

Učenje koncepta

Matko Bošnjak, 2011

Uvod

- Centralni problem strojnog učenja: indukcija općih funkcija iz posebnih primjera
- Svi mi konstantno učimo neke koncepte i kategorije poput: “student”, “stol”, “riba”
- Svaki koncept se može promatrati kao opis (pod)skupa objekata ili događaja nad većim skupom
- Koncept – booleova funkcija definirana nad skupom (sjetite se (s mukom) $\forall x \text{ Student}(x)$)
 - Točno za studente, netočno za ostale vrijednosti domene
- **Učenje koncepata je zaključivanje/induciranje booleove funkcije iz skupa primjera za učenje koji predstavljaju njen ulaz i izlaz**

Zadatak učenja koncepta

- Primjer: naučiti koncept “dani kojima moj prijatelj Aldo uživa u svom omiljenom vodenom sportu”
- Atribut EnjoySport označava da li Aldo uživa u sportu ili ne na taj dan
- Zadatak je predvidjeti vrijednost EnjoySport za proizvoljan dan

Primjer	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Reprezentacija hipoteza

- Kako ćemo reprezentirati hipoteze?
- Jednostavna reprezentacija: svaka hipoteza je konjunkcija ograničenja na attribute instanci
- U ovom primjeru, hipoteza je vektor od šest vrijednosti ograničenja (*Sky*, *AirTemp*, *Humidity*, *Wind*, *Water* i *Forecast*)
- Za svaki atribut, hipoteza može imati neku od ovih vrijednosti:
 - “?” – bilo koja vrijednost je prihvatljiva
 - Jedinična vrijednost atributa (npr. “Sunny”, “Warm”)
 - “∅” – nijedna vrijednost nije prihvatljiva
- Ukoliko neka instanca x zadovoljava sva ograničenja hipoteze h , tada h klasificira x kao pozitivan primjer: **$h(x) = 1$**
- Npr:

<?, Cold, High, ?, ?, ?>

Reprezentacija hipoteza

- Najgeneralnija hipoteza – svaki dan je dan za sport

$$h = \langle ?, ?, ?, ?, ?, ? \rangle$$

- Najspecifičnija hipoteza – nijedan dan nije za sport

$$h = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$$

- Generalno se svaki zadatak učenja koncepata može opisati sa:
 - skupom instanci nad kojima je definirana funkcija cilja
 - funkcijom cilja
 - skupom kandidata hipoteza koje uopće dolaze u razmatranje
 - skupom primjera za učenje

Terminologija

- Skup objekata nad kojima je koncept definiran (skup instanci) – X
 - Reprezentiran atributima
- Instanca tj. primjer je konjunkcija vrijednosti atributa
- Ciljni koncept c je booleova funkcija definirana nad X kao $c : X \rightarrow \{0, 1\}$
 - To je funkcija koju želimo naučiti
- Skup za učenje se sastoji od instanci iz X , zajedno s pripadajućim $c(x)$ – D
 - Pozitivni primjeri ($c(x) = 1$)
 - Negativni primjeri ($c(x) = 0$)
 - Jedan primjer za učenje označavamo s $\langle x, c(x) \rangle$
- Zadatak je procijeniti c
- Skup svih dopustivih hipoteza koje možemo promatrati u sklopu problema učenja – H
- Određuje i opisuje ga čovjek koji odabire reprezentaciju
- h iz H je booleova funkcija definirana nad X , takva da je $h : X \rightarrow \{0, 1\}$
- Cilj je pronaći h takav da vrijedi $h(x) = c(x)$

Na našem primjeru

- Što imamo
 - Instance X : Mogući dani opisani atributima:
 - Sky (Sunny, Cloudy, Rainy)
 - AirTemp (Warm, Cold)
 - Humidity (Normal, High)
 - Wind (Strong, Weak)
 - Water (Warm, Cool)
 - Forecast (Same, CHange)
 - Hipoteze H : Svaka hipoteza opisana kao konjunkcija ograničenja na attribute
 - Ciljni koncept c : $EnjoySport: X \rightarrow \{0,1\}$
 - Skup primjera za učenje D : Pozitivni i negativni primjeri ciljne funkcije
- Što tražimo
 - Hipotezu h iz H tako da vrijedi $h(x) = c(x)$ za svaki $x \in X$

Hipoteza induktivnog učenja

- Zadatak je odrediti h identičnu c -u, ali na ograničenom broju instanci (jer to je sve što imamo za učenje)
- Najbolje što induktivni algoritam za učenje može garantirati je da će se dobivena hipoteza najbolje slagati s primjerima za učenje, i to je sve.
- Hipoteza induktivnog učenja:
Bilo koja hipoteza koja dobro aproksimira ciljnu funkciju nad dovoljno velikim skupom primjera za učenje, također će dobro aproksimirati ciljnu funkciju na neviđenim primjerima

Učenje koncepata kao pretraga

- Učenje koncepata možemo promatrati kao pretragu kroz veliki skup hipoteza implicitno definiranih svojom reprezentacijom
- Cilj je pronaći hipotezu koja se najbolje slaže s raspoloživim podacima (primjerima za učenje)
- Odabirom reprezentacije dizajner implicitno definira skup hipoteza koji se može pretraživati tj. učiti – i to je domena našeg učenja
- Pogledajmo naš primjer
 - 6 atributa, broj vrijednosti: 3,2,2,2,2,2
 - Ukupno $3*2*2*2*2*2 = 96$ mogućih instanci
 - $5*4*4*4*4*4 = 5120$ sintaktički različitih hipoteza
 - No samo $1 + (4*3*3*3*3*3) = 973$ semantički različitih
 - A ovo je prilično jednostavan primjer
- Tražimo efikasno pretraživanje velikih i/ili beskonačnih prostora hipoteza

Poredak hipoteza

- U ovom našem prostoru hipoteza postoji postoji interesantna struktura: poredak posebno-prema-općem (eng. general-to-specific ordering)
- Naravno da ćemo to iskoristiti za pretragu
 - Možemo pretraživati beskonačne prostore bez eksplicitnog pobrojavanja svake hipoteze
- Primjer:

$$h_1 = \langle \text{Sunny}, ?, ?, \text{Strong}, ?, ? \rangle$$

$$h_2 = \langle \text{Sunny}, ?, ?, ?, ?, ? \rangle$$

- h_2 manje ograničava instancu
 - više instanci će klasificirati kao pozitivne
 - generalnija je od h_1 (svaka instanca koju h_1 klasificira kao pozitivnu, h_2 će također klasificirati kao pozitivnu)

Poredak hipoteza

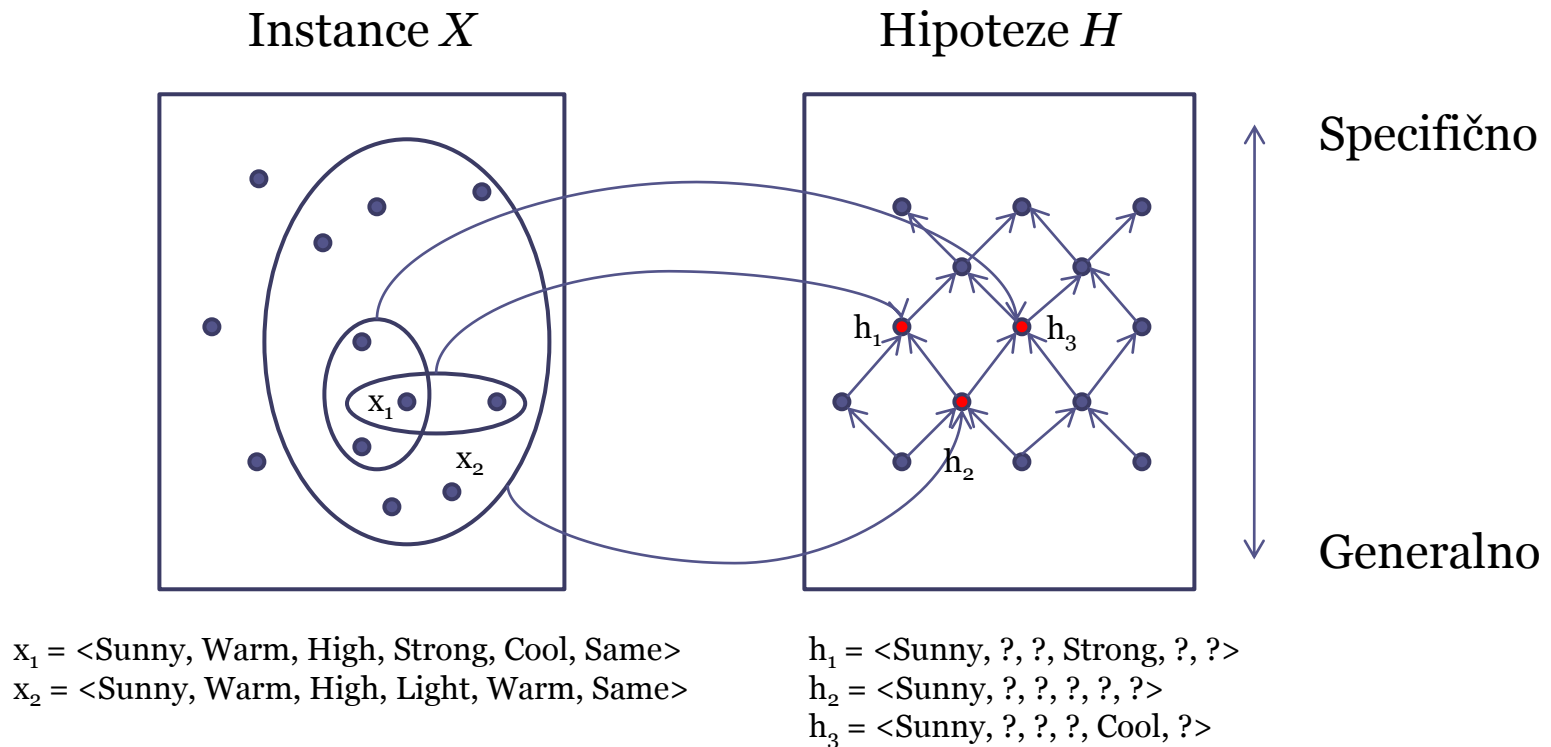
- **Definicija:** Neka su h_j i h_k booleove funkcije definirane nad X . h_j je **generalnija_ili_jednako_generalna_od** h_k ($h_j \geq_g h_k$) akko

$$(\forall x \in X) [(h_k(x) = 1) \rightarrow (h_j(x) = 1)]$$

- Relacija definira parcijalno uređen skup (refleksivna, antisimetrična i tranzitivna)
- Ukoliko uvodimo striktnost u poredak:

$$h_j >_g h_k \text{ akko } (h_j \geq_g h_k) \wedge (h_k \not\geq_g h_j)$$

Poredak hipoteza, primjer



Find-S

- Započnimo s najspecifičnijom hipotezom i generalizirajmo ju (“oslabljujmo ju”) svaki puta kada ne uspije dobro klasificirati pozitivan primjer za učenje

FIND-S

```
inicijaliziraj  $h$  na najspecifičniju hipotezu iz  $H$ 
```

```
za svaki pozitivan primjer  $x$ 
```

```
    ako  $x$  zadovoljava ograničenje  $a_i$ 
```

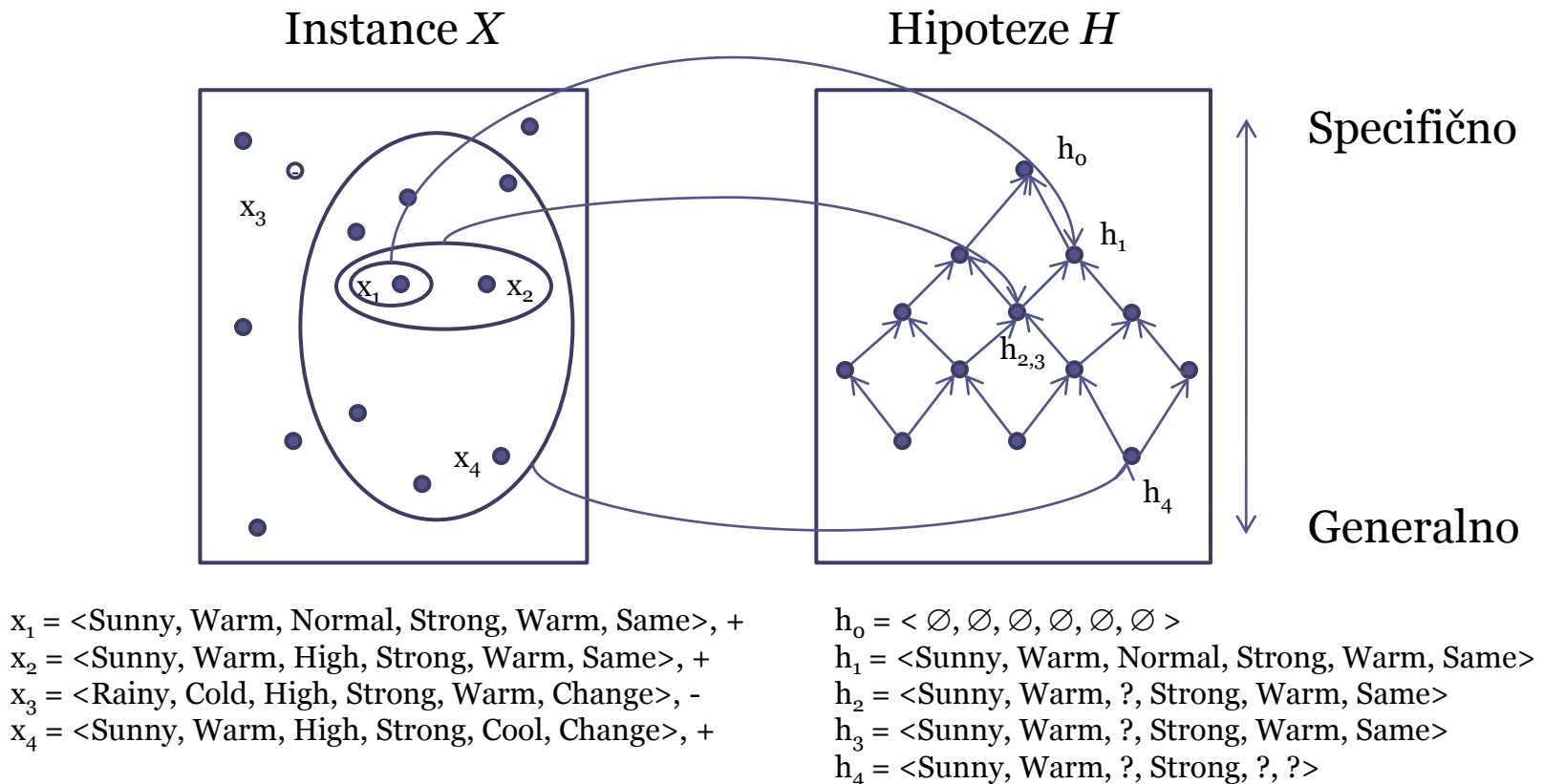
```
        ne čini ništa
```

```
    inače
```

```
        zamijeni  $a_i$  u  $h$  sa sljedećim generalnijim ograničenjem  
        koje  $x$  zadovoljava
```

```
vraati  $h$  kao rezultat
```

Find-S, primjer



Find-S (komentari)

- Algoritam ignorira svaki negativan primjer
 - Pretpostavke da se H sadrži hipotezu koja može opisati c i da podaci za učenje ne sadržavaju greške
 - c se nalazi u H po pretpostavci, konzistentan sa D , mora biti generalniji ili jednako generalan s h . No c nikad neće pokriti negativne primjere, pa neće ni h
- Za prostor hipoteza definiran kao konjunkcija ograničenja na attribute, Find-S garantirano pronalazi najspecifičniju hipotezu u H koja je konzistentna s primjerima za učenje
- Bit će konzistentna i s negativnima pod uvjetom da se ispravan ciljni koncept nalazi u H i da nema grešaka u primjerima za učenje
- Pitanja
 - Da li je algoritam konvergirao u točan ciljni koncept (pronaći će konzistentnu hipotezu, ali ne zna da li je pronašao jedinu te da li postoje druge)
 - Zašto preferirati najspecifičnije hipoteze - on će pronaći najspecifičniju, ali što s ostalima?
 - Da li su primjeri za učenje konzistentni? – greške, šum?
 - Što ako postoji nekoliko maksimalno specifičnih konzistentnih hipoteza?

Prostor konzistentnih hipoteza

- Želimo algoritam koji će nam ispisati sve hipoteze konzistentne s primjerima za učenje, ne samo najspecifičniju kao Find-S
- Koristeći parcijalno uređenje **generalniji_od**, možemo pretražiti prostor svih hipoteza bez potrebe za enumeracijom svih hipoteza
- Candidate-Elimination algoritam
- No prije toga, “nova” reprezentacija

Reprezentacija

- Candidate-Elimination algoritam pronalazi sve hipoteze koje su konzistentne s primjerima za učenje
- **Definicija:** Hipoteza h je konzistentna sa skupom primjera za učenje D ako i samo ako je $h(x) = c(x)$ za svaki primjer $\langle x, c(x) \rangle$ iz D

$$\text{Consistent}(h, D) \equiv (\forall \langle x, c(x) \rangle \in D) h(x) = c(x)$$

- Konzistentnost i zadovoljivost nisu iste ⁽⁴⁾

Reprezentacija

- Candidate-Elimination algoritam predstavlja skup svih hipoteza koje su konzistentne s primjerima za učenje
- Taj skup zovemo prostorom konzistentnih hipoteza (eng. version space) s obzirom na prostor hipoteza H i primjere za učenje D jer sadrži sve prihvatljive verzije ciljnog koncepta
- **Definicija:** Prostor konzistentnih hipoteza $VS_{H,D}$ s obzirom na prostor hipoteza H i primjere za učenje D , je podskup hipoteza iz H konzistentnih s primjerima za učenje iz D

$$VS_{H,D} \equiv \{h \in H \mid \text{Consistent}(h, D)\}$$

List-Then-Eliminate algoritam

- “Brute force napad na prostor hipoteza”
 - Enumerirajmo sve hipoteze prostora konzistentnih hipoteza

LIST-THEN-ELIMINATE

```
ProstorKonzistentnihHipoteza ← popis svih hipoteza iz H
za svaki primjer za učenje  $\langle x, c(x) \rangle$ 
    ukloni iz ProstorKonzistentnihHipoteza sve hipoteze  $h$  koje
    nisu konzistentne ( $h(x) \neq c(x)$ )
vraati ProstorKonzistentnihHipoteza kao rezultat
```

- Problem je dakako jako velik do beskonačan H

Kompaktnija reprezentacija

- Candidate-Elimination radi kao i List-Then-Eliminate, međutim ipak je malo pametniji
- Koristi kompaktniju reprezentaciju prostora konzistentnih hipoteza
 - Reprezentiran najgeneralnijim i najmanje generalnim hipotezama
 - Tvore generalne i specifične granice skupa koje ograničavaju prostor konzistentnih hipoteza u sklopu parcijalnog uređenja prostora hipoteza

Kompaktnija reprezentacija

- **Definicija:** Generalna granica G , s obzirom na prostor hipoteza H i skup za učenje D je skup maksimalno generalnih članova H konzistentnih sa D

$$G \equiv \{g \in H \mid \text{Consistent}(g, D) \wedge (\neg \exists g' \in H) [(g' >_g g) \wedge \text{Consistent}(g', D)]\}$$

- **Definicija:** Specifična granica S s obzirom na prostor hipoteza H i skup za učenje D je set minimalno generalnih (maksimalno specifičnih) članova H konzistentnih sa D

$$S \equiv \{s \in H \mid \text{Consistent}(s, D) \wedge (\neg \exists s' \in H) [(s >_g s') \wedge \text{Consistent}(s', D)]\}$$

Kompaktnija reprezentacija

- **Teorem reprezentacije prostora konzistentnih hipoteza**
- Neka je X proizvoljan skup instanci te H skup booleovih hipoteza definiranih nad X . Neka je $c: X \rightarrow \{0, 1\}$ proizvoljan ciljni koncept definiran na X te D proizvoljan skup primjera za učenje $\{ \langle x, c(x) \rangle \}$. Za sve X, H, c i D takve da su S i G dobro definirani vrijedi:

$$VS_{H,D} = \{h \in H \mid (\exists s \in S)(\exists g \in G)(g \geq_g h \geq_g s)\}$$

- Dakle, prostor konzistentnih hipoteza se nalazi između generalne granice G i specifične granice S

Candidate-Elimination algoritam

CANDIDATE-ELIMINATION

```
inicijaliziraj G na skup maksimalno generalnih hipoteza u H
inicijaliziraj S na skup maksimalno specifičnih hipoteza u H
za svaki primjer za učenje d
    ako je d pozitivan primjer
        iz G ukloni sve hipoteze nekonzistentne s d
        za svaku hipotezu s u S koja nije konzistentna s d
            ukloni s iz S
            dodaj u S sve minimalne generalizacije h od s takve da je
                | h konzistentna s d i neki članovi G-a generalniji od h
            ukloni iz S sve hipoteze generalnije od ostalih hipoteza u S
    ako je d negativan primjer
        iz S ukloni sve hipoteze nekonzistentne s d
        za svaku hipotezu g u G koja nije konzistentna s d
            ukloni g iz G
            dodaj u G sve minimalne specijalizacije h od g takve da je
                | h konzistentna s d i neki članovi S-a specifičniji od h
            ukloni iz G sve hipoteze manje generalne od ostalih hipoteza u G
```

Candidate-Elimination primjer

S_0 $\{ \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle \}$

G_0 $\{ \langle ?, ?, ?, ?, ?, ? \rangle \}$

Candidate-Elimination primjer

$x_1 = \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle, +$

S_1

$\{ \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle \}$

G_1

$\{ \langle ?, ?, ?, ?, ?, ? \rangle \}$

Candidate-Elimination primjer

$x_2 = \langle \text{Sunny, Warm, High, Strong, Warm, Same} \rangle, +$

S_2

$\{ \langle \text{Sunny, Warm, ?, Strong, Warm, Same} \rangle \}$

G_2

$\{ \langle ?, ?, ?, ?, ?, ? \rangle \}$

Candidate-Elimination primjer

$x_3 = \langle \text{Rainy, Cold, High, Strong, Warm, Change} \rangle, -$

S_3 $\{ \langle \text{Sunny, Warm, ?, Strong, Warm, Same} \rangle \}$

G_3 $\{ \langle \text{Sunny, ?, ?, ?, ?, ?} \rangle, \langle \text{?, Warm, ?, ?, ?, ?} \rangle, \langle \text{?, ?, ?, ?, ?, Same} \rangle \}$

Candidate-Elimination primjer

$x_4 = \langle \text{Sunny, Warm, High, Strong, Cool, Change} \rangle, +$

S_4

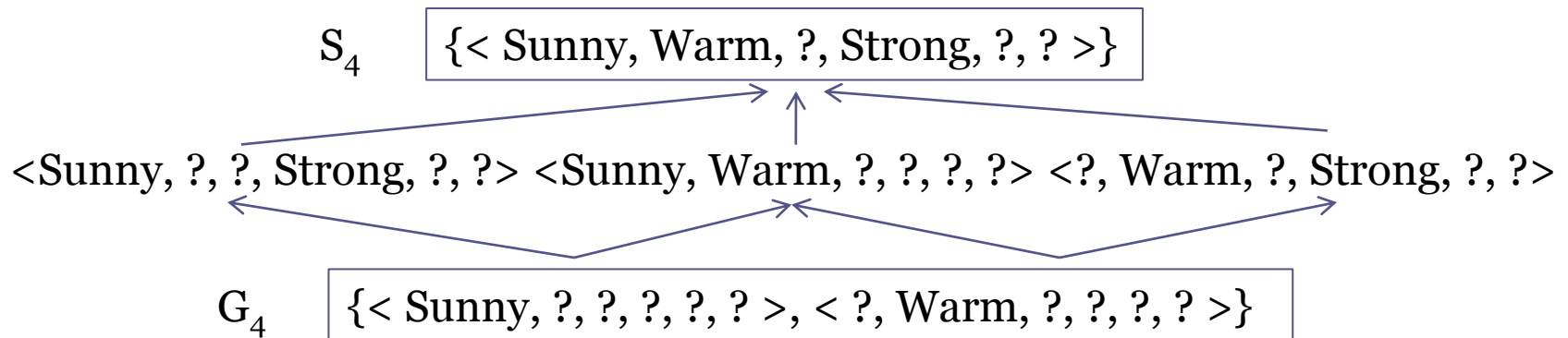
$\{ \langle \text{Sunny, Warm, ?, Strong, ?, ?} \rangle \}$

G_4

$\{ \langle \text{Sunny, ?, ?, ?, ?, ?} \rangle, \langle \text{?, Warm, ?, ?, ?, ?} \rangle \}$

Candidate-Elimination primjer

Konačan prostor konzistentnih hipoteza za naš primjer



Candidate-Elimination komentari

- Da li će Candidate-Elimination algoritam konvergirati u točnu hipotezu?
 - Hoće, pod uvjetom da
 - Ne postoje pogreške u skupu za učenje
 - U prostoru hipoteza H postoji hipoteza koja može opisati ciljni koncept
- Kakav bi primjer za učenje algoritam mogao zatražiti
 - Najinformativnije, onaj primjer koji pola hipoteza skupa konzistentnih hipoteza proglašava pozitivnim, a pola negativnim
 - Primjer: <Sunny, Warm, Normal, Light, Warm, Same>
- Parcijalno naučeni koncepti
 - Možemo ih iskoristiti za glasanje, time možemo dobiti pouzdanost/vjerojatnost klasifikacije

Induktivna pristranost

- Već spomenuto, Candidate-Elimination će konvergirati prema ciljnom konceptu ako su primjeri za učenje ispravni te prostor hipoteza sadrži ciljni koncept
- Što ako ciljni koncept nije u prostoru hipoteza?

Primjer	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Cool	Change	Yes
2	Cloudy	Warm	Normal	Strong	Cool	Change	Yes
3	Rainy	Warm	Normal	Strong	Cool	Change	No

Nepristrano učenje

- Ok, proširimo prostor hipoteza na prostor svih mogućih hipoteza, ne samo konjunkciju ograničenja ($2^{|X|}$ elemenata)
- U našem primjeru to je prostor od 2^{96} hipoteza, a naš dosadašnji pristrani prostor je imao 973 hipoteze!
- Sada možemo predstaviti koncept

$$\text{Sky} = \text{Sunny} \vee \text{Sky} = \text{Cloudy}$$

$$\langle \text{Sky}, ?, ?, ?, ?, ? \rangle \vee \langle \text{Cloudy}, ?, ?, ?, ?, ? \rangle$$

- Uzmimo 3 pozitivna i 2 negativna primjera te pripadajuće S i G granice

$$S = \{(x_1 \vee x_2 \vee x_3)\}$$

$$G = \{\neg(x_4 \vee x_5)\}$$

najspecifičnija granica

Nepristrano učenje

- Problem je da će svaka S granica uvijek biti disjunkcija promatranih pozitivnih primjera, a G negativna disjunkcija promatranih negativnih primjera
- Samo ćemo za primjere za učenje znati klasifikaciju, za ostale ne
- Nema nikakve generalizacije
- Glasanje svih ostalih hipoteza ne pomaže jer ih ima točno pola koji će glasati za pozitivnu, a točno pola za negativnu klasifikaciju (za svaki primjer postoji hipoteza h koja ga pokriva i hipoteza identična toj hipotezi, osim u rezultatu nad x)

Uzaludnost nepristranog učenja

- x_i – nepoznat primjer
- D_c – podaci za učenje
- $L(x_i, D_c)$ je klasifikacija koju algoritam L dodjeljuje nepoznatom primjeru x_i nakon učenja nad podacima D_c
- Klasifikacija je induktivno zaključena iz nepoznatog primjera i podataka za učenje

$$(D_c \wedge x_i) \succ L(x_i, D_c)$$

- Klasifikacija je deduktivno zaključena iz nepoznatog primjera, podataka za učenje i skupa dodatnih tvrdnji B koje su dovoljne da bismo induktivno zaključivanje prikazali kao deduktivno (to je induktivna pristranost)

$$(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)$$

Uzaludnost nepristranog učenja

- **Definicija:** Uzmimo na razmatranje algoritam učenja koncepata L na skupu primjera X . Neka je c proizvoljan koncept definiran nad X i $D_c = \{ \langle x, c(x) \rangle \}$ skup primjera za učenje. Neka je $L(x_i, D_c)$ klasifikacija novog primjera x_i algoritmom L nakon učenja nad podacima D_c . Induktivna pristranost algoritma L je minimalni skup tvrdnji B takvih da za bilo koji ciljni koncept c i pripadajuć mu skup za učenje D_c vrijedi:

$$(\forall x_i \in X) [(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)]$$

Induktivna pristranost, primjer

- Rote-Learner
 - Sačuvaj u memoriju sve primjere za učenje (štreberko)
 - Ako instanca postoji u memoriji, vrati njenu klasifikaciju, inače ništa
 - Nema induktivne pristranosti (nema generalizacije)
- Candidate-Elimination
 - Izračunaj VS
 - Nove instance klasificiraj prema slaganju s prostorom konzistentnih hipoteza, inače ništa
 - Ciljni koncept se može reprezentirati u prostoru hipoteza
- Find-S
 - Pronađi najspecifičniju hipotezu konzistentnu s primjerima za učenje
 - Nove instance klasificiraj prema maksimalno specifičnoj hipotezi
 - Također, ciljni koncept se može reprezentirati u prostoru hipoteza, sve instance su negativne osim ako se sigurno zna drugačije