

Strojno učenje

- izgradnja sustava koji poboljšavaju performanse kroz iskustvo koristeći kriterijsku funkciju.
- programiranje računala da podesava parametre modela na temelju danih primjera

Model (hipoteza)

- definiran i izgrađen do razine nepoznatih parametara
- prediktivni i deskriptivni

Učenje

- učenje pravila
- nadzirano učenje (podaci (x,y) , x je ulaz, y cilj, treba naći preslikavanje $y = f(x)$)
- nenadzirano učenje (nema cilja, treba naći pravilnosti u ulaznim podacima)
- podržano učenje

Nadzirano učenje

- regresija
- klasificiranje/raspoznavanje uzoraka
- predviđanje budućih slučajeva, ekstrakcija znanja, kompresija, detekcija ekstremnih vrijednosti

Podržano učenje

- učenje strategije na temelju serije izlaza
- nema nadziranog učenja, samo odgodjena nagrada
- pr. Igranje igara, robot u labirintu

Problem učenja

- zadatak T , mjera P , iskustvo E
- program uči ako se povećava uspješnost obavljanja zadataka

Odabir načina stjecanja iskustva

- 1. Svojstvo iskustva – utjecaj iskustva na učenje
- 2. Svojstvo iskustva – stupanj u kojem učenik kontrolira smjer primjera za učenje
- 3. Svojstvo iskustva – reprezentativnost primjera za učenje

Odabir ciljne funkcije

- primjer – odabir najboljeg poteza iz skupa dozvoljenih

- alternativno – evaluacijska funkcija V , funkcija pridružuje veću numeričku vrijednost boljim stanjima

Odabir predstavljanja ciljne funkcije, Odabir algoritma za aproksimaciju ciljne funkcije

Učenje koncepata

- uobličavanje definicije općenite klase/kategorije ako je dan skup pozitivnih i negativnih primjera te kategorije
- pretraživanjem prostora potencijalnih hipoteza za onom koja najbolje odgovara primjerima za učenje

Koncept – podskup objekata (dogadjaja) nekog većeg skupa

Koncept (klasa) – funkcija koja poprima boolove vrijednosti definirana na nekom nadskupu

Učenje koncepata – izvođenje funkcije boolovih vrijednosti na temelju skupa pozitivnih i negativnih primjera

Dakle, imamo skup primjera, predvidjeti vrijednost nekog atributa na temelju ostalih. Kako predstaviti hipotezu?

Hipoteza – konjunkcija uvjeta na vrijednosti atributa. Za svaki atribut, hipoteza može sadržavati:

- $?$ - svaka je vrijednost prihvatljiva
- \emptyset - nijedna vrijednost nije prihvatljiva
- Određena vrijednost

Notacija

- X – skup primjera
- c – ciljna funkcija ili ciljni koncept, $c: X \rightarrow \{0,1\}$
- H – skup svih potencijalnih hipoteza, opisana s konjunkcijom uvjeta
- D – skup primjera za učenje $(x, c(x))$

Traži se hipoteza $h \in H$, takva da je $h(x) = c(x)$ za $\forall x \in X$

HIPOTEZA INDUKTIVNOG UCENJA – bilo koja hipoteza koja dobro aproksimira ciljnu funkciju na dovoljno velikom skupu primjera, aproksimirat će ciljnu funkciju dovoljno dobro i nad novim primjerima

Za neki $x \in X$ i hipotezu $h \in H$ kažemo da x zadovoljava h ako $h(x) = 1$

Find-S Algoritam

Find-S Algoritam:

1. Inicijaliziraj h na najspecifičniju hipotezu u H

2. Za svaki pozitivni primjer za učenje x
 - a. Za svaki atribut a_i u h
 - i. AKO je uvjet a_i zadovoljen sa x TADA ne cini nista INACE zamjeni a_i u h sa sljedećim općenitijim uvjetom koji je zadovoljen sa x
3. Predoci na izlazu h

Find-S algoritam pretpostavlja da se c nalazi u H i nema pogresaka u podacima. Algoritam ne gleda negativne primjere, te nalazi najspecifičniju hipotezu konzistentnu s pozitivnim primjerima. Algoritam garantira nalazanje najspecifičnije hipoteze, te je ona konzistentna i s negativnim primjerima uz pretpostavku da je ciljni koncept c u H i da su primjeri za učenje ispravni.

Induktivna pristranost

- Ciljni koncept c je sadržan u prostoru hipoteza H !
- Svi drugi primjeri su negativni osim ako to nije suprotno postavljeno drugim znanjem

Algoritam Eliminacije Kandidata

- nalazi sve hipoteze konzistentne s primjerima za učenje – PROSTOR INACICA (oznaka $VS_{H,D'}$)

Listaj-zatim-eliminiraj algoritam

1. Prostor inacica – lista koja sadrži sve hipoteze iz H
2. Za svaki primjer za učenje $(x, c(x))$ iz D ukloni iz prostora inacica sve hipoteze za koje je $h(x) \neq c(x)$
3. Predoci na izlazu hipoteze iz prostora inacica

Najopćenitija granica G – skup maksimalno općenitih članova skupa hipoteza H konzistentnih s primjerima za učenje D

Najspecifičnija granica S – skup maksimalno specifičnih članova skupa hipoteza H konzistentnih s primjerima za učenje D

Algoritam Eliminacije Kandidata

1. Inicijaliziraj G na skup maksimalno općenitih hipoteza u H
2. Inicijaliziraj S na skup maksimalno specifičnih hipoteza u H
3. Za svaki primjer za učenje $d \in D$ učini
 - a. Ako je d pozitivni primjer
 - i. Ukloni iz G sve hipoteze nekonzistentne s d
 - ii. Za svaku hipotezu $s \in S$ koja nije konzistentna s d
 1. Ukloni $s \in S$

2. Dodaj u S sve minimalne generalizacije h od s takve da je h konzistentna s d i neki član iz G je općenitiji od h (osigurava konzistentnost s prethodnim negativnim primjerima)
 3. Ukloni iz S sve hipoteze koje su općenitije od drugih hipoteza u S
- b. Ako je d negativni primjer
- i. Ukloni iz S sve hipoteze nekonzistentne s d
 - ii. Za svaku hipotezu $g \in G$ koja nije konzistentna s d
 1. Ukloni $g \in G$
 2. Dodaj u g sve minimalne specijalizacije h od g takve da je ha konzistentna s d i neki članovi iz S su specifičniji od h (osigurava konzistentnost s prethodnim pozitivnim primjerima)
 3. Ukloni iz G sve hipoteze specifičnije od drugih hipoteza u G

Ako je tijekom učenja S ili G postanu prazni skupovi, ciljni koncept nije moguće predstaviti u tom prostoru hipoteza.

Induktivna pristranost

- Ciljni koncept c je sadržan u prostoru hipoteza H!

Algoritam ROTE LEARNER nema induktivne pristranosti

Stabla odluke

- Neparametarska metoda
- Metoda aproksimiranja funkcije diskretnih (i realnih) vrijednosti, robusna na sum, može učiti disjunktivne koncepte
- Pretražuje potpun prostor hipoteza
- Induktivna pristranost – preferiraju se mala stabla u odnosu na velika
- skup ako-onda pravila

Opcenito stablo odluke predstavlja disjunkciju konjunkcije uvjeta na vrijednosti atributa.

ID3 (Induction of Decision Trees)

- testira se svaki atribut da se odredi koliko dobro klasificira primjere
- najbolji atribut postaje cvor, njegove vrijednosti su silazne grane
- postupak se ponavlja
- pohlepni algoritam, nikad se ne vraća zbog ponovnog razmatranja prethodnih cvorova

Informacijska dobit – mjera kako dobro pojedini atribut odjeljuje primjere za učenje u skladu s ciljnom klasifikacijom.

$$Entropija(S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_-$$

p_+ - proporcija pozitivnih primjera u S (broj pozitivnih/ukupan broj)

p_- - proporcija negativnih primjera u S

Po definiciji $0 \log_0 \equiv 0$

Informacijska dobit atributa A u odnosu na skup primjera S jest

$$Informacijska_{dobit(S,A)} = Entropija(S) - \sum_{v \in Vrijednost(A)} \frac{|S_v|}{|S|} Entropija(S_v)$$

Postupak:

1. Odrediti informacijsku dobit svakog atributa
2. Atribut s najvećom vrijednošću postaje korijen
3. Skup na učenje podijeliti na podskupove određene različitim vrijednostima atributa korijena
4. Ponoviti postupak na svakom od podskupova

ID3

- Pretražuje potpun prostor hipoteza
- Pronalazi samo jednu hipotezu
- Ne vraća se unatrag u postupku
- Koristi sve primjere za učenje u svakom koraku da bi statistički rafinirao tekucu hipotezu

Približna induktivna pristranost – preferiraju se kraća stabla odluke nad većima

Bolja približna induktivna pristranost – preferiraju se kraća stabla odluke nad većima, preferiraju se stabla koja stavljaju atribute s većom informacijskom dobiti bliže korijenu

Opcenito, ID3 gradi jednostavna stabla odluke, ali pristup koji koristi ne garantira da se bolje stablo ne može pronaći.

Ako podaci imaju sum, ili je skup za učenje premalen, može doći do prenaucenosti. Hipoteza h je prenaucena ako postoji hipoteza h_1 takva da h ima manju pogrešku od h_1 na primjerima za učenje, ali h_1 ima manju pogrešku u cijelom prostoru primjera.

Prenaucenost se izbjegava

- Zaustavljanjem rasta stabla prije savrsene klasifikacije
- Naknadnim podrezivanjem prenaucenog stabla

Razumna veličina stabla se određuje statistički, uvođenjem eksplicitne mjere kompleksnosti, uvođenjem skupa za vrednovanje, itd.

Podrezivanje stabla – uklanjanje cvora i pripadnog podstabla zamjenjujući ga s listom tako da se listu pridruži najčešća vrijednost ciljnog atributa u tom podcvoru.

Atributi s kontinuiranim vrijednostima

- Vrijednosti atributa sloze se u rastucem redoslijedu
- Odrede se susjedne vrijednosti koje se razlikuju u klasifikaciji ciljnog atributa
- Nadje se srednja vrijednost takvih vrijednosti atributa. Te srednje vrijednosti cine kandidate za granicnu vrijednost c
- Racuna se informacijska dobit za svakog kandidata
- Odabire se c s najvecom vrijednoscu I

Mjere uspjesnosti

		Stvarno Stanje	
		Pozitivan	Negativan
Predvidjeno stanje	Pozitivan	Tocno Pozitivan True Positive = TP	Lazno Pozitivan False Positive = FP
	Negativan	Lazno Negativan False Negative = FN	Tocno Negativan True Negative = TN

Ukupan broj primjera = TP + TN + FP + FN

Broj točno klasificiranih primjera = TP + TN

Broj pozitivnih primjera = TP + FN

Broj negativnih primjera = TN + FP

Tocnost (accuracy) – udio točno klasificiranih primjera u skupu svih primjera

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Preciznost (precision) – udio točno klasificiranih primjera u skupu pozitivno klasificiranih primjera

$$\frac{TP}{TP + FP}$$

Odziv (recall) – udio točno klasificiranih primjera u skupu svih pozitivnih primjera

$$\frac{TP}{TP + FN}$$

Specificnost (specificity) – udio točno klasificiranih primjera u skupu svih negativnih primjera

$$\frac{TN}{TN + FP}$$

F mjera – harmonijska sredina preciznosti i odziva

$$F_{\beta} = \frac{(1 + \beta^2) \times \text{preciznost} \times \text{odziv}}{\beta^2 \times \text{reciznost} + \text{odziv}}$$

Nadzirano učenje

-učenje koncepata (klasa, razreda) na temelju primjera

-primjer, Klasa C „obiteljski auto“

- predviđanje : Je li auto x obiteljski auto“
- izlaz: pozitivni i negativni primjeri
- ulaz: x_1 - cijena, x_2 – snaga motora
- skup za učenje $X = \{x, r\}$
- pretpostavka (na temelju prethodnog ispitivanja) o C:
 - $(p_1 \leq \text{cijena} \leq p_2) \ \&\& \ (e_1 \leq \text{snaga motora} \leq e_2)$
- prostor hipoteza H – skup kvadrata (p_1, p_2, e_1, e_2)

Problem generalizacije – kako dobro će se klasificirati budući primjeri

Vapnik – Chervonenkisova dimenzija

- N primjera se može označiti na 2^n načina kao pozitivni i negativni
- Ako postoji skup od N primjera tako da za svako označavanje postoji hipoteza h iz H koja je konzistentna s primjerima, kažemo da H razdjeljuje N primjera
- VC dimenzija skupa H je najveći N za koji H razdjeljuje N primjera

Kombiniranje strojno naučenih algoritama

1. Pregled značajnijih pristupa
 - a. Nezavisno naučeni eksperti (varijante glasanja)
 - b. Viserazinsko učenje (boostanje, kaskadiranje)
2. Postupak Viole i Jonesa

Ideja – kombiniranjem naučenih osnovnih algoritama postići sinergiju

Varijacija

- Različite grupe algoritama
- Različiti parametri istih algoritama
- Različiti modaliteti ulaza
- Različiti odabiri skupova za učenje

Notacija

- L – broj osnovnih algoritama

- K – broj razreda kod klasifikacije
- Ako algoritmi imaju po jedan izlaz
 - o $d_j(x)$ - rezultat j-tog algoritma M_j za ulaz x
 - o $y = f(d_1, d_2, \dots, d_L | \varphi)$ - konacni rezultat uz parametre φ
- Ako algoritmi imaju po K Izlaza
 - o $d_{ji}(x)$ - i-ti rezultat j-tog algoritma za ulaz x
 - o $y_i = f(d_{1i}, d_{2i}, \dots, d_{Li} | \varphi_i)$ - konacni i-ti rezultat uz parametre φ_i

Glasanje

- Nezavisni eksperti usporedni evaluiraju izlaz
- Konacni rezultat linearna kombinacija pojedinačnih osnovnih rezultata
- ECOC, Bagging, MoE, itd
- Bagging – manja nestabilnost, manja varijanca, otpornost na zasicenje

Boostanje – skup osnovnih algoritama konstruiramo tako da sljedbenici uče na greskama prethodnika

Kaskadiranje – isto kao boostanje, ali konacan rezultat ne dobivamo linearnom kombinacijom

Uvod u statisticko zakljucivanje

Statistika – deskriptivna, inferencijalna, univarijantna, bivarijantna, multivarijantna, parametarska, neparametarska

Varijable – zavisne (kriterijske), nezavisne (prediktorske)

Ciljevi statisticke analize:

- Skupljanje podataka > Manipulacija podacima > Interpretacija

Normalna distribucija

- Ocekivanje μ
- Standardna devijacija σ
- Normalna distribucija s ocekivanjem μ i varijancom σ^2 – $N(\mu, \sigma^2)$
- Normalna distribucija s ocekivanjem 0 i standardnom devijacijom 1 – U
- STANDARDIZACIJA
 - o svodjenje $X: N(\mu, \sigma^2)$ na U
 - o transformacijom $U = \frac{X - \mu}{\sigma}$

Podrzano ucenje

- Q-ucenje
- Procjenitelj (critic) ne govori unaprijed sto raditi, nego daje odgovarajuće nagrade ili kazne na kraju niza akcija

- Problem priznanja zasluge – kako odrediti koje su akcije odgovorne za konacan ishod
- Odgodjene nagrade – priznanje zasluga
- Istrazivanje – agent utjece na izbor primjera za učenje
 - o Istrazivanje nepoznatih stanja u potrazi za informacijama
 - o Iskoristavanje poznatih stanja
- Cijeloživotno učenje
- Djelomicno vidljiva stanja

Pojmovi

- Skup stanja (state) - $s \in S$
- Skup akcija (action) - $a \in A$
- Strategija (policy) - $\pi: S \rightarrow A$
- Dobrota stanja s (value) - $V(s)$
- Dobrota akcije a u stanju s (quality) - $Q(s,a)$
- Nagrada (reward) - $r(s)$ ili $r(s,a)$
- Koeficijent umanjenja nagrade - γ
- Dobrota strategije odlucivanja, pocevsi iz stanja s
 - o $V^\pi(s) = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$
- Optimalna strategija
 - o $\pi^* = \operatorname{argmax} V^\pi(s), (\forall s)$

Učenje s unaprijed poznatim modelom svijeta

- Odgodjena nagrada
- Rijesenje – dinamičko programiranje
- Izuzetno rijetki slucajevi

Učenje s iterativnim upoznavanjem svijeta

- Svijet nije u potpunosti unaprijed pozna
- Pokriva vecinu problema

Nacini učenja strategije

- Iteracija po vrijednostima, iteracijom po strategijama
- Najprikladnije - odredjivanje dobrote para (stanje, akcija) pomocu funkcije $Q(s,a)$

Q učenje

- Za svako stanje postavi vrijednost $Q(s,a)$ na nulu
- Postavi se u neko stanje s
- Ponavljaj beskonacno
 - o Odaber i izvrsi akciju a
 - o Primi nagradu r
 - o Osvjezi zapis $Q(s,a)$ prema formuli:
 - Bez stope učenja - $Q \leftarrow r(s,a) + \gamma \max_{a'} Q(s',a')$
 - stopa učenja - $Q(s,a) \leftarrow \eta(r(s,a) + \gamma \max_{a'} Q(s',a')) + (1 - \eta)Q(s,a)$

- Uzmi stanje s' kao novo stanje s
 - $\eta = \eta \times k, 0 < k < 1$

Algoritam konvergira pod određenim uvjetima

1. sustav je deterministički markovljev proces odlučivanja
2. nagrade su ograničene
3. svaki par stanje-akcija se izvede beskonačno često

Nacin osvježavanja – 1 korak, učenje unatrag, epizoda

Učenje na temelju primjera

- metoda k-najbližih susjeda
- metoda lokalne regresije s težinskim faktorima
- zaključivanje na temelju slučajeva
- radijalne bazne funkcije

Lazy vs. Eager

- lazy metode
 - odadjaju odluku o klasifikaciji do trenutka predocavanja novog primjera
 - +konstruiraju razlucitu aproksimaciju ciljne funkcije za svaki razluciti novi upit
 - +procjenjuju ciljnu funkciju samo lokalno, u okolini novog primjera
 - +pogodno za kompleksne ciljne funkcije
 - –visoka cijena klasificiranja novog primjera
 - –razmatraju se svi atributi iako samo neki mogu imati utjecaj
 - k-nn, lokalna regresija, zakljucivanje na temelju slucajeva
- eager metode
 - sve do sada iznesene (ID3)
 - radijalne bazne funkcije

Algoritam k-nn

- algoritam za učenje
 - za svaki primjer za učenje $(x, f(x))$ dodaj primjer na listu primjeri _za_učenje
- algoritam klasifikacije
 - za dani primjer x_q s nepoznatom klasifikacijom
 - neka x_1, x_2, \dots, x_k oznacavaju k primjera koji su najblizi x_q
- vrati
 - $\hat{f}(x_q) \leftarrow \underset{v \in V}{argmax} \sum_{i=1}^k \delta(v, f(x_i))$ gdje je $\delta(a, b) = 1$ ako $a = b$, 0 inace
- $f(x_q)$ je najcesca vrijednost ciljne funkcije koja se pojavljuje medju k primjera za učenje koji su najblizi upitu x_q

Regresija pomocu k-nn algoritma

- umjesto najcesce pojavljivane vrijednosti ciljne funkcije, odgovor na upit je srednja vrijednost ciljnih funkcija k najblizih susjeda
- $\hat{f}(x_q) \leftarrow \frac{1}{k} \sum_{i=1}^k f(x_i)$

Induktivna pristranost – pretpostavka da je klasifikacija upita x_q slicna klasifikaciji primjera u blizini

- Nije prednost

Grupiranje podataka

- Pridruziti objekte u grupe na temelju slicnosti
- Metrika – grupiranje objekata
- Korelacijski koeficijenti – grupiranje varijabli

Metrika

Minkovski metrika : $d(a, b) = \sqrt[n]{\sum_{i=1}^n (a_i + b_i)^n}$

- o $n = 2$ – euklidska metrika
- o $n = 1$ – manhattan ili cityblock metrika, ili u slucaju binarnih vektora, hammingova metrika
- o $n \rightarrow \infty$ – cebisevljeva udaljenost $d(a, b) = \max_{1 \leq i \leq n} \{|a_i - b_i|\}$

Mahalanobisova udaljenost

- $d(x, y) = \sqrt{(x - y)' \Sigma^{-1} (x - y)}$
- Pozitivno definitna kvadratna forma oblika $x'Ax$ gdje je $A = \Sigma^{-1}$ poopcenje euklidske udaljenosti ako varijable imaju razlicite standardne devijacije i korelirane su

Vrste grupiranja

- Particijska
- Hijerarhijska – graficki prikazano dendogramom

Metode povezivanja

- Single linkage – minimalna udaljenost
- Averagelinkage – srednja udaljenost
- Maksimalna udaljenost

K-means algoritam

- Particija objekata, ne varijabli
- Odabere se k pocetnih centara grupa (centroida)
- Svi objekti se rasporede u ka grupa po pravilu minimalne udaljenosti

- Racuna se novih ka centroida
- Ponavljaj korake dok vise nema promjena

Bayesova teorija odlucivanja i parametarske metode

A priori vjerojatnost – $P(C)$

A posteriori vjerojatnost - $P(C|x)$

Izglednost (likelihood) – $p(x|C)$

Cinjenica (evidence) – $p(x)$

K=2 razreda

$$P(C = 0) + P(C = 1) = 1$$

$$p(x) = p(x|C = 1)P(C = 1) + p(x|C = 0)P(C = 0)$$

- vrijedi samo ako su hipoteze C_i medjusobno iskljucive, a zbroj njihovih vjerojatnost iznosi 1)

$$P(C = 0|x) + P(C = 1|x) = 1$$

K>2 razreda

$$P(C_i|x) = \frac{p(x|C_i)P(C_i)}{p(x)} = \frac{p(x|C_i)P(C_i)}{\sum_{k=1}^K p(x|C_k)P(C_k)}$$

Pri cemu mora vrijediti:

$$P(C_i) \geq 0 \text{ i } \sum_{i=1}^K P(C_i) = 1$$

Odabiremo:

$$C_i \text{ ako } P(C_i|x) = \max_k P(C_k|x)$$