

Ognjen Joldžić
Dino Kosić

MAŠINSKO UČENJE

Akadska misao, Beograd
Univerzitet u Banjoj Luci

Prof. dr Ognjen Joldžić
Prof. dr Dino Kosić

MAŠINSKO UČENJE

Recenzenti:
Dr Momir Ćelić
Dr Mladen Veletić

Izdavači
Akademska misao, Beograd
Univerzitet u Banjoj Luci – Elektrotehnički fakultet, Banja Luka

Dizajn naslovne strane
Borislav Maljenović

Štampa
Akademska misao, Beograd

Tiraž
200 primeraka

ISBN 978-86-7466-838-2

Mesto i godina izdanja: Beograd, Banja Luka, 2020.

SADRŽAJ

1	OSNOVNI KONCEPTI MAŠINSKOG UČENJA	1
1.1	Kognitivni procesi u ljudskom mozgu	6
1.2	Osnovni pojmovi mašinskog učenja	9
1.3	Istorijski razvoj mašinskog učenja	14
1.4	Organizacija poglavlja knjige	17
2	REGRESIJA	19
2.1	Osnovne osobine regresione analize	19
2.2	Opšti postupak određivanja regresione funkcije	24
2.3	Iterativne metode regresije	30
2.3.1	Metoda gradijenta za određivanje minimuma funkcije	31
2.3.2	Njutnova metoda za određivanje minimuma funkcije	38
2.3.3	Regresiono stablo odlučivanja	44
2.4	Neiterativne metode regresije	53
2.4.1	Regresija korištenjem metode normalnih jednačina	53
2.5	Implementacija regresionih funkcija	57
2.5.1	Regresija sa jednom promjenljivom	57
2.5.2	Regresija sa više promjenljivih	65
2.6	Validacija regresionih funkcija	69
2.7	Zaključak	71
3	KLASIFIKACIJA	75
3.1	Osnovne osobine algoritama za klasifikaciju	75
3.1.1	Mjere ispravnosti klasifikacije	78
3.2	Logistička regresija	91
3.2.1	Funkcije greške logističke regresije	92
3.2.2	Metoda gradijentnog opadanja za logističku regresiju	99
3.3	Analiza mašinskog učenja logističkom regresijom	102
3.3.1	Logistička regresija sa jednom promjenljivom	102
3.4	Bajesov klasifikator	111
3.5	Klasifikacija na osnovu k najbližih susjeda	121
3.5.1	Pojam i načini određivanja distance	122

3.5.2	Uticaj parametara na stabilnost kNN algoritma	126
3.6	Klasifikaciono stablo odlučivanja	127
3.6.1	Algoritam za formiranje stabla odlučivanja	129
3.7	Šuma nasumičnih stabala	147
3.8	Pojačavanje klasifikacionih algoritama	151
3.8.1	AdaBoost algoritam	152
3.8.2	Algoritmi za gradijentno pojačavanje	159
3.9	Zaključak	163
4	GRUPISANJE	169
4.1	Osnovne osobine algoritama za grupisanje	169
4.2	Mjere sličnosti tačaka i priprema vrijednosti	172
4.3	Algoritmi za hijerarhijsko grupisanje	174
4.3.1	Aglomerativno hijerarhijsko grupisanje	175
4.3.2	Partitivno hijerarhijsko grupisanje	188
4.4	Algoritmi bazirani na centroidima	189
4.5	Algoritmi bazirani na gustini tačaka	200
4.5.1	DBSCAN algoritam	200
4.6	Algoritmi bazirani na modelu	208
4.7	Validacija algoritama za grupisanje	213
4.8	Zaključak	218
5	NEURONSKE MREŽE	224
5.1	Ljudski mozak	225
5.2	Vještački neuron	228
5.2.1	Tipovi aktivacionih funkcija	229
5.3	Topologije neuronskih mreža	232
5.3.1	Neuronske mreže bez povratnih veza	232
5.3.2	Rekurentne neuronske mreže	233
5.4	Obučavanje neuronskih mreža	233
5.5	Perceptron i Adaline	235
5.5.1	Jednoslojne mreže sa funkcijama praga	235
5.5.2	Pravilo obučavanja perceptrona i teorema konvergencije	236
5.5.3	Adaptivni linearni element - Adaline	239
5.5.4	Mreže sa linearnim aktivacionim funkcijama i delta pravilo	240
5.6	Propagacija greške unazad	241
5.6.1	Višeslojne mreže bez povratnih veza	242
5.6.2	Uopšteno delta pravilo	242
5.6.3	Brzina obučavanja i momentum član	244

5.6.4	Nedostaci propagacije greške unazad	245
5.6.5	Implementacija neuronskih mreža	246
6	POTPORNI VEKTORI	255
6.1	Klasifikator na bazi granične hiperpovrši	256
6.2	Klasifikator na bazi potpornih vektora	269
6.3	Mašina potpornih vektora	272
7	REDUKCIJA BROJA DIMENZIJA	279
7.1	Transformacija skupa deskriptora	280
7.1.1	Analiza glavnih komponenti	281
7.1.2	Analiza linearnih diskriminanti	286
7.2	Selekcija deskriptora	289
7.2.1	Selekcija u nadgledanom učenju	290
7.2.2	Selekcija u nenadgledanom učenju	292
8	UČENJE PODSTICAJEM	296
8.1	Osnovne osobine učenja podsticajem	296
8.2	Definicija učenja podsticajem	300
8.3	Q-učenje	302
8.3.1	Kompletan primjer Q-učenja	303
8.4	Duboko Q-učenje	314
8.5	Zaključak	318

And now for something completely different...

PREDGOVOR

Oblast mašinskog učenja u formi u kojoj je poznata i danas postoji već dugi niz godina, ali početkom XXI vijeka doživljava značajan rast popularnosti. Ovo je dijelom uslovljeno potrebom za pronalaženje načina za obradu velikih količina podataka, ali dijelom i širokom dostupnošću i jednostavnošću upotrebe alata ove namjene.

Gotovo svi algoritmi koji se danas mogu svrstati pod okvir mašinskog učenja se mogu svesti na matematičke procedure koje su postojale mnogo prije pojave računarskih sistema. Međutim, u velikom broju knjiga koje takve algoritme analiziraju primarno kroz matematički aspekt, gubi se fokus na konkretnu primjenu i praktične probleme implementacije. Sa druge strane, za upotrebu nekog algoritma mašinskog učenja u današnje vrijeme gotovo da nije potrebno znati bilo šta osim njegove namjene, jer je najveći broj javno dostupnih implementacija napisan imajući u vidu samo jednostavnost pokretanja, istovremeno sakrivajući većinu detalja o suštinskim osobinama.

Osnovni cilj ove knjige je upravo da pokuša da odgovori na prethodne probleme. U svakom poglavlju je akcenat naročito stavljen na praktične aspekte realizacije algoritama, uz onoliko matematičke pozadine koliko je potrebno da bi se moglo govoriti o tome kako i zašto dati pristup funkcioniše i koja su njegova ograničenja u praksi. Istovremeno, date su detaljne implementacije, u najvećem broju slučajeva realizovane bez upotrebe gotovih biblioteka. Na taj način se čitaocu daje prilika da se u dovoljnoj mjeri upozna sa osobinama svakog algoritma kako bi ga mogao prilagoditi vlastitim potrebama i specifičnostima problema koji rješava.

Mišljenje je autora da je izbor programskog jezika sa funkcionalne strane potpuno nebitan za implementaciju bilo kojeg od algoritama opisanih u knjizi, iako su predrasude po pitanju prikladnosti nekih programskih jezika za mašinsko učenje duboko ukorijenjene u popularnoj literaturi. Kao ilustracija ove tvrdnje, za implementaciju primjera iz knjige osnovni kriterijum je bio kompaktnost i čitljivost napisanog programa, pa su svi algoritmi realizovani korištenjem programskog jezika Java i paketa Matlab. Korišteni skupovi podataka i implementacije algoritama su javno dostupni kao prateći materijali uz ovu knjigu¹.

¹ <https://github.com/ognjen-j/masinsko-ucenje>

Imajući u vidu sve prethodno rečeno, knjiga je namijenjena svim čitaocima koji imaju bar elementarno poznavanje osnovnih matematičkih principa (prvenstveno rada sa matricama i osnovana matematičke analize) i iskustvo u programiranju. Ovo u prvom redu uključuje studente tehničkih fakulteta, ali bilo koga ko želi da se rekreativno bavi mašinskim učenjem.

1

OSNOVNI KONCEPTI MAŠINSKOG UČENJA

*You keep using that word. I do not think it means
what you think it means.*

— Inigo Montoya

Informacione tehnologije predstavljaju jednu od najbrže rastućih oblasti ljudske djelatnosti i u današnje vrijeme obuhvataju gotovo sve aspekte života i rada ljudi. U relativno kratkom periodu, ova oblast je prešla dug put od usko definisane naučne i eksperimentalne uloge u akademskim krugovima, pa do sveprisutne pojave čak i u oblastima koje nisu direktno vezane za računarstvo. Za svo to vrijeme, naučni aspekti informacionih tehnologija se i dalje razvijaju nesmanjenom jačinom, ali se istovremeno konstantno povećava broj djelatnosti u kojima se podrazumijeva da dio zadataka koji je ranije bio obavljan od strane ljudi sad bude delegiran u manjoj ili većoj mjeri na mašine. Riječ "mašina" je ovdje upotrebljena u izuzetno širokom značenju i podrazumijeva bilo kakvu softversku ili hardversku komponentu koja je u stanju da automatizuje i pojednostavi neki proces izvršavanjem unaprijed definisanih zadataka ili čak prividnim donošenjem samostalnih odluka.

Ovakav razvoj događaja je u prvom redu moguć zbog brzog razvoja hardvera i povećanja dostupnosti tehnologije za široku upotrebu. U periodu od otprilike pola vijeka je došlo do ogromne transformacije tehnologije projektovanja računarskih sistema, pa samim tim i njihovih performansi. Od mašina koje su inicijalno zauzimale ogroman fizički prostor, a često bile slabijih mogućnosti od današnjih džepnih kalkulatora, došlo se do uređaja široke potrošnje koji su neizostavan dio svakog domaćinstva. Prosječan pametni telefon je nekoliko puta moćniji od

najmoćnijih računarskih sistema iz prethodne decenije, pa se korisnici sve češće oslanjaju na takve uređaje u organizaciji svakodnevnih zadataka. Ovakva brzina razvoja je čak i formalizovana sredinom 60-tih godina XX vijeka kroz ono što je u literaturi poznato kao Murov zakon¹, prema kojem se očekuje da će se broj tranzistora po jedinici površine udvostručavati svake 2 godine (ili 18 mjeseci, prema nekim izvorima). Od objavljivanja, Murov zakon se pokazao kao precizna procjena i čak je u velikoj mjeri oblikovao brzinu razvoja tehnologije iz ove oblasti. Iako se originalno odnosio na broj tranzistora na čipu, vrlo brzo je definicija proširena da uključuje uopštene performanse sistema.

Iako je realno očekivati da će se prije ili kasnije doći do fizičkih ograničenja za dalje važenje Murovog zakona (sam Mur je predvidio da će se to desiti u toku druge decenije XXI vijeka), to ipak ne znači da se u budućnosti očekuje stagnacija razvoja procesnih mogućnosti računarskih sistema. Naime, povećanje broja transistora (ili povećanje njihove brzine) je samo jedan oblik skaliranja procesne moći, i on je poznat pod nazivom vertikalno skaliranje. Kod vertikalnog skaliranja se i intuitivno može očekivati da postoji neko (najčešće prostorno) ograničenje po pitanju količine dostupnih resursa. Ako nije u pitanju broj tranzistora koji se može smjestiti na neku površinu, onda se vrlo često radi o ograničenjima koja nameću druge komponente jer nisu u stanju da isprate zahtjeve izuzetno brzih procesora ili o ograničenjima napajanja ili hlađenja takvih sistema itd. Dodatno, smanjivanjem veličine tranzistora, odnosno povećanjem broja transistora po jedinici površine, povećavaju se troškovi izrade i testiranja, pa i to može postati faktor koji će usporiti primjenu Murovog zakona.

Kao odgovor na prethodne probleme je razvijen koncept horizontalnog skaliranja, kod kojeg se polazi od pretpostavke da krajnjem korisniku nije od prevelikog značaja kako je fizički organizovan sistem koji obrađuje njegove zahtjeve, dok god je količina resursa zadovoljavajuća i dok god je moguće dobiti traženi odgovor u prihvatljivom vremenskom roku. Praktičnim jezikom rečeno, ako se umjesto procesora sljedeće generacije mogu iskoristiti dva procesora prethodne generacije, a da konačni rezultat bude isti, onda se dobija ekvivalentan sistem za koji nije potrebno uložiti sredstva za razvoj i testiranje novih tehnologija izrade. Naravno, ovo podrazumijeva da moraju postojati i prateći softverski elementi koji omogućavaju distribuciju nekog problema na veći broj jednakih čvorova.

¹ Gordon Moore