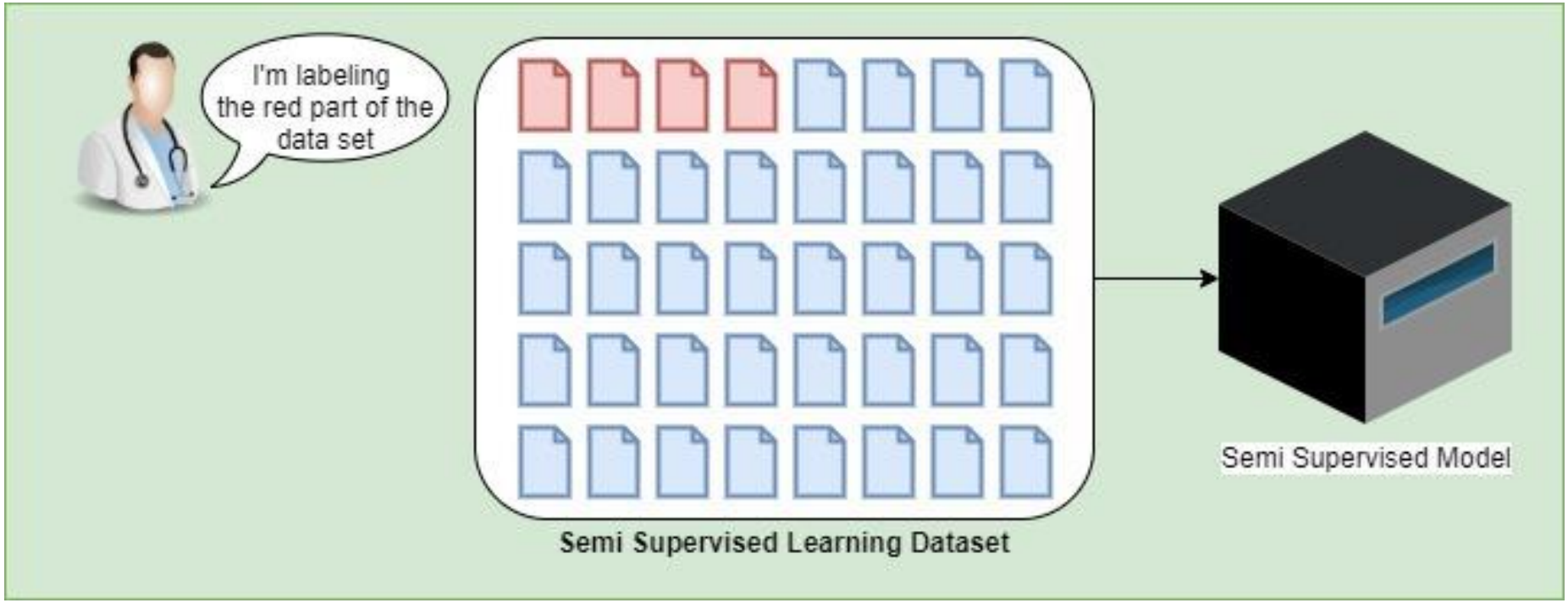


Polu-nadgledano obučavanje

Semi-Supervised Learning

Motivacija

- U nadgledanom obučavanju (*supervised learning*) trening skup se sastoji od anotiranih primera $T = \{(x^{(i)}, y^{(i)}), i = 1, \dots, N\}$

- Problem:
  - Da bi rezultujući predikcioni model bio kvalitetan, neophodno je prikupiti što veći i što raznovrsniji obučavajući skup
  - Obučavajući skupovi se formiraju ručnom anotacijom od strane eksperata i zbog toga njihovo formiranja može biti dugotrajno i skupo

- Možemo ublažiti ovaj problem:
  - Uključivanjem neanotiranih podataka u proces obučavanja
  - Lako je i jeftino doći do dovoljne količine podataka koji nisu anotirani

Cilj

- Za date trening podatke:
  - Anotirani skup $L = \{(x^{(i)}, y^{(i)})\}$
  - Neanotirani skup $U = \{(x^{(i)})\}$
  - Obično je $|U| \gg |L|$

- Obučiti klasifikator f koji je bolji od klasifikatora obučenog samo na anotiranim podacima L

Tri pravca

Induktivno obučavanje

1. Polu-nadgledano obučavanje (*Semi-Supervised Learning*, SSL)

1. Aktivno obučavanje (*Active Learning*)

3. Transduktivno obučavanje (*Trunsductive Learning*)

Induktivno obučavanje

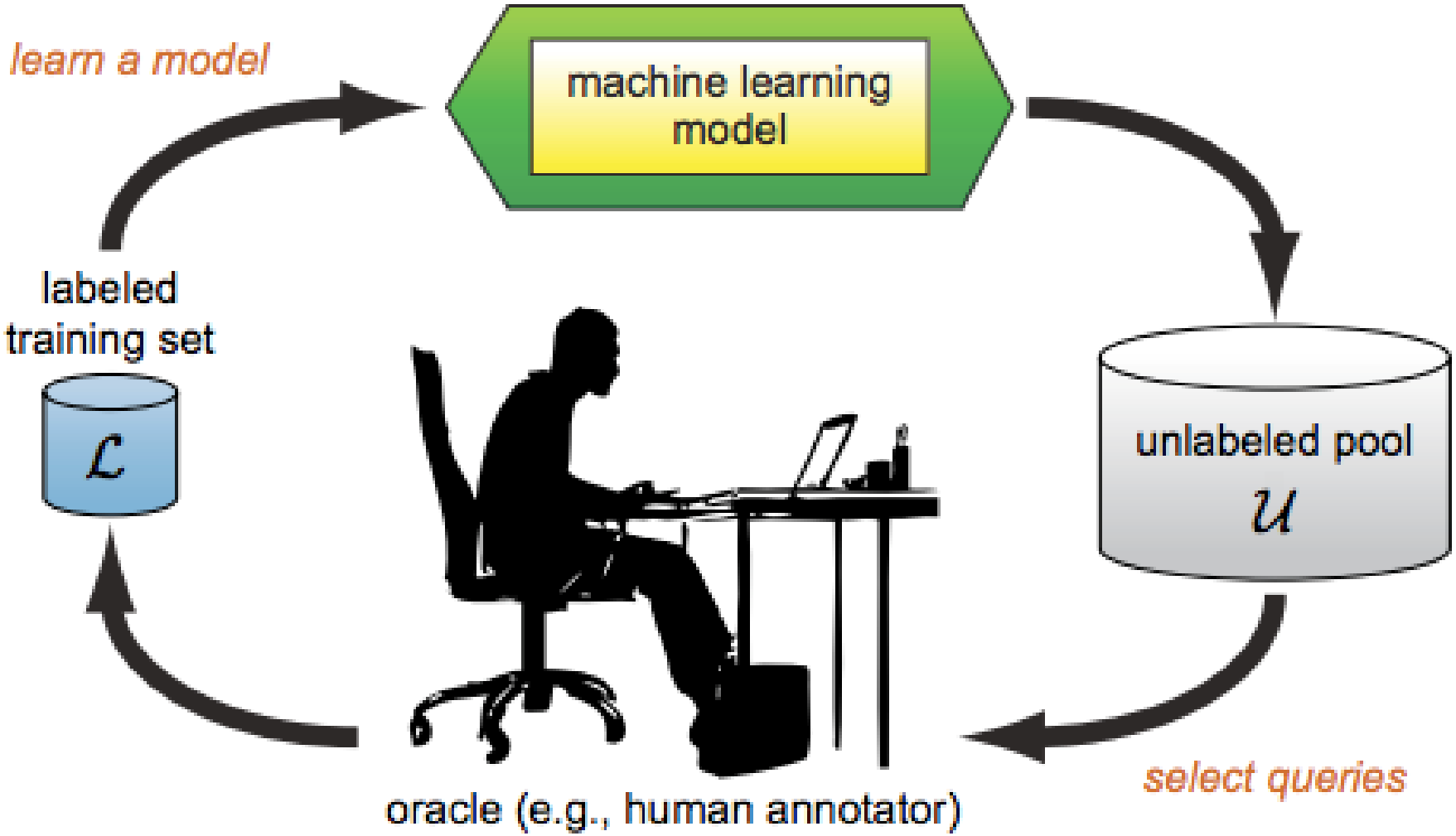
- U induktivnom obučavanju učimo funkciju *f* koju ćemo primeniti na test skup
- Za date trening podatke $L = \{(x^{(i)}, y^{(i)})\}$ i $U = \{(x^{(i)})\}$, naučiti funkciju f
- Koristiti f da se odrede labele budućih (neanotiranih) primera

Transduktivno obučavanje

- Dati su nam trening podaci:
  - $L = \{(x^{(i)}, y^{(i)})\}$ i $U = \{(x^{(i)})\}$

- Ne učimo eksplicitnu funkciju
- Ne dobijamo neke „buduće“ test podatke
- Sve što nas zanima jeste da odredimo labele za U
  - Neanotirani podaci = test podaci
  - Test podaci U su nam dostupni tokom obučavanja

Aktivno obučavanje



Cilj

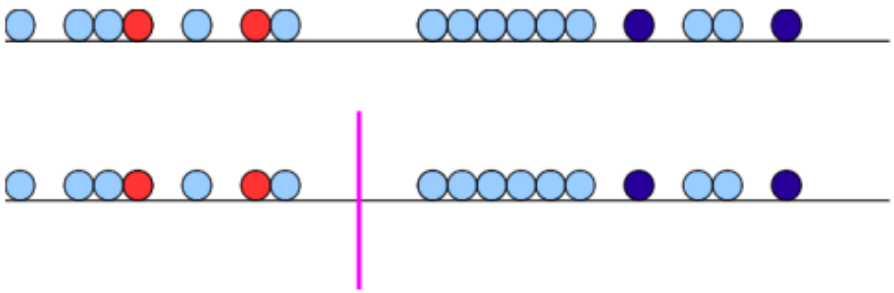
- U sva tri slučaja cilj je isti:
  - Minimizovati broj anotiranih instanci neophodnih za obučavanje, a zadržati performanse
  - Ovo u značajnoj meri smanjuje neophodan ljudski rad pri kreiranju obučavajućeg skupa

- U ovoj prezentaciji koncentrisaćemo se na tehnike polu-nadgledanog obučavanja

Kako neanotirani podaci mogu pomoći?

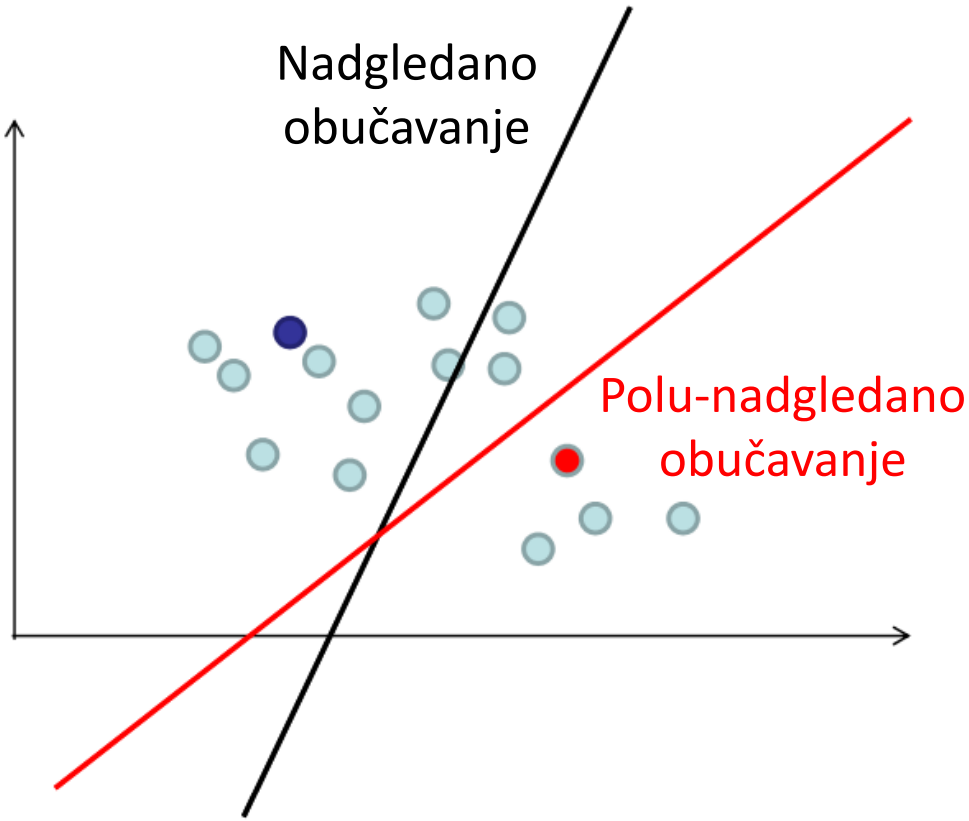
- Crvene tačke: +1, Tamno plave tačke: -1

- Uključimo dodatne neanotirane podatke (svetlo plave tačke):



- Pretpostavka: primeri iste klase prate koherentnu distribuciju
- Neanotirani podaci nam mogu dati bolji osećaj za granicu odluke

Kako neanotirani podaci mogu pomoći?



- Pretpostavka:
  - Svaka klasa se sastoji od grupe koherentnih tačaka, npr., Gausijan
  - Granica odluke treba da prolazi kroz regije male gustine
- Neanotirani podaci nam pomažu da steknemo bolji osećaj za $P(x)$ - omogućiti nam da odredimo ovu distribuciju preciznije
- A pretpostavljamo da postoji veza između $P(x)$ i $P(y|x)$

Nema besplatnog ručka...

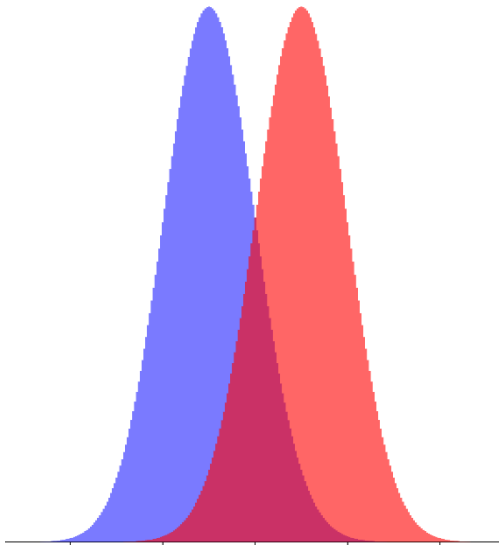
- Ukoliko uvedene pretpostavke nisu tačne, polu-nadgledano obučavanje može degradirati performanse
- Generalno, ne postoji algoritam polu-nadgledanog obučavanja koji je univerzalno superioran u odnosu na ostale
- U praksi, najbolje će raditi model čije pretpostavke najbolje odgovaraju podacima

Izbegavanje promena u gustim regijama

- Pretpostavka: granica odluke ne treba da prolazi kroz regije u kojima je $P(x)$ visoko

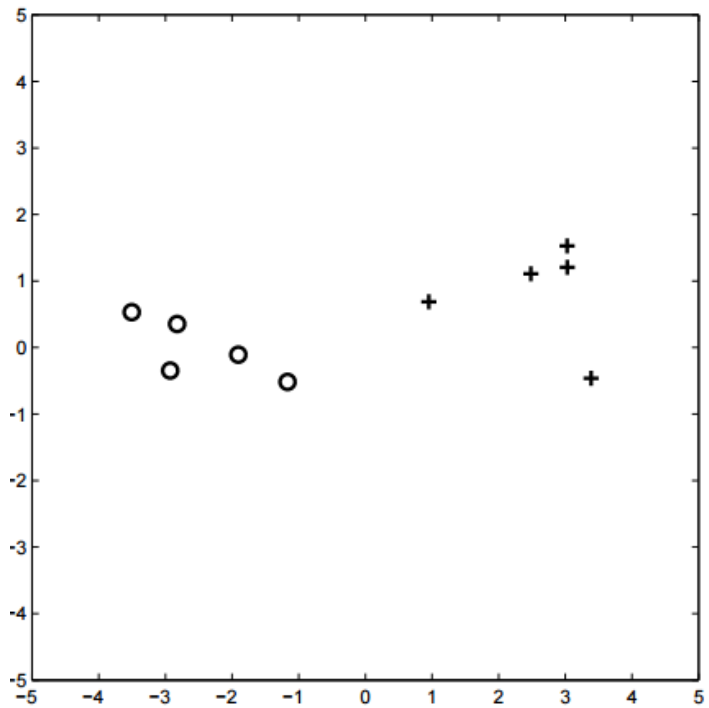
- Transduktivne mašine potpornog vektora, gausovski procesi, regularizacija informacija, minimizacija entropije

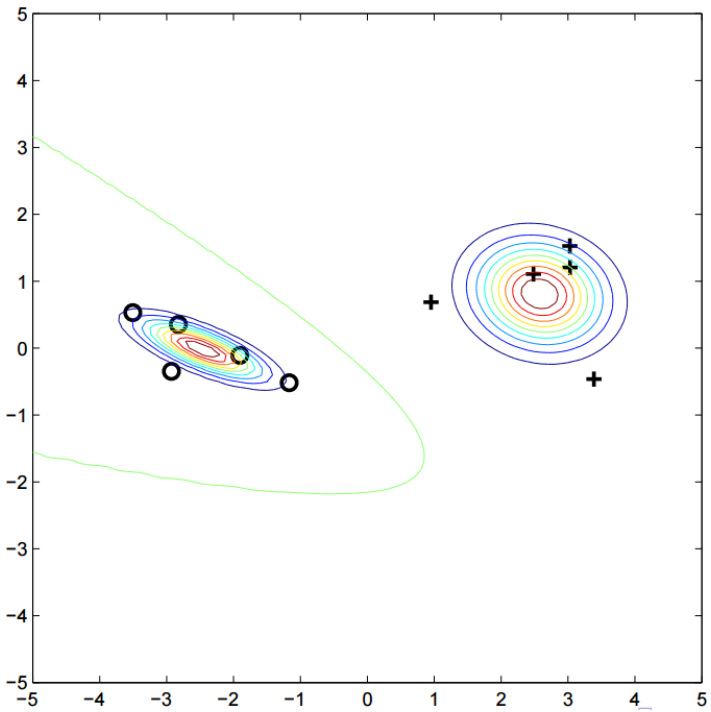
- Kao i kod svih polu-nadgledanih modela, ukoliko uvedene pretpostavke nisu zadovoljene, ove metode će imati slabije performanse



Generativni modeli

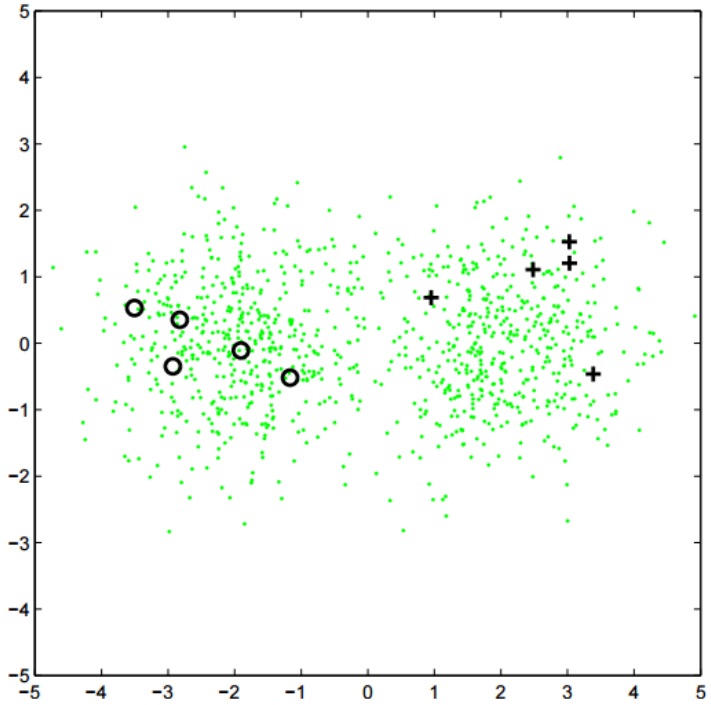
- Statistička distribucija se modeluje mešavinom (težinskom sumom) drugih distribucija
- Pretpostavka: primeri obe klase prate Gausovu distribuciju

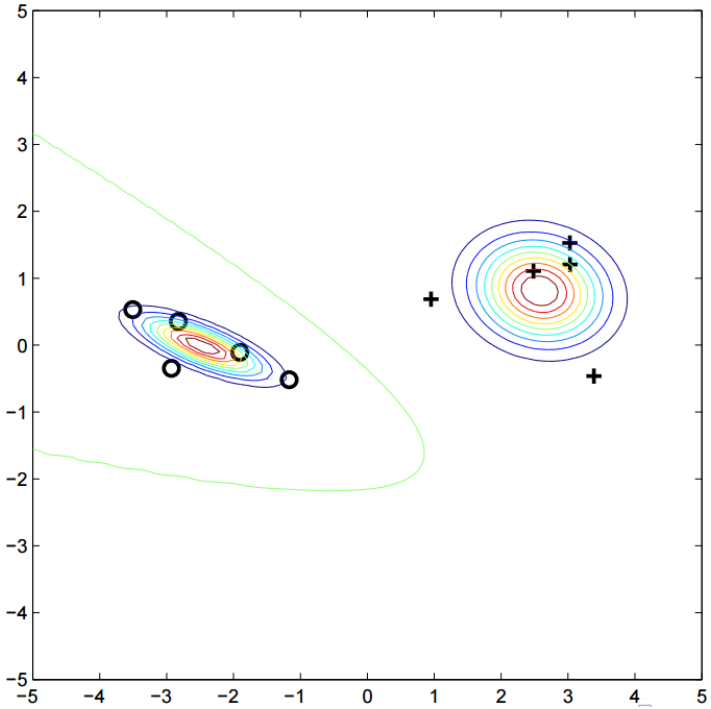




Generativni modeli

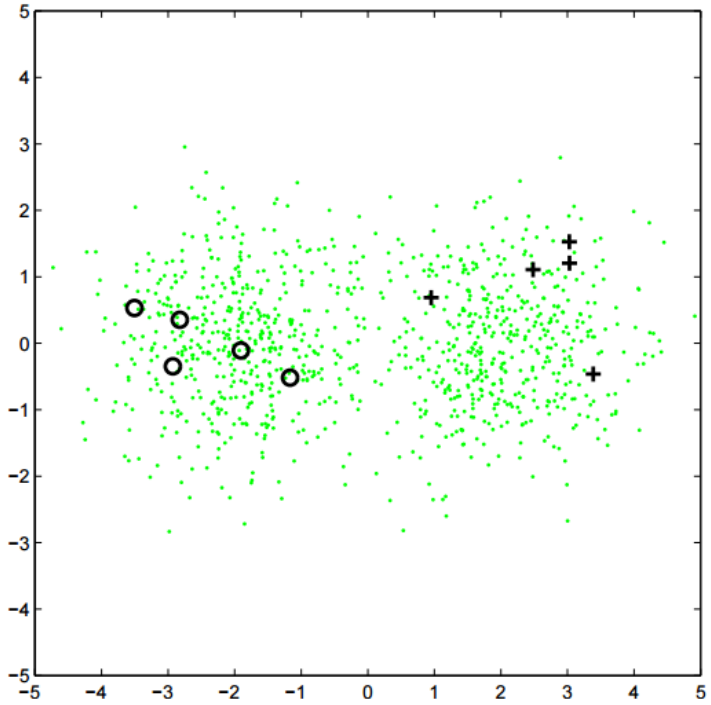
- Statistička distribucija se modeluje mešavinom (težinskom sumom) drugih distribucija
- Pretpostavka: primeri obe klase prate Gausovu distribuciju

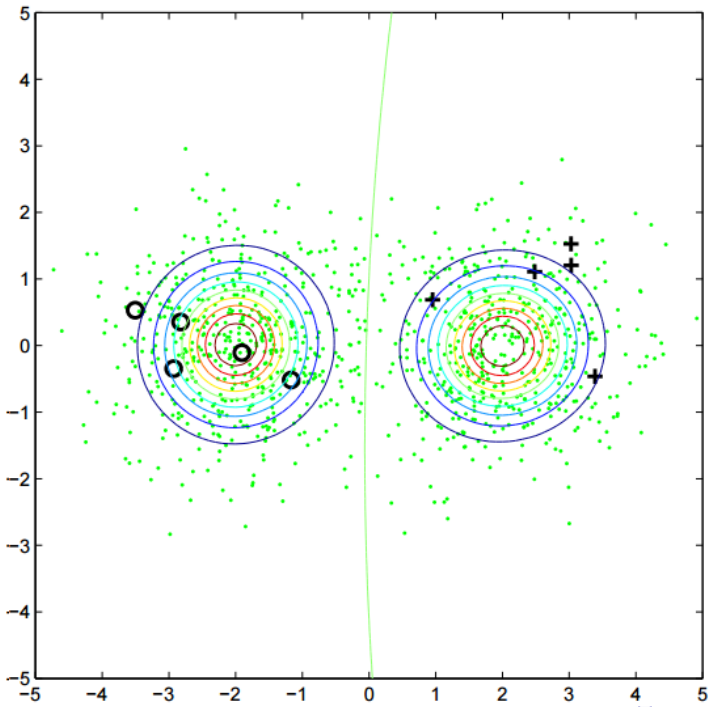




Generativni modeli

- Statistička distribucija se modeluje mešavinom (težinskom sumom) drugih distribucija
- Pretpostavka: primeri obe klase prate Gausovu distribuciju





Generativni modeli

- Često korišćeni u polu-nadgledanom učenju:
  - Mešavina Gausovih distribucija (GMM)
    - Klasifikacija slika
    - Optimizacija pomoću EM algoritma
  - Mešavina multinominalnih distribucija (Naïve Bayes)
    - Kategorizacija teksta
    - Optimizacija pomoću EM algoritma
  - Hidden Markov Models (HMM)
    - Prepoznavanje govora
    - Baum-Welch algoritam

Metoda očekivanje-maksimizacija

- Parametri generativnog modela se mogu estimirati metodom očekivanje-maksimizacija (*Expectation-Maximization*, *EM*)
- EM metoda je primenljiva kada se u podacima javljaju nedostajuće vrednosti
- U kontekstu polu-nadgledanog obučavanja, klasna obeležja tretiramo kao nedostajuće vrednosti

- Iterativno ponavljati ova dva koraka (do konvergencije):
  1. Nadgledanim obučavanjem (skup *L*) odrediti parametre generativnog modela. Iskoristiti model da se neanotiranim instancama (skup *U*) dodele klasna obeležja
  2. Re-evaluirati parametre modela koristeći sve instance (uključujući i one anotirane u koraku 1)

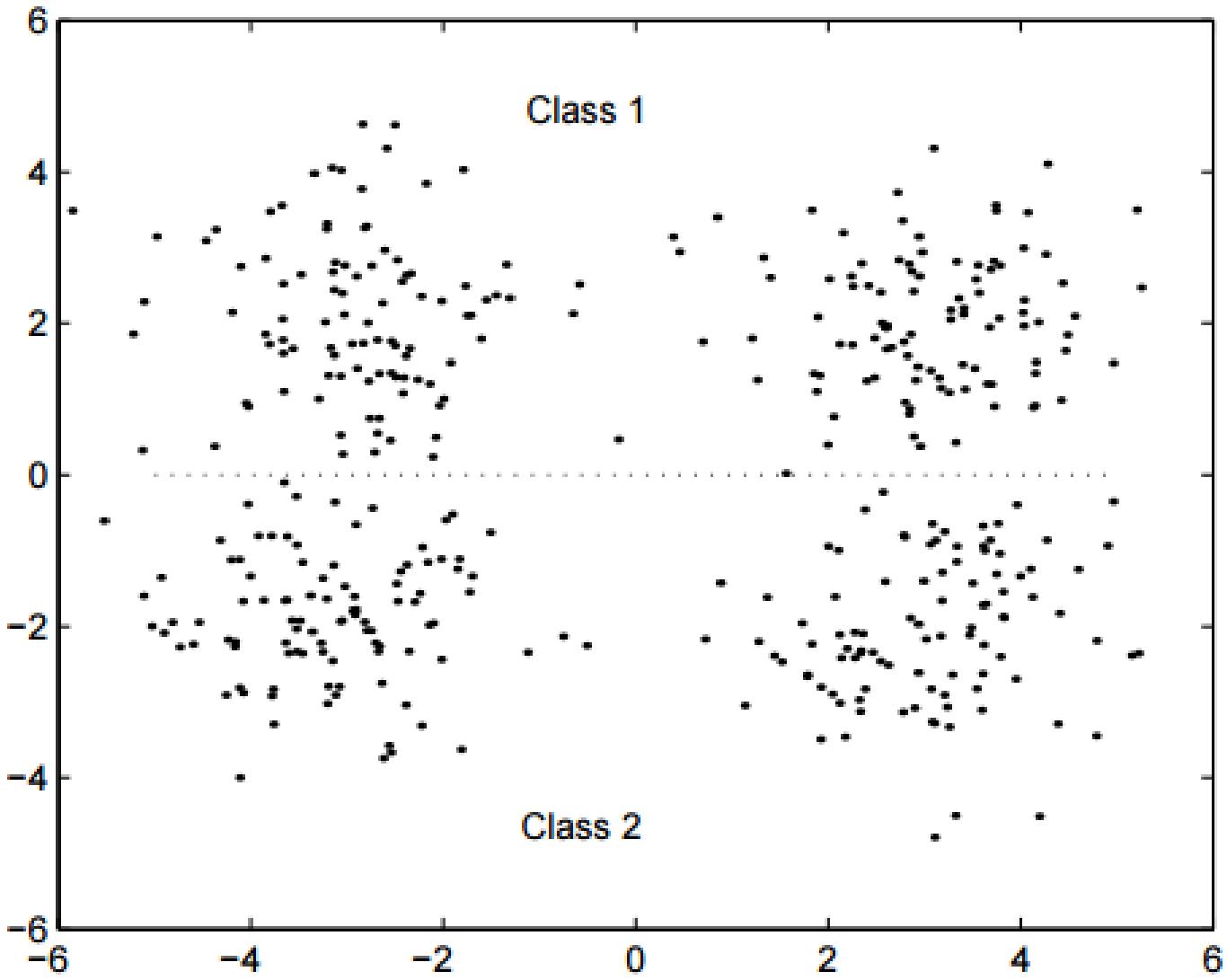
Generativni modeli

- Uvedena pretpostavka:
  - Generativni model $P(x, y|\theta)$

- Prednosti:
  - Jasno, dobro proučeno probabilističko okruženje
  - Može da bude izuzentno efektivno, ukoliko je pretpostavljeni model blizak stvarnom modelu

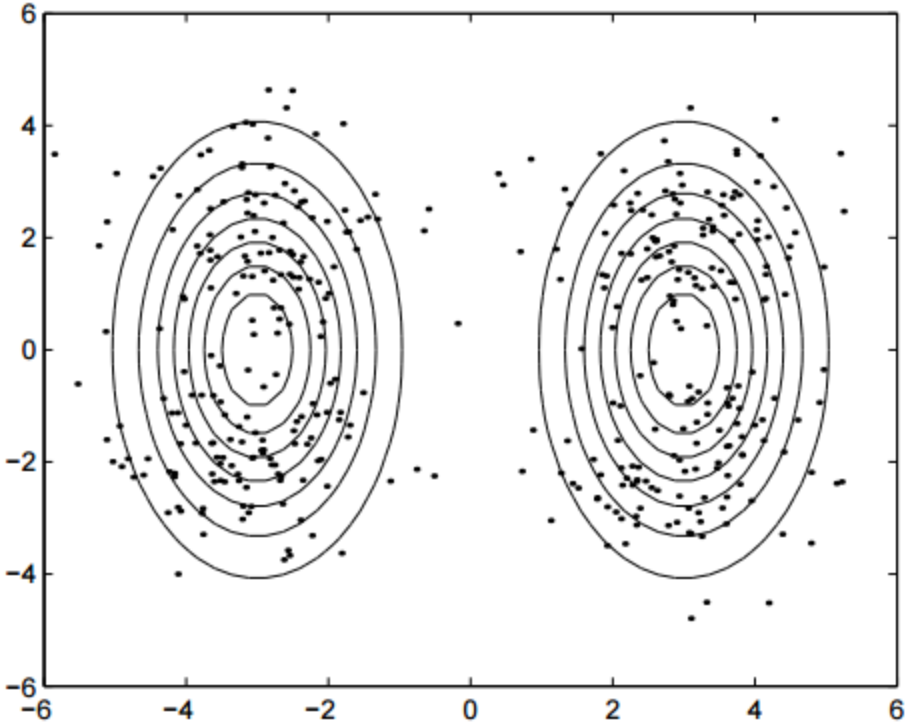
- Mane:
  - Često je teško verifikovati da li pretpostavljeni model odgovara podacima
  - A ukoliko ne odgovara, neanotirani podaci mogu degradirati performanse modela
  - Parametri modela se često procenjuju primenom EM metode koja je podložna upadanju u lokalni optimum

Generativni modeli – loš slučaj

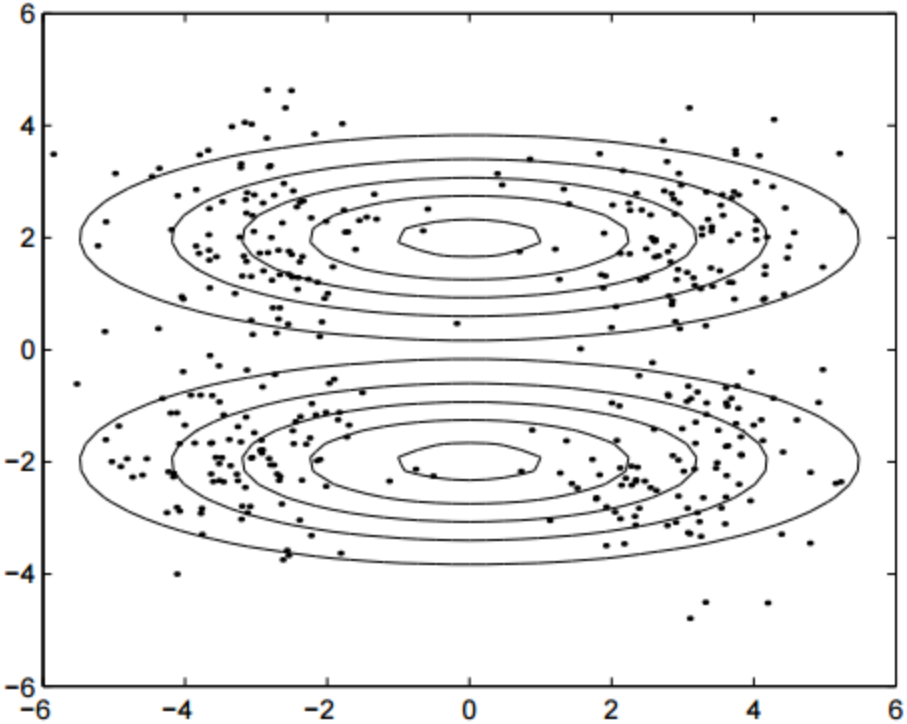


Generativni modeli – loš slučaj

high likelihood
wrong



low likelihood
correct



Sličan model – klasterovati-i-anotirati

- Umesto probabilističkih generativnih modela, možemo koristiti bilo koji algoritam za klasterovanje:
  - Primeniti model za klasterovanje na *L* i *U*
  - Anotirati sve neanotirane instance unutar klastera većinskim glasanjem anotiranih instanci u klasteru
  - Trenirati nadgledani model nad svim instancama

- Pretpostavka:
  - Ako dve tačke pripadaju istom klasteru, onda verovatno pripadaju istoj klasi
- Prednost:
  - jednostavan metod, poznati algoritmi
- Mana:
  - Može biti težak za analizu
  - Degradira performanse ukoliko pretpostavka nije tačna

Modeli bazirani na grafovima

- Generišemo graf:
  - Čvorovi: instance (anotirane i neanotirane)
  - Grane: reflektuju sličnost instanci
    - *K*-NN graf (bez težina)
    - Potpuno povezan graf gde se težine smanjuju sa udaljenošću (npr. Euklidskom)

- Ove metode se obično baziraju na pretpostavci o „glatkoći“ labela u grafu
  - Bliski čvorovi (povezani granom velike težine) trebaju da imaju slične labele
  - Ova ideja se naziva i *Graph-based regularization*

Modeli bazirani na grafovima – primer

- Klasifikacija teksta: klasifikovati članke na *astronomy* i *travel*
- Sličnost teksta ćemo meriti preklapanjem reči (preklapanjem sadržaja članka)

	d_1	d_3	d_4	d_2
asteroid	•			
bright	•			
comet				
year				
zodiac		•		
⋮				
airport			•	
bike			•	
camp				
yellowstone				•
zion				•

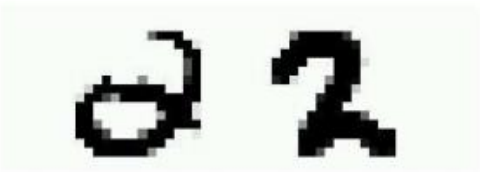
Skup *L*: nema reči koje se preklapaju...

	d_1	d_5	d_6	d_7	d_3	d_4	d_8	d_9	d_2
asteroid	•								
bright	•	•							
comet		•	•						
year			•	•					
zodiac				•	•				
⋮									
airport						•			
bike						•	•		
camp							•	•	
yellowstone								•	•
zion									•

Koristićemo neanotirane podatke kao „stepping stones“ – labele se „propagiraju“ kroz slične neanotirane članke

Modeli bazirani na grafovima – primer

- Prepoznavanje rukom pisanog teksta
- Sličnost: *pixel-wise* Euklidska udaljenost

	
not similar	'indirectly' similar with stepping stones

Učenje efikasnog enkodiranja domena

- Na osnovu neanotiranih podataka moguće je naučiti efikasno enkodiranje obeležja problemskog domena
- Primena metode za redukciju dimenzionalnosti korišćenjem neanotiranih podataka (npr. PCA)

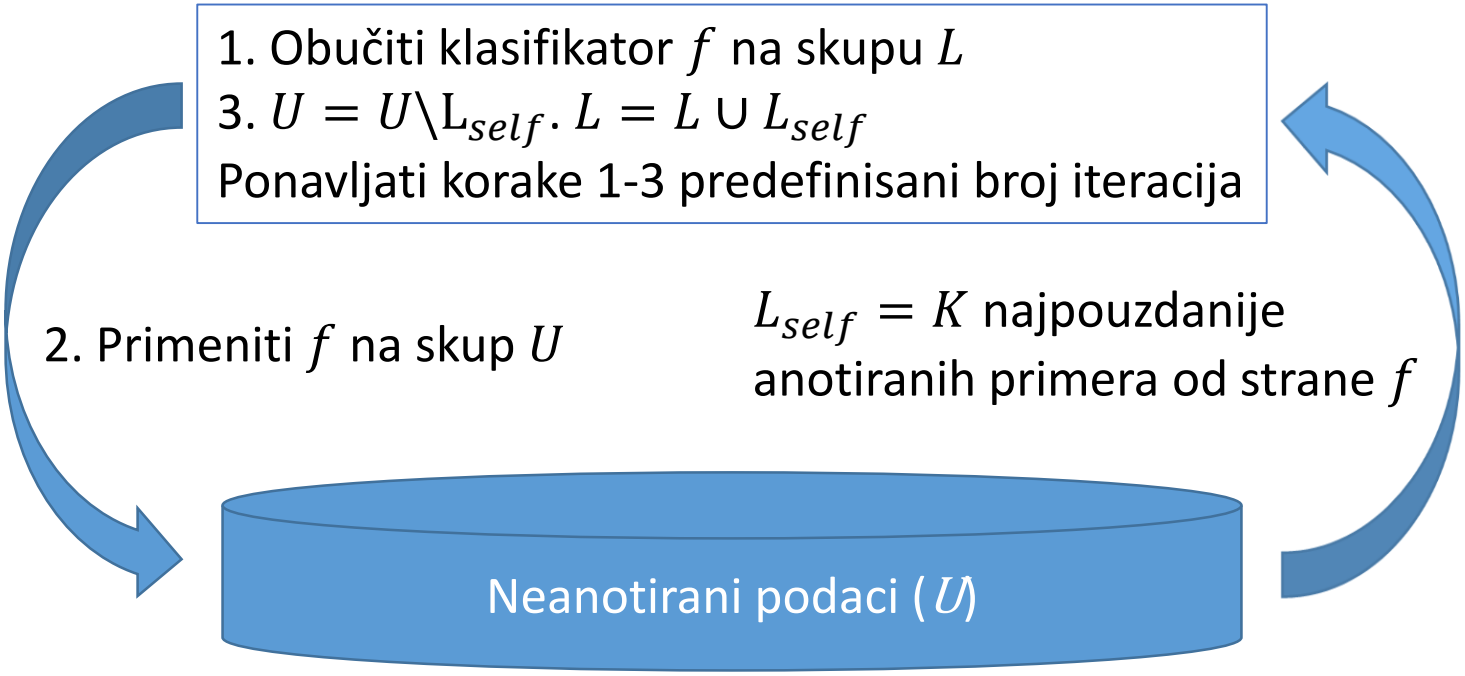
- Za anotirane podatke možemo koristiti rezultujuću niže-dimenzinu reprezentaciju
- Klasifikacija se potom vrši primenom standardnog nadgledanog obučavanja

Korišćenje znanja o proporcijama klasa

- Pod proporcijama klasa ovde se misli na proporcije instanci klasifikovanih u svaku od klasa
  - npr. 20% pozitivnih i 80% negativnih
- Bez uvođenja ograničenja o proporcijama, metode polu-nagledanog obučavanja imaju tendenciju da proizvode nebalansirani izlaz
- U ekstremnom slučaju se može desiti da se svi neanotirani podaci svrstaju u istu klasu, što je veoma nepoželjno
- Zbog toga, mnoge metode polu-nadgledanog obučavanja koriste neku formu ograničenja nad klasnim proporcijama
- Željene klase propocija se ili daju kao ulaz u algoritam ili se procenjuju na osnovu proporcija klasa anotiranog skupa podataka

Samo-obučavanje (*self-training*)

- Dat je anotirani skup L i neanotiran skup U:

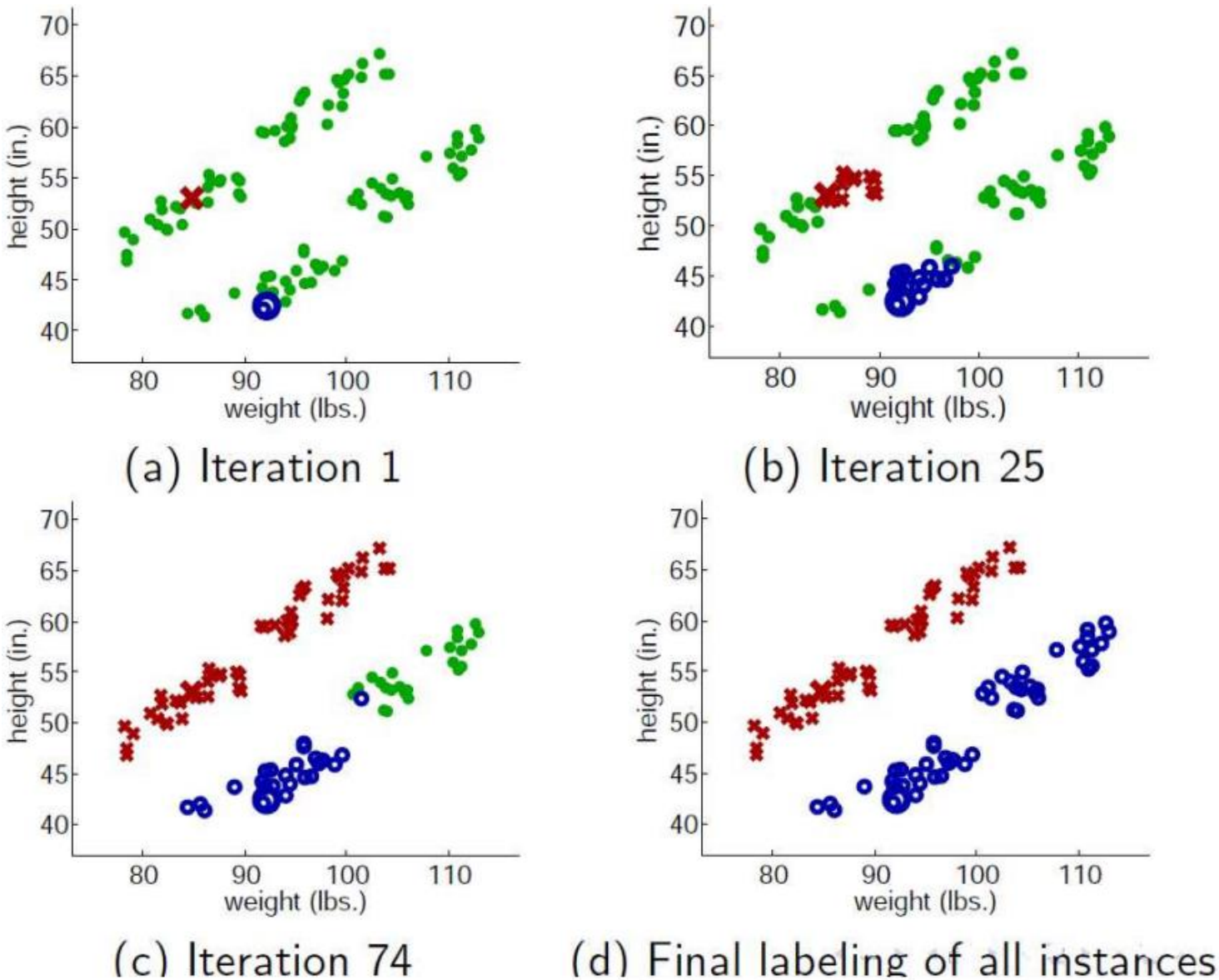


- Klasifikator koristi sopstvene (najbolje) predikcije da iterativno obučava samog sebe

Samo-obučavanje (*self-training*)

- Pretpostavka:
  - Sopstvene predikcije visoke pozdanosti (*confidence*) su tačne
- Prednosti:
  - Najjednostavniji metod polu-nadgledanog obučavanja
  - Može se primeniti bilo koji nadgledani (*supervised*) model bilo koje kompleksnosti
  - Često radi dobro u praksi (često korišćen u procesiranju prirodnog jezika)
- Potencijalni problem:
  - Rane greške u klasifikaciji mogu da pojačavaju same sebe

Samo-obučavanje dobar slučaj (*K*-NN)



(a) Iteration 1

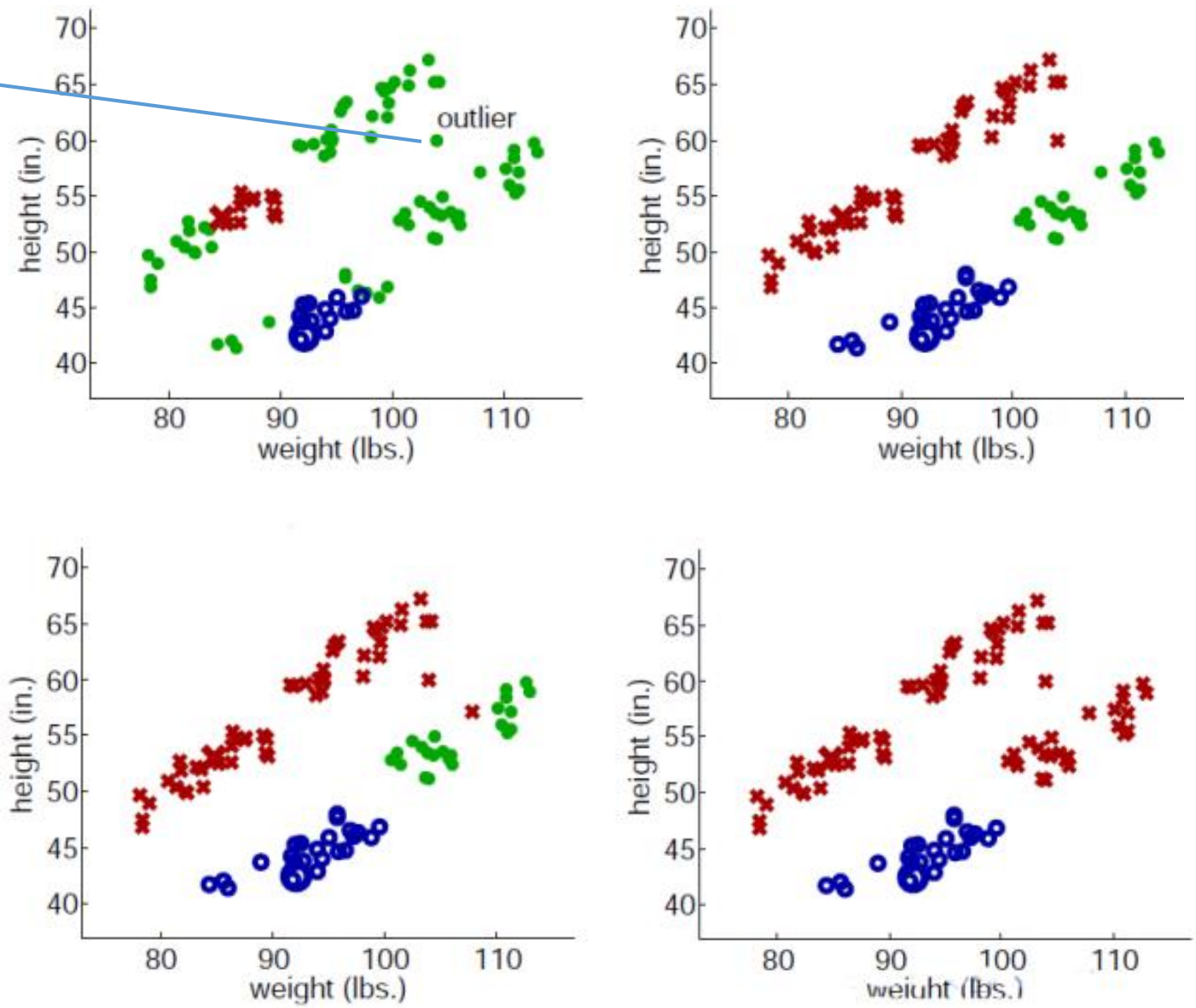
(b) Iteration 25

(c) Iteration 74

(d) Final labeling of all instances

Samo-obučavanje loš slučaj (*K*-NN)

Stvari mogu da se pogoršaju ako postoje outlieri: greške se pojačavaju



Ko-trening (*Co-Training*)

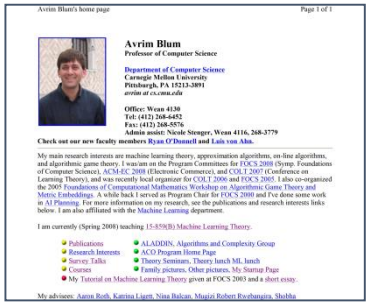
- Kod određenih problema, postojeće podatke možemo opisati na osnovu dva različita izvora informacija
- Primer: klasifikacija web stranica
  - želimo elektronski da posetimo veb sajt univerzitata i automatski skinemo sve veb stranice njegovog nastavničkog osoblja (pozitivna klasa)
  - Ručna anotacija je mukotrpna i dugotrajna
  - Međutim, web sadrži ogromne količine neanotiranih web stranica, do kojih možemo doći jednostavno uz pomoć crawler-a
  - Svaku web stranicu možemo predstaviti uz pomoć dva odvojena izvora informacija:

1. Tekst koji se nalazi u linkovima koji ukazuju na datu stranicu

2. Tekst same web stranice

My Advisor

Prof. Avrim Blum

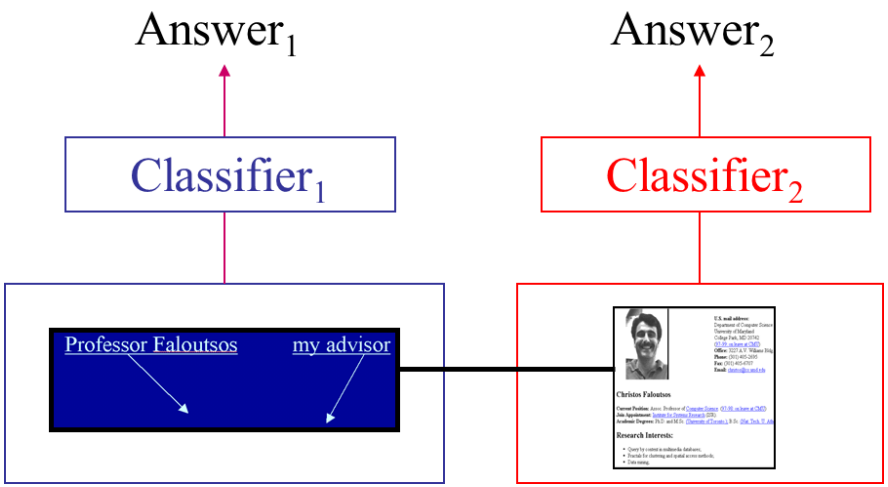


Ko-trening – primer različitih pogleda

- Slika i HTML tekst

Ko-trening intuicija

- Pretpostavka: svaki od izvora informacija je zasebno dovoljan za klasifikaciju
  - veb stranicu bi smo mogli klasifikovati posmatrajući isključivo reči iz linkova koji vode na datu stranicu
  - veb stranicu bi smo mogli klasifikovati posmatrajući isključivo reči same stranice
- Koristeći različite izvore podataka obučiti različite klasifikatore (Classifier$_1$ i Classifier$_2$)



- Klasifikatori obučeni na ovaj način bi trebalo da:
  1. Korektno klasifikuju anotirane podatke
  2. Se slažu oko klasifikacije neanotiranih podataka

Ko-trening intuicija

- Koristeći mali broj raspoloživih anotiranih primera, naučiti polazna pravila
  - Tradicionalno: obučiti jedan klasifikator koristeći *sva* obeležja
  - Co-training: obučiti dva klasifikatora na dva različita *podskupa* obeležja
  - Prvi klasifikator (link): *my advisor* → pozitivna klasa
  - Drugi klasifikator (tekst): *I am teaching* → pozitivna klasa

- Među neanotiranim instancama potražiti one koje jedan od klasifikatora može pouzdano da klasifikuje, a drugi ne može. Dozvoliti prvom klasifikatoru da anotira primer radi obuke drugog
  - Tekst stranice ne sadrži tekst *I am teaching* → drugi klasifikator ne zna da je klasifikuje
  - Link iste stranice sadrži tekst *my advisor* → prvi klasifikator je klasifikuje kao pozitivnu klasu
  - Tekst date stranice sadrži *my publications* → drugi klasifikator uči pravilo *my publications*→ pozitivna klasa

Ko-trening algoritam

Ulaz

Mali skup L anotiranih instanci, opisan pomoću skupa obeležja X
Znatno veći skup U primera koji nisu anotirani
Podela skupa X na dva neprazna podskupa X_1 i X_2
Ko-trening parametri:

k – broj iteracija;
n, p - broj instanci pozitivne i negativne klase koji će se u svakoj iteraciji dodavati u obučavajući skup;
u - veličina poskupa *U'*

Treniranje modela

Slučajnim odabirom u primera kreirati podskup U' skupa U

Za svako i, i=1..k:

- Iskoristiti L za treniranje klasifikatora h_1 koji uzima u obzir samo podskup obeležja X_1 i klasifikatora h_2 koji uzima u obzir samo podskup obeležja X_2
- Dozvoliti svakom od klasifikatora h_1 i h_2 da označi p pozitivnih i n negativnih najpouzdanije klasifikovanih primera iz U'. Dodati ovako anotirane primere u L
- Slučajnim izborom $2\cdot(p+n)$ primera iz U dopuniti skup U'

Klasifikacija instanci

Za datu instancu se za svaku klasu izračuna verovatnoća da instanca pripada datoj klasi tako što se pomnože verovatnoće koje za datu klasu daju klasifikatori h_1 i h_2. Instanci se dodeljuje klasa najveće verovatnoće

Parametri ko-treninga

- Zašto koristiti manji skup *U'*?
  - Empirijski je pokazano da njegova primena dovodi do boljih rezultata
  - Moguće objašnjenje jeste da ovo prisiljava klasifikatore h_1 i h_2 da odabiraju primere koji su reprezentativniji u odnosu na distribuciju koja generiše *U*

- Odabir p i n:
  - Odnos p/n trebao bi da se poklapa sa odnosom pozitivnih i negativnih primera u okviru distribucije originalnog skupa podataka
  - Ovo predstavlja potencijalni problem jer je poznata samo distribucija klasa malog anotiranog skupa L, što ne mora da se poklapa sa pravom distribucijom klasa u podacima

- Ko-treninga bi trebao da konvergira nakon određenog broja iteracija. Različiti autori koriste različite kriterijume zaustavljanja, npr. nekada se algoritam zaustavlja tek kada su svi primeri iz *unlabeled* skupa anotirani

Eksperiment

- 1051 *web* stranica sa 4 CS departmana
- Web stranice je potrebno razvrstati u 2 klase – pozitivnu (početna stranicu kursa) i negativnu (ostalo)

- Izvršena je podela na trening (75%) i test (25%) skup
- Trening podaci su podeljeni na *labeled set* (*L*) i *unlabeled set* (*U*)
  - *L:* slučajno odabranih 3 pozitivnih i 9 negativnih primera (očuvana je distribucija originalnog skupa podataka)
  - *U*: preostalim primerima je uklonjena klasna oznaka

Eksperiment

- Skup obeležja podeljen je na 2 pogleda:
  - reči koje se nalaze na web stranici
  - reči koje se nalaze u linkovima koji ukazuju na web stranicu

  (Za reprezentaciju teksta odabran je *bag-of-words* model u kome se zanemaruju gramatika i redosled reči)

- Obučavajući algoritam: *Naive Bayes*
  - Empirijski pokazan kao kvalitetan na različitim problemima klasifikacije teksta
  - Podrazumeva da je svaka reč u dokumentu nezavisna od ostalih, što ga čini lako primenljivim na *bag-of-words* reprezentaciju

- Ostali parametri: p=1; n=3; k=30; u=75

Rezultati

- Prosek *error rate* (1-*accuracy*) u 5 slučajnih training/test podela (manja vrednost znači tačniji klasifikator):

Obeležja: tekst stranice

Obeležja: tekst linkova

Sva obeležja kombinovana

NB klasifikator na L

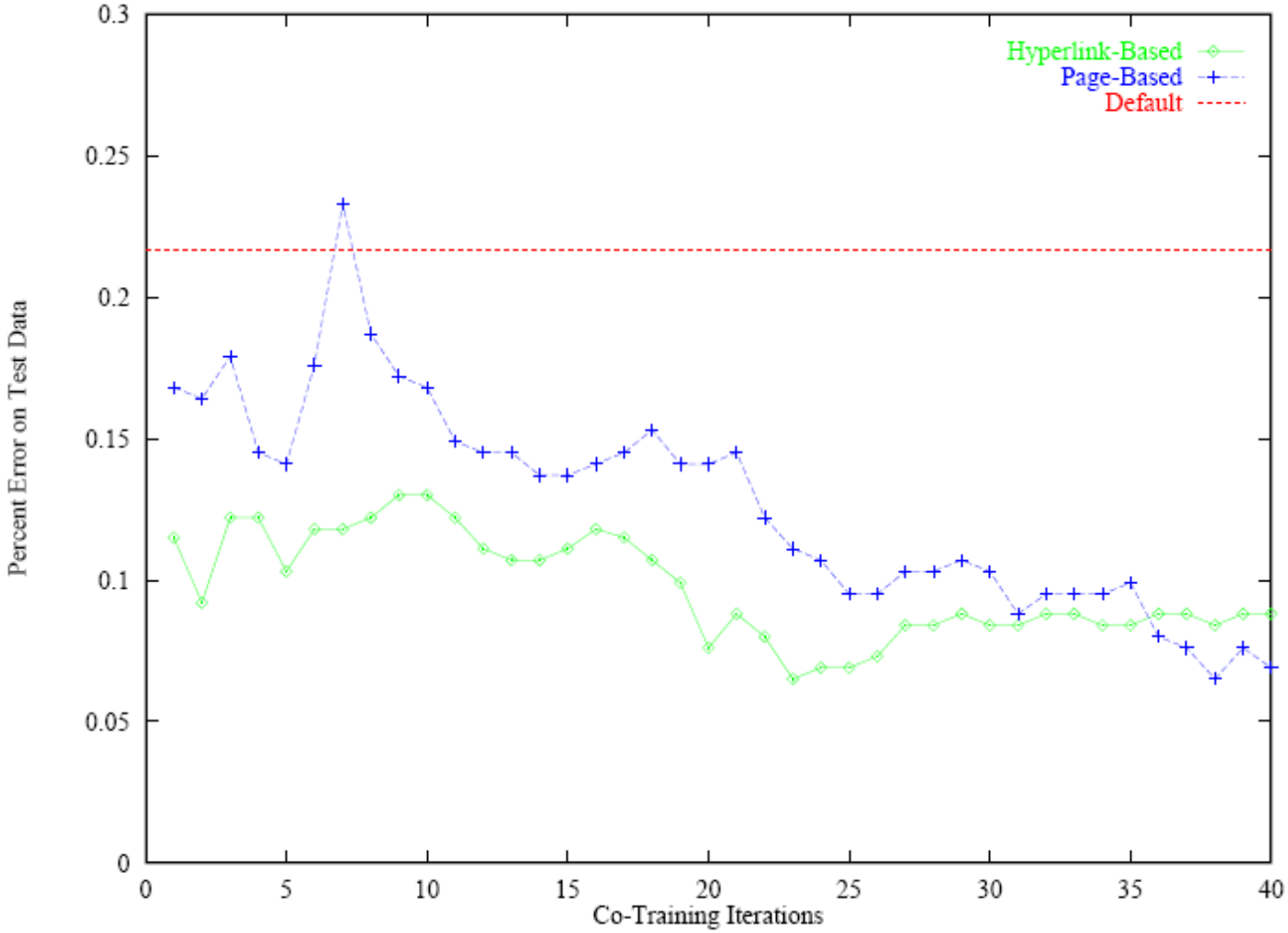
Co-training na L+U

Baseline klasifikator koji sve primere klasifikuje kao negativne

	Page-based	Hyperlink-based	Combined
Std. Supervised	12.9	12.4	11.1
Co-training	6.2	11.6	5.0
Just say neg	22	22	22

Za sve tipove klasifikatora, ko-trening je rezultovao povećanjem performansi u odnosu na polazni klasifikator

Rezultati – tačnost ko-treninga sa brojem iteracija



Zaključak

- Primenom ko-trening algoritma se može dobiti značajno pobojšanje tačnosti

- Prednosti:
  - Jednostavan metod koji se može kombinovati sa bilo kojim nadgledanim modelom
  - Manje osetljiv na greške od samo-obučavanja
- Mane:
  - Prirodna podela obeležja ne mora da postoji
  - Modeli koji koriste oba pogleda istovremeno bi trebali da budu bolji

Kada se ko-trening može primeniti?

- Može se primeniti na skupove podataka kod kojih postoji prirodna podela skupa obeležja na dva podskupa (pogleda)

- Uspešna primena ko-treninga diktira:
  1. da je svaki od pogleda dovoljan za kvalitetnu klasifikaciju
  2. da su pogledi međusobno nezavisni u odnosu na labelu (predstavljaju dva odvojena izvora informacija)

- Problem:
  - U praksi se veoma retko nailazi na prirodnu podelu obeležja koja ispunjava zadate uslove
  - Ovo čini ko-trening algoritam retko primenljivim u praksi, zbog čega su se mnogi istraživači posvetili problemu primene ko-treninga na skupove podataka bez prirodne podele obeležja