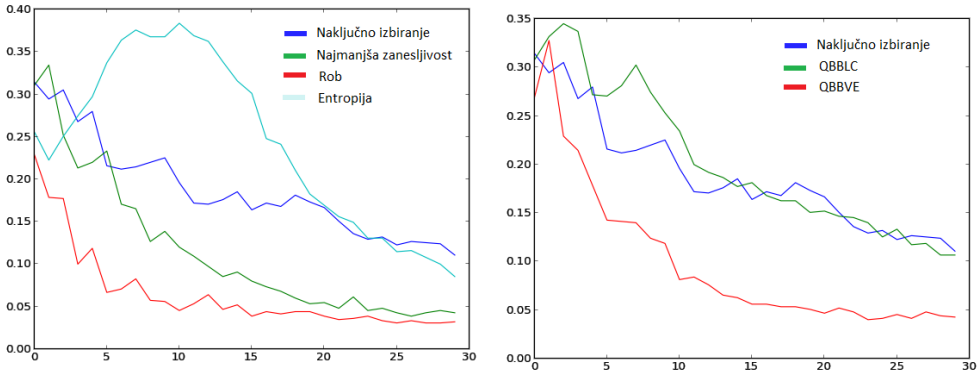
Slika 5.6: Na sliki je predstavljeno padanje kvalitete učenjaglede na število učnih primerov, ki jih zahtevamo v vsaki iteraciji. Vidimo, da za pospešitev učenja plačamo s slabšo klasifikacijsko točnostjo.

prikazan na sliki 5.8.

Lahko zaključimo s trditvijo, da je v naših testiranjih aktivno učenje res delovalo in prineslo boljšo klasifikacijsko točnost z enakim številom učnih primerov. Preizkušene metode so delovale zelo podobno, pri čemer je metoda najmanšega roba delovala za odtenek boljše kot ostale.



Slika 5.7: Prikazana je napaka glede na število opravljenih poizvedb na domeni *digits*. Učenje je bilo opravljeno z naivnim Bayesom.



Slika 5.8: Prikazana je napaka, glede na število opravljenih poizvedb na domeni *iris*.

Poglavje 6

Zaključne misli

V tem diplomskem delu smo predstavili pregled aktivnega učenja; pokazali smo, v kakih scenarijih ga lahko uporabimo ter kopico kriterijev, s katerimi lahko izberemo naslednji učni primer.

Začeli smo z željo, da bi z dobro izbiro informativnih učnih primerov lahko dosegli eksponentne prihranke v številu potrebnih učnih primerov. V poglavju s teoretičnimi evalvacijami smo našteli in kratko opisali nekaj zadostnih kriterijev za učne probleme, ki bi se jih (vsaj teoretično) dalo zelo učinkovito rešiti z aktivnim učenjem. Opazili smo, da ti kriteriji veliko zahtevajo in zato za večino (realnih) učnih problemov ne veljajo. Prav tako je težko pokazati, ali nek učni problem, ki nam je na razpolago, dejansko zadošča kateremu izmed naštetih kriterijev. Po drugi strani smo v poglavju s praktičnimi evalvacijami in v lastnih testiranjih ugotovili, da aktivne metode v resnici v večini primerov izboljšajo kvaliteto učenja. Izboljšava sicer ni eksponentna, kot bi si morda želeli, vendar je vseeno občutna.

Aktivno učenje nam torej omogoča, da v primerih, ko lahko zastonj oziroma relativno poceni pridobimo učne primere, izboljšamo učenje. Omenjeni pogoj je marsikdaj izpolnjen, zato lahko te metode uporabimo v raznovrstnih situacijah. Glavna ovira pri uporabi aktivnega učenja je dodatna časovna zahtevnost, ki jo prinese. Videli smo, da je lahko ta dodatek k času nezanemarljiv in lahko onemogoči uporabo teh metod v časovno občutljivih

primerih. Dodatni čas lahko privarčujemo z uporabo enostavnih metod za izbiro naslednjega primera (le te so se v testiranjih obnesle skoraj tako dobro, kot časovno zahtevnejše) in z izbiro večih primerov v vsaki iteraciji učenja.

Predstavljeno področje se še vedno zelo razvija; izboljšuje se tako teoretična podlaga, kot praktični napotki za uspešno uporabo opisanih metod. Kljub temu je področje že dozorelo do te meje, da se lahko zanesljivo uporablja v resničnih programih.

Dodatek A

Dokazi

Ta razdelek je namenjen dokazom, ki so bili v prejšnjih razdelkih zaradi boljše berljivosti izpuščeni. V sledečih dokazih pogosto uporabljamo oceno $1 + x < e^x$, ki izhaja iz razvoja funkcije e^x v Taylorjevo vrsto. Iz tega takoj sledi tudi ocena

$$1 - x < e^{-x}$$

A.0.1 Izrek 4.4

V tej sekciji pokažemo dokaz, vzet iz [9], za izrek 4.4. Pri tem potrebujemo naslednje 3 leme.

Lema A.1 *Recimo, da je $S \subset H$ (ρ, Δ, τ)-ločljiva. Potem bo z verjetnostjo vsaj $1 - \frac{1}{\rho}\ln(|Q_0|)e^{-m\tau}$, procedura* deli$(S, \Delta)$ *končala in pri tem zahtevala največ $1 - \frac{1}{\rho}\ln(|Q_0|)$ poizvedb.*

Dokaz. Vzemimo poljuben $t' > 0$.

$$\begin{aligned} P(\text{Za nek } t < t' \text{ ne obstaja noben } x_{ti}, \text{ ki bi } \rho\text{-ločil } Q_t) &\leq \sum_{t=0}^{t'-1} P(\text{Noben } x_{ti} \text{ ne } \rho\text{-loči } Q_t) \\ &\leq t'(1-\rho)^m \\ &\leq t' e^{-m\tau} \end{aligned}$$

Če se ta neugoden dogodek ne zgodi, potem imamo v času t' še $|Q_{t'}|$ povezav.

$$|Q_{t'}| \leq |Q_0|(1-\rho)t' < |Q_0|e^{-\rho t'}$$

Če za t' vzamemo $\frac{1}{\rho}\ln|Q_0|$ potem je $|Q_{t'}|$ ¡1 in procedura se ustavi. Ni težko videti, da število zahtevanih poizvedb v tem primeru ustreza zahtevam leme. □

Lema A.2 *Recimo, da je S_t $(\rho, 1/2^{t+1}, \tau)$-ločljiv za vse $t = 1, 2, ..., \lg\frac{2}{\epsilon-1}$. Potem z verjetnostjo vsaj $1-\frac{1}{\rho}\ln(|S_0|^2)\lg(\frac{2}{\epsilon})e^{-m\tau}$ je celotno število zahtevanih neoznačenih učnih primerov enako*

$$M \leq \frac{m}{\rho}\ln(|S_0|^2)\lg\frac{2}{\epsilon}$$

Dokaz. To lemo bomo dokazali z uporabo leme A.1 in seštevanjem čez vse klice *loči*. V primeru, da se vsi klici končajo po $\frac{1}{\rho}\ln|Q_0|$ iteracijah in upoštevajoč da v vsaki iteraciji potrebujemo m neoznačenih primerov, ter da je $|S_0|^2 \geq |Q_0|$ je število zahtevanih neoznačenih primerov

$$M \leq \sum_{t=0}^{\lg(2/\epsilon)} \frac{m}{\rho}\ln|S_0|^2 = \frac{m}{\rho}\ln(|S_0|^2)\lg\frac{2}{\epsilon}$$

Ta dogodek se zgodi z verjetnostjo, ki jo izračunamo na podoben način. □

Ko se celoten algoritem konča, imamo ϵ_0-pokritje prostora H zreducirano na polmer dolžine $\leq \frac{\epsilon}{2}$. Preveriti moramo ali ni morda prazen.

Lema A.3 *Vzemimo ciljno hipotezo h^* in njej najbližji element v S_0 označen z h_0. Če smo zahtevali M neozačenih učnih primerov, potem je verjetnost da je $h_0 \in S_T$ vsaj $1 - M\epsilon_0$.*

Dokaz. Verjetnost, da poljubni primer loči $h*$ od h_0 je $d(h_0, h^*) \leq \epsilon_0$. Torej z verjetnostjo $1 - M\epsilon_0$ noben izmed neoznačenih primerov ne loči h_0 in h^*, torej tudi nobena poizvedba ne. □

Vrnimo se h izreku 4.4. *Dokaz.* Naj bo h_0 najbližja hipoteza ciljni hipotezi h^*. Če h_0 ni v nobenem koraku odstranjena in $\epsilon_0 \leq \epsilon/2$, potem:

$$S_t \subset B(h_0, 1/2^t) \subset B(h^*, 1/2^t + \epsilon_0) \subset B(h^*, 1/2^{t-1})$$

za vsak $t \leq T-1$. Tako je vsaj S_t $(\rho, 1/2^{t+1}, \tau)$-ločljiv. Izrek je dokazan z upoštevanjem vseh prejšnjih lem in izborom

$$\frac{1}{\epsilon_0} = \max\{\frac{2}{\epsilon}, O(\frac{d}{\tau\rho\sigma}\log\frac{1}{\epsilon}\log\frac{1}{\tau})\}$$

$$m = O(\frac{1}{\tau})$$

□

A.0.2 Izrek 4.5.2

V tej sekciji pokažemo dokaz, vzet iz [12], za izrek 4.5.2. Za dokaz potrebujemo naslednje 3 leme.

Lema A.4 *Če se algoritem ustavi, ima klasifikacijsko točnost manjšo od ϵ s verjetnostjo $1-\delta/2$.*

Dokaz. Vzemimo $P(\text{napaka}) > \epsilon$. Tedaj je $P(\text{V naslednjem koraku bi sprejeli učni primer}) > \epsilon$ in torej $P(t_n \text{ zapored zavrnjenih učnih primerov}) \leq (1-\epsilon)^{t_n}$

$$\begin{aligned}(1-\epsilon)^{t_n} &= (1-\epsilon)^{\frac{1}{\epsilon}\ln(\pi^2\frac{(n+1)^2}{3\delta})} \\ &< e^{-\ln(\pi^2\frac{(n+1)^2}{3\delta}} \\ &= \frac{3\delta}{\pi^2(n+1)^2} \end{aligned} \tag{A.1}$$

Prehod na drugi korak dobimo, če upoštevamo, da je $1+x < e^x$ (to lahko vidimo iz njegovega razvoja v Taylorjevo vrsto) in namesto x vstavimo $-\epsilon$

Verjetnost, da se algoritem ustavi v n-tem korakupod pogojem, da je $P(\text{napaka}) > \epsilon$ je torej manjša od $\frac{3\delta}{\pi^2(n+1)^2}$. Ocenimo še verjetnost, da se algoritem ustavi na poljubnem koraku.

$$\begin{aligned}P(\text{algoritem se ustavi na poljubne koraku}) &< \sum_n \frac{3\delta}{\pi^2(n+1)^2} \\ &= \frac{1}{4}\delta\end{aligned}$$

□

Lema A.5 *Če je pričakovan takošnji informacijski dobiček spodaj omejen s q potem velja:*

$$P_{\Delta}(\textit{komulativni inf. dobitek}(X_{I_n}, c(X_{I_n})) < \frac{qn}{2}) \leq e^{\frac{-qn}{10}}$$

Lema A.6 *Vzamimo naključen c iz razreda hipotez H s končno VC dimenzijo d. Potem, če fiksiramo X_M velja*

$$P_{\Delta}(\textit{komulativni inf. dobitek}(X_M, c(X_M)) \geq (d+1)\log\frac{\epsilon m}{d}) < \frac{d}{\epsilon m}$$

Dokaza zgornjih dveh lem lahko najdemo v [12]. Nadaljujmo z dokazom glavne trditve.

Dokaz. Navedli bomo 5 pogojev, ki zagotavljajo resničnost trditve ter dokaze, da so ti pogoji uresničeni s visoko zanesljivostjo.

1. Komulativen informacijski dobitek prvih n_0 poizvedb je večji ali enak $\frac{gn_0}{2}$
   Vzemimo nek $n_0 \geq \frac{10}{g}\ln(\frac{4}{\delta})$, potem je po lemi A.5 verjetnost pogoja enaka $1 - \frac{\delta}{4}$.

2. Komulativni informacijski dobitek prvih m_0 učnih primerov je manjši enak $(d+1)\ln(\frac{\epsilon m_0}{d})$
   Vzamimo nek $m_0 \geq \frac{4d}{\epsilon\delta}$, potem je po lemi A.6 verjetnost pogoja enaka $1 - \delta/4$.

3. Med prvimi m_0 učnimi primeri je algoritem zahteval manj kot n_0 poizvedb.
   Komulativni informacijski dobitek vseh m_0 poizvedb očitno mora biti večji ali enak komulativnemu informacijskemu dobitku n_0 učnih primerov za katere smo zahtevali poizvedbo. Ta neenakost vselej drži in skupaj s zgornjima pogojema tvorijo zahtevo: število narejenih poizvedb $< \frac{2(d+1)}{g}\ln(\frac{\epsilon}{m}{}_0 d)$ Torej za vsak $n_0 \geq \frac{2(d+1)}{g}\ln(\frac{\epsilon m_0}{d})$, zgornji pogoj drži.

4. Število zahtevanih neoznačenih primerov ne bo preseglo m_0
   Če je med prvimi m_0 primeri zaporednih t_n zavrnjenih, potem se algoritem ustavi. Opomnimo, da se pogoj t_i veča skupaj z i. Če naredimo

n poizvedb, je največje število zahtevanih neoznačenih učnih primerov enako nt_n (V najslabšem primeru imamo $t_n - 1$ zavrnjenih primerov, nato enega sprejetega za vsakega izmed n sprejetih primerov). Torej moramo imeti $m_0 \geq nt_n$. Če upoštevamo še omejitev iz tretjega pogoja in definicijo števila t_n dobimo $m_0 \geq 2\frac{(n0+1)}{\epsilon} \ln(\pi^2 \frac{(n_0+1)^2}{3\delta})$

5. Če se algoritem ustavi, potem ima z verjetnostjo $1 - \frac{\delta}{2}$, klasifikacijsko točnost boljšo od ϵ. Ta pogoj drži, kot je dokazano v lemi A.4.

Če vzamemo m_0 in n_0 iz trditve, potemo so vsi zgoraj navedeni pogoji zadoščeni s verjetnostjo $1 - \delta$ in algoritem se res ustavi ter zahteva ustrezno število neoznačenih primerov ter njihovih oznak. Tako je trditev dokazana.

□

Dodatek B

Razširitve

Ta razdelek je namenjen omembi nekaterih drugače postavljenih, težjih problemov, ki jih lahko rešujemo v duhu aktivnega učenja.

Manjkajoči atributi

V določenih problemskih okoliščinah se pojavi težava mankajočih atributov. Učni primeri so nepopolni v smislu, da vsi njihovi atributi niso znani. Primer tega so medicinski problemi, kjer o pacientu nekaj simptomov vemo, drugih ne – testiranje je recimo drago ali zamudno. Obstajajo klasifikatorji, ki poskušajo klasificirati kljub mankajočim podatkom, ampak želeli bi si povečati njihovo učinkovitost. Lahko si pomagamo z aktivno komponento, ki bi povedala, kateri atributi so na tem učnem primeru informativni. Osnovna ideja je podobna kot pri navadnem aktivnem učenju, vendar namesto da s poizvedbo zahtevamo razred, zahtevamo vrednost enega od atributov učnega primera. Aktivni učenec mora znati primerjati ceno pridobitve atributa in izboljšavo modela, ki ga tako dobi v zameno.

Različna cena označevanja

Običajno aktivno učenje predpostavlja, da je cena označevanja poljubnega učnega primera enaka in zato poskuša le zmanjšati število potrebnih oznak. Ta predpostavka v mnogih okoljih ne drži, saj so recimo v medicinski dia-

gnostiki nekateri poizkusi dražji od drugih itd. Empirična dejstva kažejo, da uporaba standardnih metod aktivnega učenja ne zmanjša nujno cene učenja in ne prinese izboljšanja glede na naključno izbiro primerov. Poleg informativnosti primera more aktivni učenec upoštevati tudi ceno, s katero ga pridobi. Eden izmed načinov, kako to doseči, je primerjava cene primera z zmanjšanjem pričakovane cene prihodnje nepravilne klasifikacije.

Šumni označevalec

Predpostavka aktivnega učenja je, da učitelj-označevalec vedno pravilno označi primer v poizvedbi. V realnem svetu to pogosto ni res zaradi človeškega faktorja, nedeterminističnosti poizkusov, premajhnega ekspertnega znanja itd. Aktivni učenec ima v takih okoliščinah možnost, da "podvomi"v pravilnost oznake, če se mu glede na trenutni model zdi neprimerna in zahteva ponovno označevanje že označenega primera. S tem seveda zapostavi nek neoznačen primer, ki bi ga sicer imel, zato mora to možnost uporabljati s previdnostjo. Da bi se spopadli s tem problemem, lahko modelerimo tudi nezanesljivost označevalca oz označevalcev, če jih je več (lahko tudi vsakega označevalca modeliramo posebej). Možnost napake označevalca nato uporabimo pri premisleku o ponovnem označevanju primera.

Ko-učenje

Ko-učenje (*co-training*) [2] je izraz za seminadzorovano obliko učenja, kjer se vzporedno učita dva klasifikatorja, vsak iz svoje podmnožice atributov učnih primerov. Prvi klasifikator klasificira učne primere v neoznačeni množici ter najbolj zanesljivo klasificirane med njimi vstavi v učno množico drugega klasifikatorja. Nato obratno naredi drugi klasifikator in postopek se ponavlja. Težava nastane, kadar se postopek ponavlja predolgo, saj je zanesljivost klasifikatorja v vsaki iteraciji manjša. Težavo lahko odpravimo z dodano aktivno komponento sistema, ki nekatere klasificirane primere da učitelju v vpogled in popravo oznake, če je to potrebno. S tem drži učno množico relativno pravilno klasificirano in kvaliteta klasifikatorja ne pada.

Dodatek C

Drugačne poizvedbe

Skozi delo smo pod pojmom poizvedbe imeli v mislih označevanje izbranega učnega primera. V tem razdelku pokažemo drugačne tipe poizvedb, kot jih običajno dovoljujemo pri aktivnem učenju in pri tem bomo povzeli [1] začetno in temeljno delo pri uporabi poizvedb. V teoretičnem poglavju smo videli omejitve navadnih poizvedb, sedaj si pa poglejmo moč splošnejšega tipa in nekatere primere, kjer bi le te bile uporabne.

Imejmo množico hipotez $H = \{h_1, h_2, .., h_N\}$ in problem binarne klasifikacije (možna razreda sta 0 in 1). Alternativno si lahko h_i predstavljamo kot množico vseh učnih primerov x, za katere je $h_i(x) = 1$. Ta predstava je pomembna in bo uporabljena nadaljnje v tem razdelku. Naloga učnega algoritma se je naučiti ciljne hipoteze, ki jo označimo s h^* (hipotezo želimo natančno identificirati). Učnim primerom, za katere velja $h^*(x) = 1$, rečemo pozitivni, ostalim rečemo negativni učni primeri. Naj bo množica vseh učnih primerov poimenovana X.

Algoritem naj ima na voljo več tipov poizvedb:

- Poizvedba vsebovanosti (*Membership query*)
  Ali $x \in h^*$? Odgovor je oblike Da/Ne in je enakovreden klasifikaciji učnega primera x. Gre torej za klasične poizvedbe, uporabljene v ostalem delu.

- Poizvedba ekvivalence (*Equivalence query*)

$h = h^*$? Ali je hipoteza v poizvedbi enaka ciljni hipotezi? V primeru, da enakost drži, poizvedba vrne Da, sicer vrne Ne in protiprimer x. Protiprimer je poljubno izbran izmed $x \in h \oplus h^*$. V analizi uspešnosti algoritma navadno predpostavimo najslabši primer: vrnjen protiprimer bo izmed vseh možnih algoritmu najmanj koristil.

- Posplošena poizvedba ekvivalence
  Poizvedba ekvivalence, kjer kot hipotezo lahko izberemo poljubno podmnožico prostora učnih primerov X. Ta množica ne rabi ustrezati nobeni hipoteze v H, zato je poizvedba močnejša od prejšnje.

- Poizvedbi podmnožice/nadmnožice (*Suberset/Superset query*)
  $h \supseteq h^*$? $h \subseteq h^*$? Poizvedba pove ali je hipoteza pod/nadmnožica ciljnega koncepta oziramo vrne protiprimer v nasprotnem.

- Poizvedba disjunktnosti (*Disjointness query*)
  $h \cap h^* = \{\}$? Poizvedba pove, če sta hipotezi disjunktni, oziroma vrne protiprimer $x \in h \cap h^*$ v nasprotnem.

- Poizvedba celovitosti (*Exhaustiness query*)
  $h \cup h^* = X$? V primeru odgovora Ne, je vrnjen protiprimer $x \notin h \cup h^*$

Najenostavneši algoritem, ki zagotavljeno vrne pravilno hipotezo, je izčrpno preizkovanje. Ta algoritem zaporedoma ustvarja poizvedbe ekvivalence za vsako možno hipotezo h_i iz H. Očitno bo v najsabšem primeru moral narediti $|H| = N$ poizvedb.

Kljub naivnosti izčpnega preizkovanja je ta algoritem v nekaterih primerih najboljše, kar lahko dosežemo. Vzemimo za primer $H = \{\{x_1\}, \{x_2\}...\}$, množico vseh singeltonov. Torej izmed vseh učnih primerov je le en pozitiven. Ni težko videti, da lahko s poljubno poizvedbo ekvivalence ali vsebovanosti v vsaki iteraciji iz množice vseh konsistentnih hipotez izločimo le eno, saj bo v najslabšem primeru odgovor na katerokoli poizvedbo Ne. To opazko lahko posplošimo z naslednjim izrekom.

Izrek C.1 *Naj obstaja $h_\cap$ za katerega velja $h_i \cap h_j = h_\cap$ za vsak različen i in j. Naj velja še da $h_\cap$ ni element X. Potem mora kateri koli algoritem, ki zna poiskati pravilni koncept h^*, v najslabšem primeru opraviti vsaj $|H| - 1$ poizvedb vsebovanosti oziroma ekvivalence.*

Dokaz. Imejmo nasprotnika, ki lahko poljubno izbira ciljno hipotezo h^*. Ciljno hipotezo lahko spreminja tudi med učenjem samim, na poljubnega izmed konsistentnih hipotez. Algoritem lahko uspešno identificira h^*, le kadar obstaja le še ena konsistenta hipoteza. Nasprotnik zato na hipoteze odgovarja tako, da učni algoritem lahko izloči kar se da malo konsistentih hipotez. Takšen nasprotnik simulira najslabši scenarij za dan učni algoritem.

Če algoritem zahteva poizvedbo vsebovanosti za x, ki je element $h_\cap$, nasprotnik odgovori z Da, v nasprotnem odgovori z Ne. V primeru odgovora Da, algoritem lahko izloči samo eno hipotezo, saj če x ne bi bil element dveh hipotez h_i in h_j , potem x ne bi niti bil element $h_\cap$. Prav tako v primeru odgovora Ne, algoritem nemore izločiti več kot ene hipoteze, saj če bi bil x element dveh hipotez h_i in h_j , bi bil tudi element $h_\cap$.

Če algoritem zahteva poizvedbo ekvivalence, nasprotnik odgovori s Ne (razen v primeru ko je možna samo ena hipoteza), za protiprimer vrne poljuben element $h \oplus h_\cap$. Tudi v tem primeru, lahko algoritem zavrže le eno hipotezo.

Vidimo, da v vsakem primeru algoritem mora narediti $|H| - 1$ poizvedb. □

Dualen dokaz obstaja tudi za unijo.

Prejšnji izrek predstavlja omejitev za učenje s poizvedbami vsebovanosti in ekvivalenci, vendar lahko dobimo boljši rezultat, če dovolimo tudi posplošne poizvedbe ekvivalence.

Izrek C.2 *Vsako končno domeno H se lahko naučimo s največ* $\log_2 |H|$ *poizvedbami.*

Oglejmo si Prepolovični Algoritem, ki za vhod jemlje H:

function PREPOLOVIČNI ALGORITEM(H)

if $H = \{L\}$ **then**
return L
else
$M_H = \{x|$ kjer je $x \in L$ za vsaj $|H|/2$ hipotez $\}$
naredi posplošeno poizvedbo ekvivalence z M_H
if odgovor Da **then**
return M_H
else
naj bo x vrnjen protiprimer
if $x \in M_H$ **then**
$H' = H$–$\{L \in H | x \in L\}$. Velja $|H|/2 \leq |H'|$
else
$H' = \{L \in H | x \in L\}$. Velja $|H|/2 \leq |H'|$ saj x ni element M_H
end if
return Prepolovični algoritem(H')
end if
end if
end function

Prepolovični algoritem v vsaki iteraciji prepolovi domeno H in tako konča v $\log_2 |H|$ iteracijah in tako dokaže zgornji izrek.

Opomba 1: Zgled s domeno vseh singeltonov se zaključi v drugi iteraciji, saj je M prazna, vrnjen protiprimer je ravno iskani singelton.

Opomba 2: Najti množico M_H je netrivialno in pogosto se jo da izračunati le v eksponentnem času. Naslednji zgled prikazuje domeno, kjer je mogoče M indirektno izračunati.

Z posplošenimi poizvedbami ekvivalence lahko tako v teoriji uspešno rešimo vsak učni problem s končnim številom hipotez. Na žalost v realnih problemih navadno ne moremo pričakovati označevalca, ki bi znal zanesljivo odgovarjati na poizvedbe uvedene v tem poglavju; slednje še posebaj velja za posplošene poizvedbe ekvivalence. Kljub temu si poglejmo nekatere učne

probleme, ki bi jih znali uspešno rešiti, če bi imeli ustreznega označevalca.

Zgled: k-CNF izrazi (*conjuctive normal form*)

k-CNF, so boolovi izrazi oblike $T_1 \wedge T_2 \wedge ... \wedge T_n$, kjer so T_i členi disjunkcija največ k spremenljivk. Spremenljivke lahko nastopajo v pozitivni ali negirani obliki. $T_i = x_1 \vee x_2 \vee \ldots \vee x_k$. V množici hipotez so vsi možni k-CNF izrazi, naloga učnega algoritma je identificirati pravega.

Algoritem inicializira začetno hipotezo h kot konjunkcijo vseh možnih členov T_i . Teh ni več kot $(2n + 1)k$, kar je polinomsko mnogo glede na n (k je v danem problemu le konstanta). V vsaki iteraciji naredi poizvedbo ekvivalence s trenutno hipotezo. V primeru odgovora Da, učni algoritem zaključi in vrne hipotezo, sicer dobi protiprimer a. Z $T_i(a)$ označimo pravilnost izjave T_i glede na vrednosti spremenljivk določene z a. V vsakem koraku bo veljalo $h(a) = 1 \Rightarrow L^*(a) = 1$, zato bo tudi za vsak protiprimer veljalo $h(a) = 0$ in $h^*(a) = 1$. Učni algoritem nato odstrani vse člene T_i za katere velja $T_i(a) = 0$, saj ti členi zagotovo niso v ciljnem konceptu h^*. V vsakem koraku se tako odstrani vsaj en člen iz hipoteze h in se zato algoritem ustavi po največ $O(nk)$ korakih.

Pokazali bomo, da zgornji algoritem v vsakem koraku indirektno tvori množico M_H in tako vsaj prepolovi prostor konsistentnih hipotez. Prostor konsistentnih hipotez sestavljajo konjunkcije neke pomnožice vseh ne odstranjenih členov. Vzemimo poljuben $a \in X$. Če je $h(a) = 1$, potem je $h'(a) = 1$ za vsak h' element konsistentnih hipotez in kriterij za množico M_H drži. Če je $h(a) = 0$, potem je $T_i(a) = 0$ za nek člen T_i v h. Za vsak h', za katerega velja $h'(a) = 1$ obstaja v h'', za katerega velja $h'(a) = 0$. $h'' = h' \wedge T_i$. Torej a ni v večini hipotez in zato ni v M_H.

Poizvedbe nadmnožice in podmnožice

Poizvedbe podmnožice oziroma nadmnožice so lahko uporabne v domeni izrazov CNF in DNF, zaradi njune dobre koordinacije z operatorjema konjunkcije in disjunkcije. V tem sklopu bomo predstavili uporabo poizvedbe nadmnožice

v domeni vzorčnih jezikov. V tem primeru protiprimera ne bomo potrebovali, zato lahko uporabimo šibkejšo poizvedbo ki vrne le Da/Ne.

Definirajmo A kot končno abecedo konstantnih simbolov, X kot števno neskončno abecedo spremenljivk ter vzorec p kot niz znakov iz A in X. Definirajmo vzorčni jezik $L(p)$ kot vse nize znakov iz A, ki jih lahko dobimo tako, da zamenjamo spremenljivke vp z nizi konstant iz A. Na primer, če je $p = 12x3y4x$ potem sta 1253645 in 12773999477 del te abecede.

Vzemimo da je p, ki ga iščemo dolžine n. $L(p)$ je torej podmnožica vseh nizov dolžine n in več. $L(x_1, x_2, ...x_i)$ predstavlja vse nize dolžine i in več. Torej je $L(x_1, x_2, ...x_i)$ nadmnožica $L(p)$ za vsak $i \leq n$. Z zaporednim klicanjem poizvedb nadmnožice z $L(x_1)$,$L(x_1, x_2)$, $L(x_1, x_2, x_3)$... lahko identificiramo dolžino niza p (poizvedba bo vračala Da do $n + 1$ poizvedbe). Naslednji korak je identifikacija mesta konstant. Zaporedno kličemo $L(x_1, x_2, ...x_{i-1}, a, x_{i+1}...x_n)$ za vsako mesto $0 \leq i \leq 1$. Za tista mesta i kjer poizvedba vrne Da, vemo da so konstanta. Vse kar preostane je ugotoviti, katere spremenljive so med seboj enake. Vsak par spremenljivk tesitramo posebej npr. z $L(x_1, x_1, ...x_n)$, pozitiven odgovor potrdi enakost spremenljivk na teh dveh mestih.

Težji zgled

Izkaže se, da nam v določenih primerih noben tip oziroma kombinacija omenjenih poizvedb (z izjemo posplošene ekvivalenčne poizvedbe) ne pomaga, da bi ciljni koncept našli v manj kot $|H| - 1$ poizvedbah v najslabšem primeru. Omenimo en tak primer. Imejmo $X = \{x_1, x_2, ...x_n\}$, $Y = \{y_1, y_2, ...y_n\}$, ter elementa z_1,z_2. Naj bo $H = \{H_1, H_2, ..., H_n\}$, kjer je $H_j = z_1, x_j \cup (Y - y_j)$. Ni težko pokazati, da ne glede na tip poizvedbe ne moremo iz prostora konsistentnih hipotez izločiti več kot eno na poizvedbo.

Dodatek D

Izdelava kriterijev specifičnih za SVM

V tem razdeleku predstavimo nekaj sorodnih kriterijev za "informativnost" učnega primera, ki so specifični za metodo učenja SVM ter utemeljimo njihovo izbiro.

Delovanje metode SVM

Iz linearne algebre vemo, da je hiperavnino najlažje opisati s predpisom: $w^T x = 0$ natanko tedaj kadar vektor x leži na premici. Hiperravnina je torej natančno določena s vektorjem w, ki je geometrisko gledano pravokoten na hiperravnino in po dogovoru normaliziran. Izkaže se tudi da velja $w^T x > 0$ kadar so točke na eni strani hiperravnine in $w^T x < 0$ kadar so na drugi, kar omogoča enostavno klasifikacijo primerov kadar je w znan. Velja tudi, da je $|w^T x|$ razdalja od točke do hiperravnine. Razdalji do najbližje točke med podatki rečemo rob. Naloga učnega algoritma je najti parametre za tak w, da bo pripadajoča hiperravnina pravilno klasificirala vse primere in bo rob največji možen. Večino (realnih) problemov ne moremo rešiti z linearnim klasifikatorjem, kar metoda SVM rešuje tako, da atribute originalnega učnega primera preslika v nek visoko dimenzionalen prostor. Marsikateri problem lahko nato v tem večjem prostoru linearno ločimo, vendar za to nimamo zagotovila. Ta razdelek bo zaradi preprostosti predpostavil da je problem

linearno ločljiv, vendar z nekaj popravki glavne ideje tudi v nasprotnem ostanejo enake.

Z X označimo prostor originalnih atributov učnih primerov, z F visoko dimenzionalen prostor dobljen iz originalnih atributov, z $\Phi(x)$ preslikavo originalnih atributov v nove, z W prostor vektorjev w, z $H = \{f|f(x) = w^T\Phi(x), w \in W\}$ vse hipoteze, z x_i i-ti učni primer in nazadnje z y_i oznako i-tega učnega primera. Naj bodo možni razredi 1 in -1.

Opazimo, da je primer pravilno klasificiran če je $y_i > 0$ in hkrati $f(x_i) > 0$ ali če je $y_i < 0$ in hkrati $f(x_i) < 0$. Tako lahko vse konsistentne hipoteze zapišemo bolj kompaktno:

$$V = \{f|f \in H, y_i f(x_i) > 0, \forall i\}$$

Hipoteza je odvisna samo od vektorja w, zato lahko V definiramo drugače:

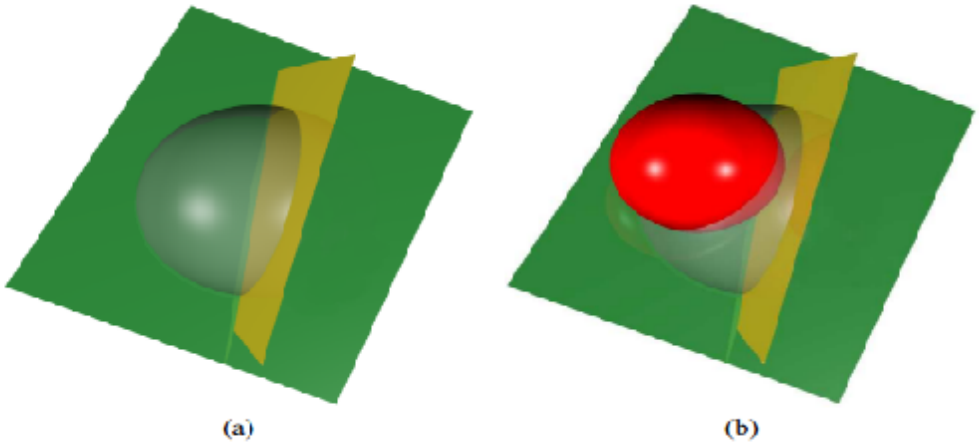
$$V = \{w|w \in W, ||w|| = 1, y_i w^T \Phi(x_i) > 0\}$$

Konsistentne hipoteze torej ležijo na enotski krožnici v prostoru W. Označimo še $Area(V)$ kot površino na enotski hipersferi, ki jo zaseda množica W.

Zaradi simetričnosti izraza $w^T\Phi(x_i)$ obstaja geometrijska dualnost med prostoroma V in F. Vsak w po definiciji predstavlja hiperravnino, ki loči prostor F na dva dela. Vendar zaradi dualnosti tudi vsak učni primer x_i linearno loči prostor hipotez, pri čemer hipoteze na eni strani hiperravnine postanejo nekonsistentne (napačno ločijo primer x_i). Na prostor hipotez lahko tako gledamo kot na hipersfero, ki jo seka n hiperravnin. Košček hipersfere, ki zadosti omejitvam vseh n hiperravnin je ravno prostor V.

Izdelava kriterijev

Za implementacijo aktivne komponente moramo najprej izbrati pristop: kako definirati in oceniti „informativnost“ učnega primera. Najpreprostejši način je z uporabo paradigme najmanjše zanesljivosti. Metoda SVM nam sicer ne vrne dejanske verjetnosti pravilnosti napovedi, vendar lahko zanesljivosti med seboj primerjamo glede na razdaljo primera do hiperravnine. Učni primer, ki je bližji ločujoči hiperravnini, je klasificiran z manjšo zanesljivostjo.



Slika D.1: Na sliki a.) je prostor konsistentnih hipotez, kadar imamo le en učni primer. Vsaka točka na tej hipersferi predstavlja w, ki pravilno loči prostor F, vendar metoda SVM išče takega z največjim robom. Denimo, da imamo za vsako točko v V kroglo, ki ima v tej točki središče in radij kar se da velik, brez da bi krogla sekala kakšno omejujočo hiperravnino. Izkaže se, da točka z največjo pripadajočo kroglo predstavlja ravno hipotezo z največjim robom. Slika b.) prikazuje prostor konnsistentnih hipotez, kjer je za eno točko označena pripadajoča krogla. Slika je vzeta iz [28].

Poizvedbo naredimo na tistem neoznačenem učnem primeru, ki je hiperavnini najbližje. Formalno:

$$X^* = \min_x y_i w^T \Phi(x_i)$$

Geometrijski razmislek o prostoru konsistentnih hipotez nam pokaže, da je le ta zvezen in njegova površina je konveksne oblike. Če nam uspe ta prostor dovolj zmanjšati, potem ne le da bo množica možnosti za izbiro pravilne hipoteze majhna, ampak tudi njeni elementi si bodo zelo podobni. Možne meje si tedaj lahko predstavljamo kot šop skoraj vzporednih premic. To načelo nam pravi, da moramo poizvedbe izbrati tako, da bo končni prostor V čim manjši.

Označimo s V_{i+} prostor, ki ga dobimo če v i-ti iteraciji označimo nek učni primer s 1. Formalno je $V_{i+} = V_i \cap \{w \in W | w^T \Phi(x) > 0\}$. Podobno definiramo V_{i-}, če v i-ti iteraciji označimo promer s -1. Z l^* označimo učenca, ki v vsaki iteraciji izbere poizvedbo da razpolovi trenutni prostor V in z V_i^* označimo tako dobljen prostor v i-ti iteraciji.

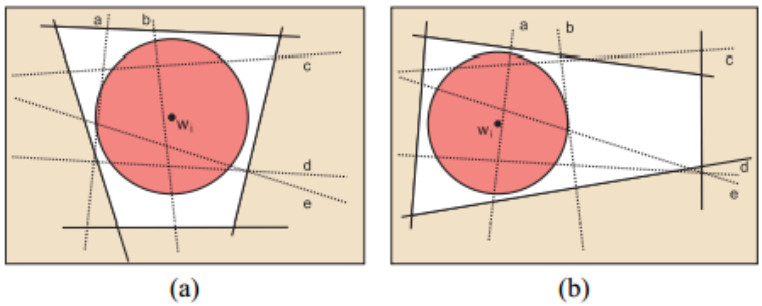
Za vsak i velja: $\sup_P E_P[Area(V_i^*)] \leq \sup_P E_P[Area(V_i)]$, kjer je P množica vseh možnih pogojnih porazdelitev y glede na x. Trditev pravi, da bo v najslabšem primeru, najbolši učenec ta, ki vedno razpolovi trenutni prostor V. Ni težko videti, da trditev drži: leva stran neenačbe se v vsakem koraku zmanjša za faktor 0.5, medtem ko se desna za faktor, ki je večji ali enak 0.5, saj bo supremum izbral večjo stran prostora.

Želimo torej da aktivni učenec v vsakem koraku učenja poišče poizvedbo, ki prepolovi prostor V, oziroma poskuša temu priti čim bližje. Izračun natančne velikosti V_{i+} in V_{i-} je v praksi zelo težak problem, zato se mu bomo izognili z uporabo približkov. Glede na način pridobivanja približkov bomo razvili tri metode: Enostavni rob (*Simple Margin*), MaxMin rob (*MaxMin Margin*), Razmerni rob (*Ratio Margin*).

Vektor w_i dobljen iz podatkov v i-ti iteraciji je tisti, ki mu v prostoru V_i pripada največja krogla (glej sliko D.1). Njegova pozicija v prostoru V_i je odvisna od same oblike prostora, vendar zaradi geometrijsko ugosdnih lastnosti prostora V, leži blizu središča prostora. Prostor želimo razpoloviti, zato iščemo hiperravnino ki gre karseda blizu središču prostora. Za vsak učni primer lahko izračunamo njegovo pripadajočo hiperravnino in opazujemo koliko blizu je le ta od w_i , torej od naše ocene središča prostora. Razdaljo lahko hitro izračunamo z $||w_i^T \Phi(x)||$, nato izberemo učni primer z najmanjšo. To metodo poimenujemo Enostavni rob. V dvorazrednem primeru, ki smo ga sedaj obravnavali, se ta metoda ujema z metodo najmanjše zanesljivosti.

Metoda enostavnega roba deluje pod predpostavko, da je prostor relativno simetričen in je zato vektor w_i centralno postavljen. Kadar ta predpostavka ni izpolnjena je tudi ocena središča slaba.

Namesto ocene središča lahko poskusimo narediti oceno velikosti prostora



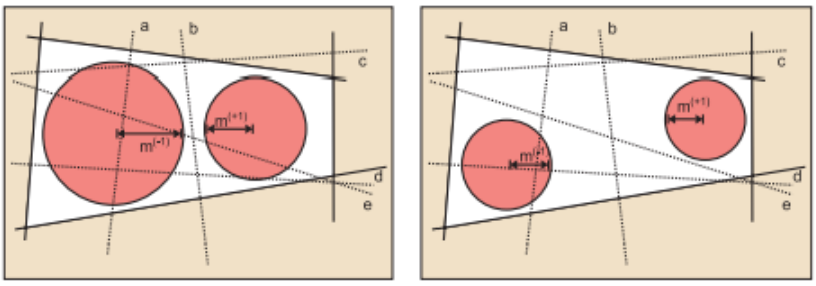
Slika D.2: Na sliki sta narisana dva primera prostora V_i, vektor w_i in njegova pripadajoča krogla. Gre za tridimenzionalni primer, kjer je relevanten kos krogle sploščen v ravnino. Metoda enostavnega roba bo v levem primeru izbrala učni primer b, v desnem učni primer a. Ta dva učna primera sta najbližje ocenjenim središčem in posledično najbolje razpolovita prostor V_i. Slika je vzeta iz [28].

V_i z velikostjo pripadajoče krogle vektorja w_i – označimo s m_i. Ocena pravi, da je prostor z večjo kroglo večji. Prostora V_{i+1} navidez ne moremo vnaprej oceniti, saj ne vemo prave oznake učnega primera. Ocenimo pa lahko prostora V_{i+} in V_{i-}. Želimo imeti čimbolj podobna V_{i+} in V_{i-}, zato kot kriterij vzamemo $\min(m_{i+}, m_{i-})$, ter iščemo učni primer, ki ima ta izraz karseda visok.

$$x^* = max_x min(m_{i+}, m_{i-})$$

To metodo poimenujemo MaxMin rob. Vsak učni primer posebej moramo torej najprej označiti s 1 in izračunati pripadajoč model in nato vsakega posebej označiti s -1 in dobiti pripadajoč model. Z n učnih primerov potrebujemo gradnjo $2n$ modelov, kar je računsko precej počasno.

Tretja pot je razmerni rob: $\min(\frac{m_+}{m_-}, \frac{m_-}{m_+})$ Gre za podoben kriterj kot MaxMin, vendar gledamo razmerje med krogi. S tem se izognemo slabi oceni v primeru zelo podolgovatega prostora.



Slika D.3: Skica kriterija MaxMin roba. Na levi sliki je bil za poizvedbo izbran učni primer b, narisana sta pripadajoča kroga po oznaki primera z 1 in z -1. Na desni sliki je izbran učni primer e. Slika je vzeta iz [28].

Razširitev na večrazredno klasifikacijo

Dobljene kriterije želimo sedaj razširiti iz binarne na večrazredno klasifikacijo. SVM je sicer binarni klasifikator, vendar lahko za večrazredno klasifikacijo naredimo več klasifikatorjev - za vsak razred enega, po principu "en proti vsem". Tako torej prvi klasifikator f_1 pove ali učni primer pripada prvemu razredu, f_2 ali pripada drugemu itd...

V binarnem primeru smo poskušali minizirati $Area(V)$ enega klasifikatorja, sedaj pa želimo da bi bile velikosti V vseh klasifikatorjev čim manjše. Minimizirati moramo izraz: $\prod_i Area(V^{(i)})$. Ker računamo ta izraz v najslabšem primeru, minimiziramo izraz po vseh učnih primerih x:

$$max_y \prod Area(V_{x,y}^{(i)})$$

Z to definicijo, ni težko razširiti kriterijev MaxMin rob in Razmerni rob.

Izraz $Area(V_{x,y}^{(i)})$ bomo poskušali še oceniti s kriterijem Enostavnega roba. Vzemimo, da je prava oznaka primera x enaka i. Če je $f_i(x) = 0$ potem gre hiperravnina čez središče krogle in približno razpolovi prostor V. Če je $f_i(x) = 1$, potem se hiperavnina krogle ravno dotika in prostor ostane enak. Simetrično se v primeru $f_i(x) = -1$, vendar tukaj hiperavnina odreže celotni

prostor V. To obnašanje lahko povzamemo z:

$$Area(V_{x,y}^{(i)}) \approx \frac{1 + f_i(x)}{2} Area(V^{(i)})$$

Simetrično, če x ni klasificiran v razred i imamo:

$$Area(V_{x,y}^{(i)}) \approx \frac{1 - f_i(x)}{2} Area(V^{(i)})$$

Z temi ocenimi lahko izračunamo zgornji min max in najdemo najboljši primer za poizvedbo.

Literatura

[1] D. Angluin, "Queries and concept learning", Machine Learning, 2:319–342, 1988.

[2] A. Blum, T. Mitchell, "Combining labeled and unlabeled data with co-training", In Proceedings of the Conference on Learning Theory (COLT), str. 92–100, Morgan Kaufmann, 1998.

[3] A. Blumer, A. Ehrenfeucht, D. Haussler, M. K. Warmuth, "Learnability and the Vapnik-Chervonenkis dimension", Journal of the Association for Computing Machinery, 36(4):929-965, October 1989.

[4] L. Breiman, "Bagging predictors", Machine Learning, 24(2):123-140, 2001.

[5] D. Cohn, L. Atlas, R. Ladner, "Improving generalization with active learning", Machine Learning, 15(2):201–221, 1994.

[6] (2011) S. Dasgupta, "Two faces of active learning". Dostopno na: http://cseweb.ucsd.edu/ dasgupta/papers/twoface.pdf

[7] S. Dasgupta and D.J. Hsu, "Hierarchical sampling for active learning", In International Conference on Machine Learning, 2008.

[8] S. Dasgupta, D.J. Hsu, and C. Monteleoni, "A general agnostic active learning algorithm", In Neural Information Processing Systems, 2007.

[9] (2006) S. Dasgupta, "Coarse sample complexity bounds for active learning". Dostopno na: http://cseweb.ucsd.edu/ dasgupta/papers/sample.pdf

[10] B. Eisenberg, R. L. Rivest, "On the sample complexity of pac-learning using random and chosen examples", In Proceedings of the 1990 Workshop on Computational Learning Theory, str. 154–162, 1990.

[11] (1996) Y. Freund, R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". Dostopno na: http://www.face-rec.org/algorithms/Boosting-Ensemble/decision-theoretic_generalization.pdf.

[12] Y. Freund and H.S. Seung and E. Shamir and N. Tishby, "Selective samping using the query by committee algorithm", Machine Learning, 28:133–168, 1997.

[13] (2009) S. Hanneke, "Theoretical Foundations of Active Learning". Dostopno na: http://reports-archive.adm.cs.cmu.edu/anon/ml2009/CMU-ML-09-106.pdf

[14] D. Lewis, W. Gale, "A sequential algorithm for training text classifiers", In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, str. 3–12. ACM/Springer, 1994.

[15] I. Kononenko, M. Robnik Šikonja, "Inteligentni sistemi", Ljubljana: Založba FE in FRI, 2010.

[16] C. Körner, S. Wrobel, "Multi-class ensemble-based active learning", Proceedings of The 17th European Conference on Machine Learning and the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, 687–694. Berlin, Germany: Springer-Verlag, 2006.

[17] S. Kullback, R. A. Leibler, "On Information and Sufficiency", Annals of Mathematical Statistics 22 (1): 79–86, 1951.

[18] K. Lang, E. Baum, "Query learning can work poorly when a human oracle is used", In Proceedings of the IEEE International Joint Conference on Neural Networks, str. 335–340, IEEE Press, 1992.

[19] D. Lewis, W. Gale, "A sequential algorithm for training text classifiers"In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, str. 3–12. ACM/Springer, 1994.

[20] F. Olsson, "A literature survey of active machine learning in the context of natural language processing", Technical Report T2009:06, Swedish Institute of Computer Science, 2009.

[21] A.I. Schein in L.H. Ungar, "Active learning for logistic regression: An evaluation", Machine Learning, 68(3):235–265, 2007.

[22] N. Roy, A. McCallum, "Toward optimal active learning through sampling estimation of error reduction", In Proceedings of the International Conference on Machine Learning (ICML), str. 441–448, Morgan Kaufmann, 2001.

[23] B. Settles, "Active Learning Literature Survey", Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[24] B. Settles and M. Craven, "An analysis of active learning strategies for sequencel abeling tasks, In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), str. 1069–1078, ACL Press, 2008.

[25] B. Settles, M. Craven, S. Ray, "Multiple-instance active learning", In Advances in Neural Information Processing Systems (NIPS), volume 20, str. 1289–1296, MIT Press, 2008.

[26] H.S. Seung, M. Opper, H. Sompolinsky, "Query by committee", In Proceedings of the ACM Workshop on Computational Learning Theory, str. 287–294,1992.

[27] C. E. Shannon, W. Weaver, "The Mathematical Theory of Communications", Urbana: The University of Illinois Press, 1994.

[28] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification", In Proceedings of the International Conference on Machine Learning (ICML), str. 999–1006, Morgan Kaufmann, 2000.

[29] V. Vapnik "The nature of statistical learning theory", Springer Verlag, 2nd edition, 2000.

[30] V.N. Vapnik and A. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities", Theory of Probability and Its Applications, 16:264–280, 1971.