# Strojno ucenje

- izgradnja sustava koji poboljsavaju performanse kroz iskustvo koristeci kriterijsku funkciju.

- programiranje racunala da podesava parametre modela na temelju danih primjera

Model (hipoteza)

* definiran i izgradjen do razine nepoznatih parametara
* prediktivni i deskriptivni

Ucenje

* ucenje pravila
* nadzirano ucenje (podaci (x,y), x je ulaz, y cilj, treba naci preslikavanje y = f(x))
* nenadzirano ucenje (nema cilja, treba naci pravilnosti u ulaznim podacima)
* podrzano ucenje

Nadzirano ucenje

* regresija
* klasificiranje/raspoznavanje uzoraka
* predvidjanje buducih slucajeva, ekstrakcija znanja, kompresija, detekcija ekstremnih vrijednosti

Podrzano ucenje

* ucenje strategije na temelju serije izlaza
* nema nadziranog ucenja, samo odgodjena nagrada
* pr. Igranje igara, robot u labirintu

# Problem ucenja

* zadatak T, mjera P, iskustvo E
* program uci ako se povecava uspjesnost obavljanja zadataka

Odabir nacina stjecanja iskustva

* 1. Svojstvo iskustva – utjecaj iskustva na ucenje
* 2. Svojstvo iskustva – stupanj u kojem ucenik kontrolira smjer primjera za ucenje
* 3. Svojstvo iskustva – reprezentativnost primjera za ucenje

Odabir ciljne funkcije

* primjer – odabir najboljeg poteza iz skupa dozvoljenih
* alternativno – evaluacijska funkcija V, funkcija pridruzuje vecu numericku vrijednost boljim stanjima

Odabir predstavljanja ciljne funkcije, Odabir algoritma za aproksimaciju ciljne funkcije

# Ucenje koncepata

* uoblicavanje definicije opcenite klase/kategorije ako je dan skup pozitivnih i negativnih primjera te kategorije
* pretrazivanjem prostora potencijalnih hipoteza za onom koja najbolje odgovara primjerima za ucenje

Koncept – podskup objekata (dogadjaja) nekog veceg skupa

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

Ucenje koncepata – izvodjenje 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 moze sadrzavati:

* ? - svaka je vrijednost prihvatljiva
* Ø - nijedna vrijednost nije prihvatljiva
* Odredjena vrijednost

Notacija

* X – skup primjera
* c – ciljna funkcija ili ciljni koncept,
* H – skup svih potencijalnih hipoteza, opisana s konjunkcijom uvjeta
* D – skup primjera za ucenje

Trazi se hipoteza

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

Za neki i hipotezu kazemo da zadovoljava akko

## Find-S Algoritam

Find-S Algoritam:

1. Inicijaliziraj h na najspecificniju hipotezu u H
2. Za svaki pozitivni primjer za ucenje x
   1. Za svaki atribut u h
      1. AKO je uvjet zadovoljen sa x TADA ne cini nista INACE zamjeni u h sa sljedecim opcenitijim 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 najspecificniju hipotezu konzistentnu s pozitivnim primjerima. Algoritam

garantira nalazenje najspecificnije hipoteze, te je ona konzistentna i s negativnim primjerima uz pretpostavku da je ciljni koncept c u H i da su primjeri za ucenje ispravni.

Induktivna pristranost

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

## Agoritam Eliminacije Kandidata

* nalazi sve hipoteze konzistentne s primjerima za ucenje – PROSTOR INACICA (oznaka )

Listaj-zatim-eliminiraj algoritam

1. Prostor inacica – lista koja sadrzi sve hipoteze iz H
2. Za svaki primjer za ucenje iz D ukloni iz prostora inacica sve hipoteze za koje je
3. Predoci na izlazu hipoteze iz prostora inacica

Najopcenitija granica G – skup maksimalno opcenitih clanova skupa hipoteza H konzistentnih s primjerima za ucenje D

Najspecificnija granica S – skup maksimalno specificnih clanova skupa hipoteza H konzistentnih s primjerima za ucenje D

Algoritam Eliminacije Kandidata

1. Inicijaliziraj G na skup maksimalno opcenitih hipoteza u H
2. Inicijaliziraj S na skup maksimalno specificnih hipoteza u H
3. Za svaki primjer za ucenje ucini
   1. Ako je d pozitivni primjer
      1. Ukloni iz G sve hipoteze nekonzistentne s d
      2. Za svaku hipotezu koja nije konzistentna s d
         1. Ukloni
         2. Dodaj u S sve minimalne generalizacije h od s takve da je h konzistentna s d i neki clan iz G je opcenitiji od h (osigurava konzistentnost s prethodnim negativnim primjerima
         3. Ukloni iz S sve hipoteze koje su opcenitije od drugih hipoteza u S
   2. Ako je d negativni primjer
      1. Ukloni iz S sve hipoteze nekonzistentne s d
      2. Za svaku hipotezu koja nije konzistentna s d
         1. Ukloni
         2. Dodaj u g sve minimalne specijalizacije h od g takve da je ha konzistentna s d i neki clanovi iz S su specificniji od h (osigurava konzistentnost s prethodnim pozitivnim primjerima)
         3. Ukloni iz G sve hipoteze specificnije od drugih hipoteza u G

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

Induktivna pristranost

* Ciljni koncept c je sadrzan u prostoru hipoteza H!

Algoritam ROTE LEARNER nema induktivne pristranosti

# Stabla odluke

* Neparametarksa metoda
* Metoda aproksimiranja funkcije diskretnih (i realnih) vrijednosti, robusna na sum, moze uciti disjunktivne koncepte
* Pretrazuje 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 vraca zbog ponovnog razmatranja prethodnih cvorova

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

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

- proporcija negativnih primjera u S

Po definiciji

Informacijska dobit atributa A u odnosu na skup primjera S jest

Postupak:

1. Odrediti informacijsku dobit svakog atributa
2. Atribut s najvecom vrijednoscu postaje korijen
3. Skup na ucenje podijeliti na podskupove odredjene razlicitim vrijednostima atributa korijena
4. Ponoviti postupak na svakom od podskupova

ID3

* Pretrazuje potpun prostor hipoteza
* Pronalazi samo jednu hipotezu
* Ne vraca se unatrag u postupku
* Koristi sve primjere za ucenje u svakom koraku da bi statisticki rafinirao tekucu hipotezu

Priblizna induktivna pristranost – preferiraju se kraca stabla odluke nad vecima

Bolja priblizna induktivna pristranost – preferiraju se kraca stabla odluke nad vecima, preferiraju se stabla koja stavljaju atribute s vecom informacijskom dobiti blize korijenu

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

Ako podaci imaju sum, ili je skup za ucenje premalen, moze doci do prenaucenosti. Hipoteza h je prenaucena ako postoji hipoteza h1 takva da h ima manju pogresku od h1 na primjerima za ucenje, ali h1 ima manju pogresku u cijelom prostoru primjera.

Prenaucenost se izbjegava

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

Razumna velicina stabla se odredjuje statisticki, uvodjenjem eksplicitne mjere kompleksnosti, uvodjenjem skupa za vrednovanje, itd.

Podrezivanje stabla – uklanjanje cvora i pripadnog podstabla zamjenjujuci ga s listom tako da se listu pridruzi najcesca 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 tocno klasificiranih primjera = TP + TN

Broj pozitivnih primjera = TP + FN

Broj negativnih primjera = TN + FP

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

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

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

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

F mjera – harmonijska sredina preciznosti i odziva

# Nadzirano ucenje

-ucenje koncepata (klasa, razreda) na temelju primjera

-primjer, Klasa C „obiteljski auto“

- predvidjanje : Je li auto x obiteljski auto“

- izlaz: pozitivni i negativni primjeri

- ulaz: x1 - cijena, x2 – snaga motora

- skup za ucenje X = {x, r}

- pretpostavka (na temelju prethodnog ispitivanja) o C:

- (p1 <= cijena <= p2) && (e1 <= snaga motora <= e2)

- prostor hipoteza H – skup kvadrata (p1,p2,e1,e2)

Problem generalizacije – kako dobro ce se klasificirati buduci primjeri

Vapnik – Chervonenkisova dimenzija

* N primjera se moze oznaciti na 2¨n nacina kao pozitivni i negativni
* Ako postoji skup od N primjera tako da za svako oznacivanje postoji hipoteza h iz H koja je konzistentna s primjerima, kazemo da H razdjeljuje N primjera
* VC dimenzija skupa H je najveci N za koji H razdjeljuje N primjera

# Kombiniranje strojno naucenih algoritama

1. Pregled znacajnijih pristupa
   1. Nezavisno nauceni eksperti (varijante glasanja)
   2. Viserazinsko ucenje (boostanje, kaskadiranje
2. Postupak Viole i Jonesa

Ideja – kombiniranjem naucenih osnovnih algoritama postici sinergiju

Varijacija

* Razlicite grupe algoritama
* Razliciti parametri istih algoritama
* Razliciti modaliteti ulaza
* Razliciti odabiri skupova za ucenje

Notacija

* L – broj osnovnih algoritama
* K – broj razreda kod klasifikacije
* Ako algoritmi imaju po jedan izlaz
  + - rezultat j-tog algoritma za ulaz x
  + - konacni rezultat uz parametre
* Ako algoritmi imaju po K Izlaza
  + - i-ti rezultat j-tog algoritma za ulaz x
  + - konacni i-ti rezultat uz parametre

Glasanje

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

Boostanje – skup osnovnih algoritama konstruiramo tako da sljedbenici uce 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
* Standardna devijacija
* Normalna distribucija s ocekivanjem i varijancom –
* Normalna distribucija s ocekivanjem 0 i standardnom devijacijom 1 – U
* STANDARDIZACIJA
  + svodjenje na U
  + transformacijom

# Podrzano ucenje

* Q-ucenje
* Procjenitelj (critic) ne govori unaprijed sto raditi, nego daje odgovarajuce 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 ucenje
  + Istrazivanje nepoznatih stanja u potrazi za informacijama
  + Iskoristavanje poznatih stanja
* Cijelozivotno ucenje
* Djelomicno vidljiva stanja

Pojmovi

* Skup stanja (state) -
* Skup akcija (action) -
* Strategija (policy) –
* 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 –
* Dobrota strategije odlucivanja, pocevsi iz stanja s
* Optimalna strategija

Ucenje s unaprijed poznatim modelom svijeta

* Odgodjena nagrada
* Rijesenje – dinamicko programiranje
* Izuzetno rijeti slucajevi

Ucenje s iterativnim upoznavanjem svijeta

* Svijet nije u potpunosti unaprijed pozna
* Pokriva vecinu problema

Nacini ucenja strategije

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

Q ucenje

* Za svako stanje postavi vrijednost Q(s,a) na nulu
* Postavi se u neko stanje s
* Ponavljaj beskonacno
  + Odaber i izvrsi akciju a
  + Primi nagradu r
  + Osvjezi zapis Q(s,a) prema formuli:
    - Bez stope ucenja -
    - stopa ucenja –
  + Uzmi stanje s' kao novo stanje s

Algoritam konvergira pod odredjenim uvjetima

1. sustav je deterministicki markovljev proces odlucivanja
2. nagrade su ogranicene
3. svaki par stanje-akcija se izvede beskonacno cesto

Nacin osvjezavanja – 1 korak, ucenje unatrag, epizoda

# Ucenje na temelju primjera

* metoda k-najblizih susjeda
* metoda lokalne regresije s tezinskim faktorima
* zakljucivanje na temelju slucajeva
* radijalne bazne funkcije

Lazy vs. Eager

* lazy metode
  + odadjaju odluku o klasifikaciji do trenutka predocavanja novog primjera
  + +konstruiraju razlicitu aproksimaciju ciljne funkcije za svaki razliciti 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 ucenje
  + za svaki primjer za ucenje (x,f(x)) dodaj primjer na listu primjeri \_za\_ucenje
* algoritam klasifikacije
  + za dani primjer xq s nepoznatom klasifikacijom
  + neka x1,x2…xk oznacavaju k primjera koji su najblizi xq
* vrati
  + gdje je
* f(xq) je najcesca vrijednost ciljne funkcije koja se pojavljuje medju k primjera za ucenje koji su najblizi upitu xq

Regresija pomocu k-nn algoritma

* umjesto najcesce pojavljivane vrijednosti ciljne funkcije, odgovor na upit je srednja vrijednost cilnih funkcija k najblizih susjeda

Induktivna pristranost – pretpostavka da je klasifikacija upita xq 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 :

* + – euklidska metrika
  + – manhattan ili cityblock metrika, ili u slucaju binarnih vektora, hammingova metrika
  + - cebisevljeva udaljenost

Mahalonobisova udaljenost

* Pozitivno definitna kvadratna forma oblika gdje je 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 –

A posteriori vjerojatnost -

Izglednost (likelihood) –

Cinjenica (evidence) –

K=2 razreda

- vrijedi samo ako su hipteze Ci medjusobno iskljucive, a zbroj njihovih vjerojatnost iznosi 1)

K>2 razreda

Pri cemu mora vrijediti:

Odabiremo: