

UČENJE NA PRIMJERIMA (PROMATRANJEM)

POGLAVLJE 18.1–3

Prema slajdovima Stuarta Russella (hvala)!

Sadržaj

- ◇ Agenti za učenje
- ◇ Induktivno učenje
- ◇ Učenje korištenjem stabla odlučivanja
- ◇ Mjerenje uspjeha učenja

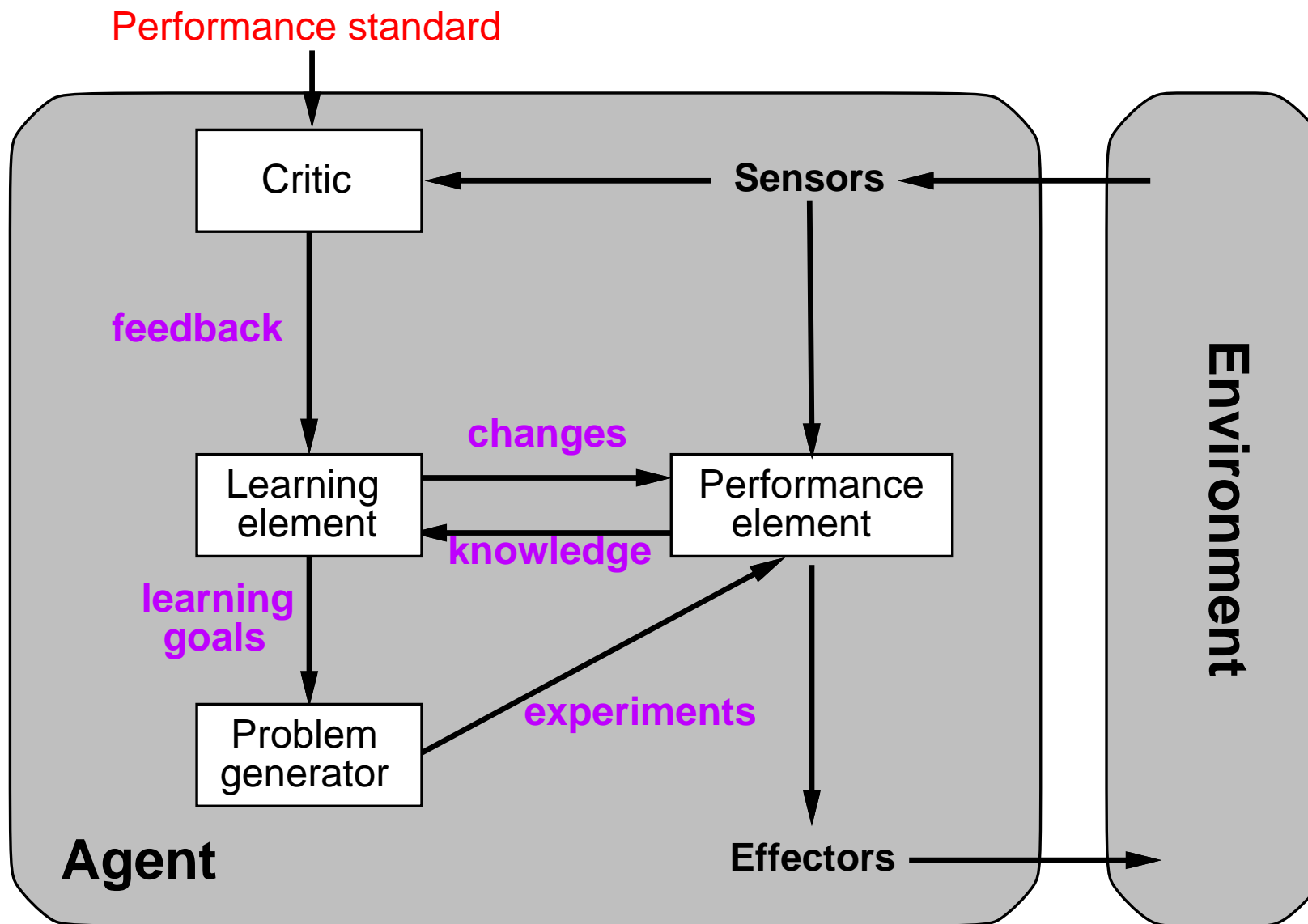
Učenje

Učenje je bitno u nepoznatim sredinama,
na primjer, kada dizajneru nedostaje potrebno znanje

Učenje je korisno kao metoda konstrukcije sustava,
na primjer, agent se izloži stvarnoj situaciji, umjesto da to pokuša
opisati

Učenje mijenja agentov mehanizam odlučivanja za poboljšanje izvođenja

Agenti za učenje



Elementi učenja

Dizajn elementa učenja diktiraju

- ◇ kakvu vrstu učinkovitosti elementa koristi
- ◇ koju funkcionalnu komponentu treba naučiti
- ◇ kako se reprezentira funkcionalna komponenta
- ◇ kakva je dostupna povratna informacija

Primjeri scenarija:

Performance element	Component	Representation	Feedback
Alpha-beta search	Eval. fn.	Weighted linear function	Win/loss
Logical agent	Transition model	Successor-state axioms	Outcome
Utility-based agent	Transition model	Dynamic Bayes net	Outcome
Simple reflex agent	Percept-action fn	Neural net	Correct action

Učenje pod nadzorom: svaki put točni odgovori

Pojačavanje učenja: povremene nagrade

Induktivno učenje (poznato i kao znanost)

Najjednostavnija forma: naučiti funkciju iz primjera (**tabula rasa**)

f je funkcija cilja

Primjer je par $x, f(x)$, na primjer,

O	O	X
	X	
X		

, $+1$

Problem: pronaći (n) hipoteza h takvih da $h \approx f$
ako je dan skup primjera za treniranje

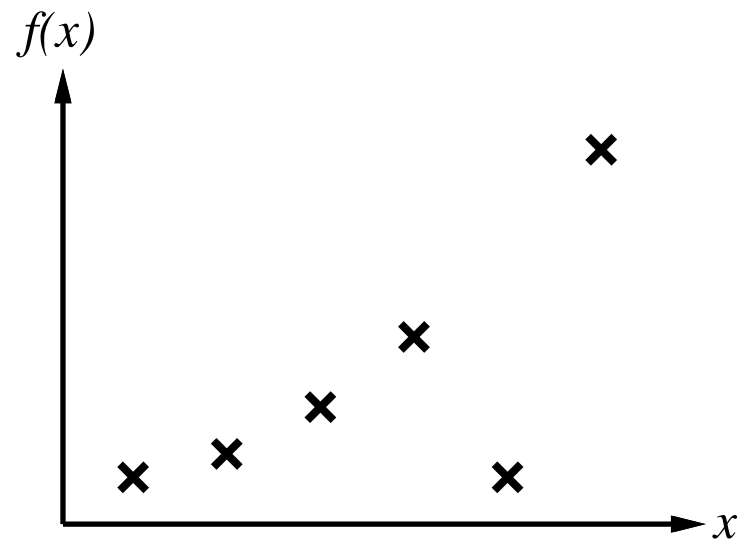
(**To je visoko pojednostvaljen model stvarnog učenja:**

- ignorira prijašnje znanje
- pretpostavlja determinističku, vidljivu “okolinu”
- pretpostavlja da su dani primjeri
- pretpostavlja da agent želi učiti f —zašto?)

Metoda induktivnog učenja

Konstruiraj/prilagodi h da se slaže s f na skupu podataka za treniranje (h je konzistentna ako se slaže s f na svim primjerima)

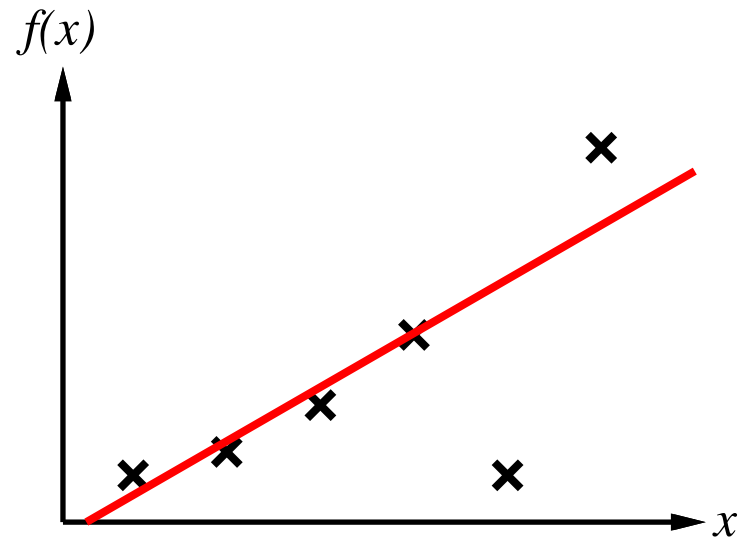
Na primjer, aproksimacija krivulja:



Metoda induktivnog učenja

Konstruiraj/prilagodi h da se slaže s f na skupu podataka za treniranje (h je konzistentna ako se slaže s f na svim primjerima)

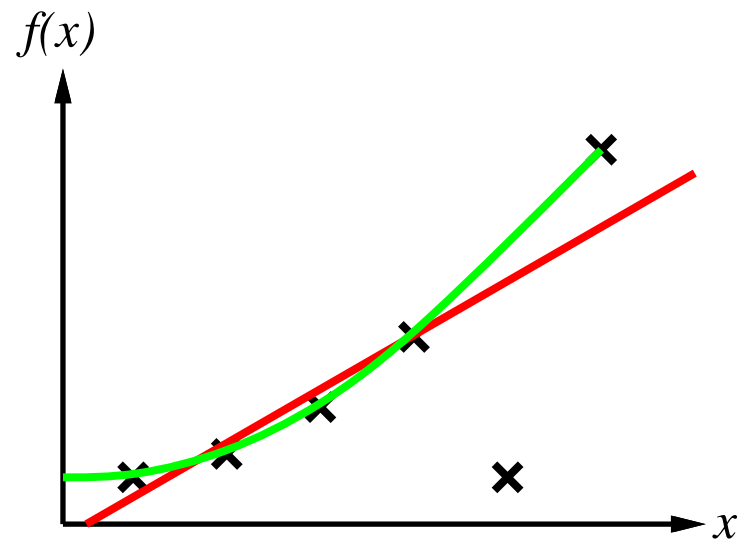
Na primjer, aproksimacija krivulja:



Metoda induktivnog učenja

Konstruiraj/prilagodi h da se slaže s f na skupu podataka za treniranje (h je konzistentna ako se slaže s f na svim primjerima)

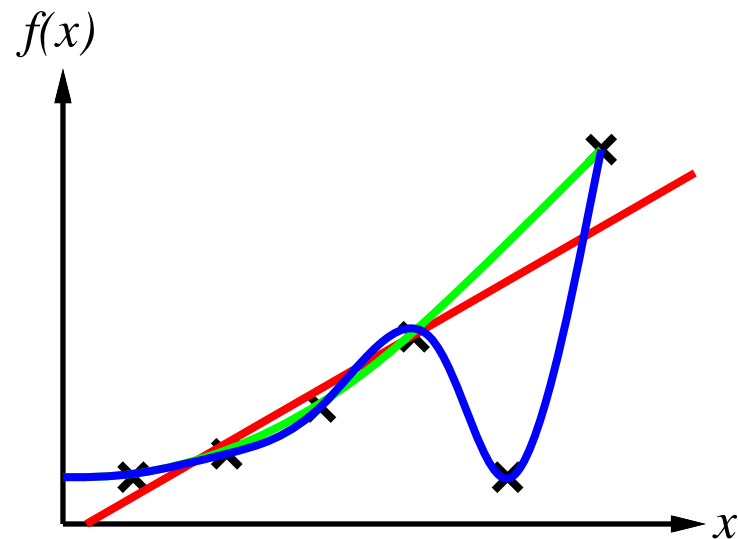
Na primjer, aproksimacija krivulja:



Metoda induktivnog učenja

Konstruiraj/prilagodi h da se slaže s f na skupu podataka za treniranje (h je konzistentna ako se slaže s f na svim primjerima)

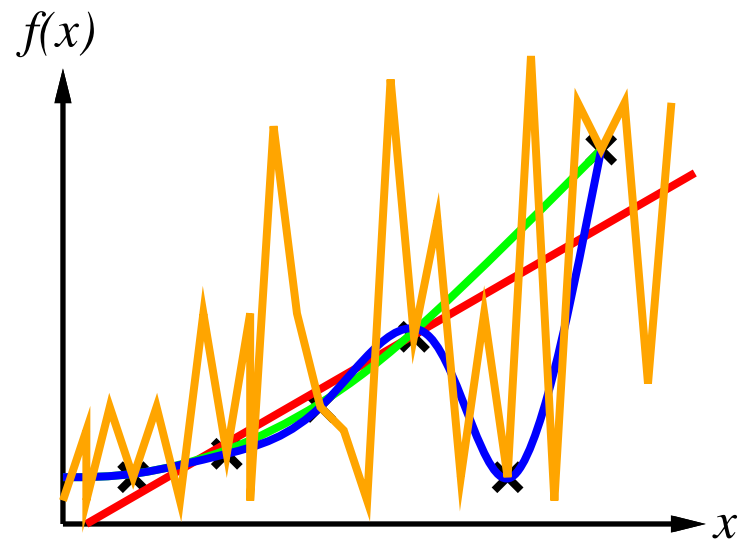
Na primjer, aproksimacija krivulja:



Metoda induktivnog učenja

Konstruiraj/prilagodi h da se slaže s f na skupu podataka za treniranje (h je konzistentna ako se slaže s f na svim primjerima)

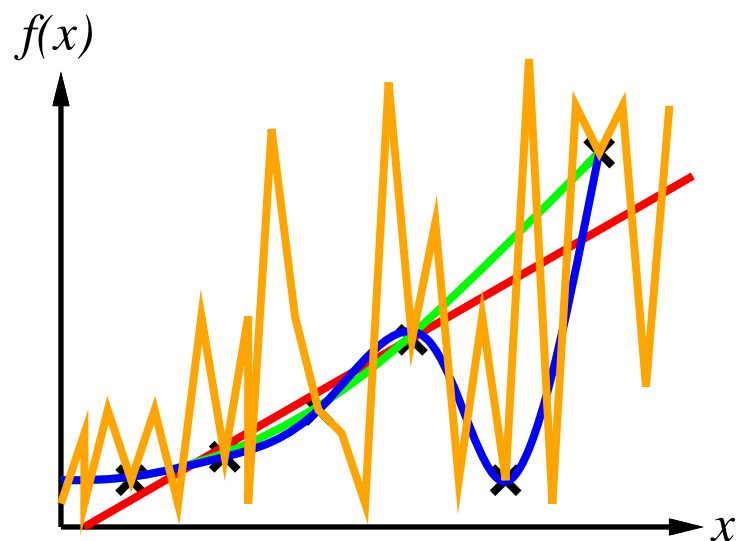
Na primjer, aproksimacija krivulja:



Metoda induktivnog učenja

Konstruiraj/prilagodi h da se slaže s f na skupu podataka za treniranje (h je konzistentna ako se slaže s f na svim primjerima)

Na primjer, aproksimacija krivulja:



Ockhamova britva:

maksimizirati kombinaciju konzistentnosti i jednostavnosti

Reprezentacija bazirana na atributima

Primjeri dani **vrijednostima atributa** (Boolovi, diskretni, neprekidni, ...)
 Na primjer, situacija želim li/ili ne čekati na stol u restoranu:

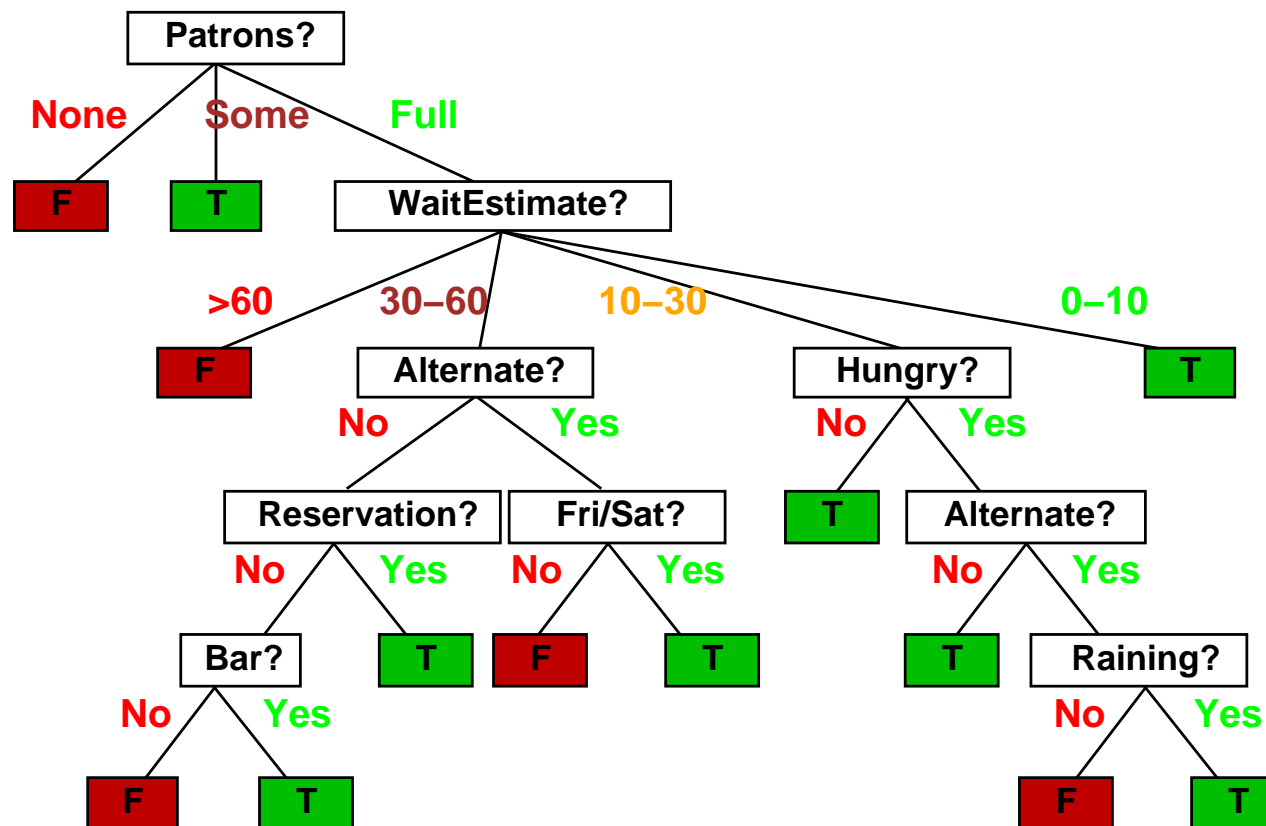
Example	Attributes										Target
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>WillWait</i>
X_1	<i>T</i>	<i>F</i>	<i>F</i>	<i>T</i>	<i>Some</i>	<i>\$\$\$</i>	<i>F</i>	<i>T</i>	<i>French</i>	<i>0-10</i>	<i>T</i>
X_2	<i>T</i>	<i>F</i>	<i>F</i>	<i>T</i>	<i>Full</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Thai</i>	<i>30-60</i>	<i>F</i>
X_3	<i>F</i>	<i>T</i>	<i>F</i>	<i>F</i>	<i>Some</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Burger</i>	<i>0-10</i>	<i>T</i>
X_4	<i>T</i>	<i>F</i>	<i>T</i>	<i>T</i>	<i>Full</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Thai</i>	<i>10-30</i>	<i>T</i>
X_5	<i>T</i>	<i>F</i>	<i>T</i>	<i>F</i>	<i>Full</i>	<i>\$\$\$</i>	<i>F</i>	<i>T</i>	<i>French</i>	<i>>60</i>	<i>F</i>
X_6	<i>F</i>	<i>T</i>	<i>F</i>	<i>T</i>	<i>Some</i>	<i>\$\$</i>	<i>T</i>	<i>T</i>	<i>Italian</i>	<i>0-10</i>	<i>T</i>
X_7	<i>F</i>	<i>T</i>	<i>F</i>	<i>F</i>	<i>None</i>	<i>\$</i>	<i>T</i>	<i>F</i>	<i>Burger</i>	<i>0-10</i>	<i>F</i>
X_8	<i>F</i>	<i>F</i>	<i>F</i>	<i>T</i>	<i>Some</i>	<i>\$\$</i>	<i>T</i>	<i>T</i>	<i>Thai</i>	<i>0-10</i>	<i>T</i>
X_9	<i>F</i>	<i>T</i>	<i>T</i>	<i>F</i>	<i>Full</i>	<i>\$</i>	<i>T</i>	<i>F</i>	<i>Burger</i>	<i>>60</i>	<i>F</i>
X_{10}	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>Full</i>	<i>\$\$\$</i>	<i>F</i>	<i>T</i>	<i>Italian</i>	<i>10-30</i>	<i>F</i>
X_{11}	<i>F</i>	<i>F</i>	<i>F</i>	<i>F</i>	<i>None</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Thai</i>	<i>0-10</i>	<i>F</i>
X_{12}	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>Full</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Burger</i>	<i>30-60</i>	<i>T</i>

Klasifikacija primjera je **pozitivna** (T) ili **negativna** (F)

Stabla odlučivanja

Jedna od mogućih reprezentacija hipoteza

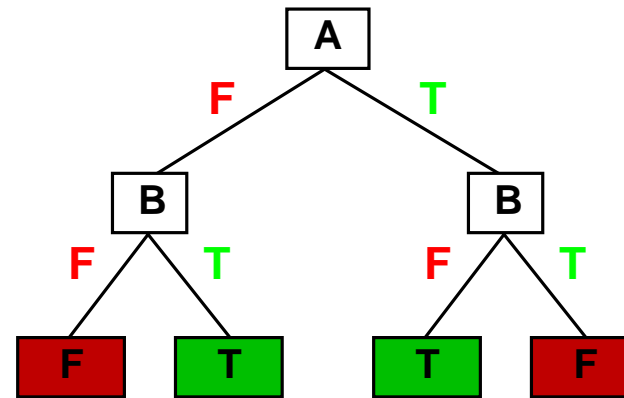
Na primjer, ovdje je “pravo” stablo za odluku treba li čekati ili ne:



Izražajnost

Stabla odlučivanja mogu izraziti bilo koju funkciju ulaznih atributa.
Na primjer, za Booleovu funkciju, istinit redak u tablici \rightarrow put do lista:

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



Trivijalno, postoji konzistentno stablo odlučivanja za
bilo koji skup podataka za treniranje
s jednim putom do lista za svaki primjer
(osim ako je f nondeterministička u x)
ali se to vjerojatno **neće** generalizirati na nove primjere.

Bolje bi bilo pronaći **kompaktnija** stabla odlučivanja

Prostori hipoteza

Koliko različitih stabala odlučivanja ima za n Booleovih atributa??

Prostori hipoteza

Koliko različitih stabala odlučivanja ima za n Booleovih atributa??

= broj Booleovih funkcija

Prostori hipoteza

Koliko različitih stabala odlučivanja ima za n Booleovih atributa??

= broj Booleovih funkcija

= broj različitih tablica istinitosti s 2^n redaka

Prostori hipoteza

Koliko različitih stabala odlučivanja ima za n Booleovih atributa??

= broj Booleovih funkcija

= broj različitih tablica istinitosti s 2^n redaka = 2^{2^n}

Prostori hipoteza

Koliko različitih stabala odlučivanja ima za n Booleovih atributa??

= broj Booleovih funkcija

= broj različitih tablica istinitosti s 2^n redaka = 2^{2^n}

Na primjer, sa 6 Booleovih atributa, ima 18 446 744 073 709 551 616 stabala

Prostori hipoteza

Koliko različitih stabala odlučivanja ima za n Booleovih atributa??

= broj Booleovih funkcija

= broj različitih tablica istinitosti s 2^n redaka = 2^{2^n}

Na primjer, sa 6 Booleovih atributa, ima 18 446 744 073 709 551 616 stabala

Koliko čisto konjunktivnih hipoteza (na pr., $Hungry \wedge \neg Rain$)??

Prostori hipoteza

Koliko različitih stabala odlučivanja ima za n Booleovih atributa??

= broj Booleovih funkcija

= broj različitih tablica istinitosti s 2^n redaka = 2^{2^n}

Na primjer, sa 6 Booleovih atributa, ima 18 446 744 073 709 551 616 stabala

Koliko čisto konjunktivnih hipoteza (na pr., $Hungry \wedge \neg Rain$)??

Svaki atribut može biti ulazni (pozitivan), ulazni (negativan), ili izlazni

$\Rightarrow 3^n$ različitih konjunktivnih hipoteza

Mnogo izražajniji prostor hipoteza

– povećava mogućnost za izražavanje funkcije cilja 😊

– povećava broj hipoteza konzistentnih sa skupom za treniranje

\Rightarrow može imati lošije predviđanje 😞

Učenje korištenjem stabla odlučivanja

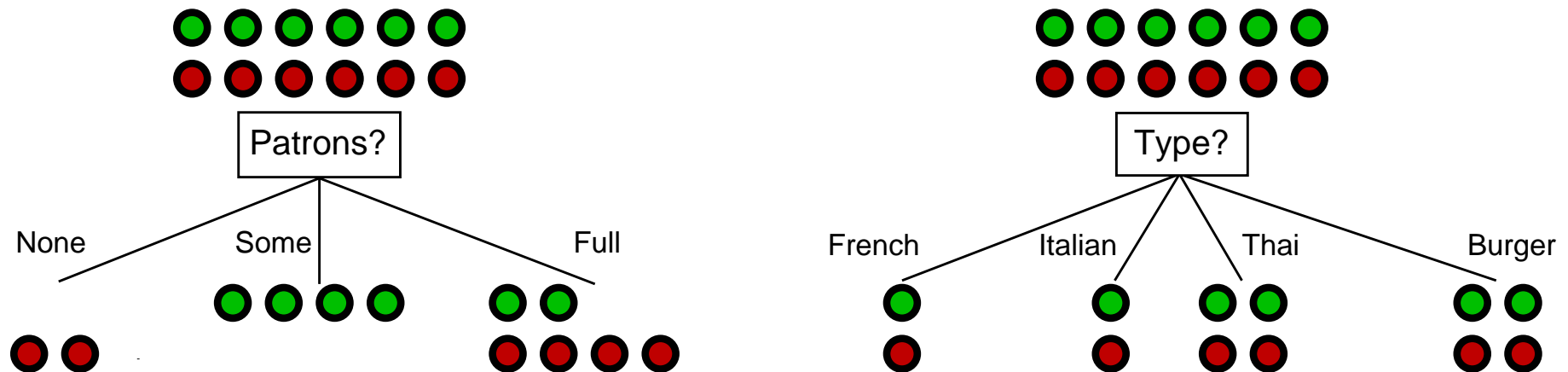
Cilj: pronaći malo stablo konzistentno s primjerima za treniranje

Idjea: (rekurzivno) izabrati “najznačajniji” atribut kao korijen (pod)stabla

```
function DTL(examples, attributes, default) returns a decision tree
  if examples is empty then return default
  else if all examples have the same classification then return the classification
  else if attributes is empty then return MODE(examples)
  else
    best ← CHOOSE-ATTRIBUTE(attributes, examples)
    tree ← a new decision tree with root test best
    for each value  $v_i$  of best do
       $examples_i$  ← {elements of examples with  $best = v_i$ }
      subtree ← DTL( $examples_i$ , attributes – best, MODE(examples))
      add a branch to tree with label  $v_i$  and subtree subtree
  return tree
```

Izbor atributa

Ideja: dobar atribut dijeli the primjere u podskupove koji su (idealno) “svi pozitivni” ili “svi negativni”



Patrons? je bolji izbor — daje **informaciju** o klasifikaciji.

Informacija

Informacija je odgovor na pitanja

Što je **manje** indicija o odgovoru na početku, **više** je informacija sadržano u odgovoru.

Skala: 1 bit = odgovor na Booleovo pitanje s *apriornom* vjerojatnošću $\langle 0.5, 0.5 \rangle$.

Informacija je odgovor kad je *apriorna* vjerojatnost $\langle P_1, \dots, P_n \rangle$

$$H(\langle P_1, \dots, P_n \rangle) = \sum_{i=1}^n -P_i \log_2 P_i$$

(također znano i **entropija** *apriorne* vjerojatnosti)

Informacija — nastavak

Pretpostavimo da imamo p pozitivnih i n negativnih primjera u korijenu
 $\Rightarrow H(\langle p/(p+n), n/(p+n) \rangle)$ bitova je potrebno
za klasifikaciju novog primjera

Na primjer za, 12 primjera restorana, $p = n = 6$ trebamo 1 bit.

Informacija — nastavak

Atribut dijeli primjere E na podskupove E_i , od kojih svaki (nadamo se) treba **manje** informacija za završetak klasifikacije.

Neka E_i imaju p_i pozitivnih i n_i negativnih primjera

⇒ $H(\langle p_i/(p_i + n_i), n_i/(p_i + n_i) \rangle)$ bitova je potrebno za klasifikaciju novog primjera

⇒ **očekivani** broj bitova po primjeru po svim granama je

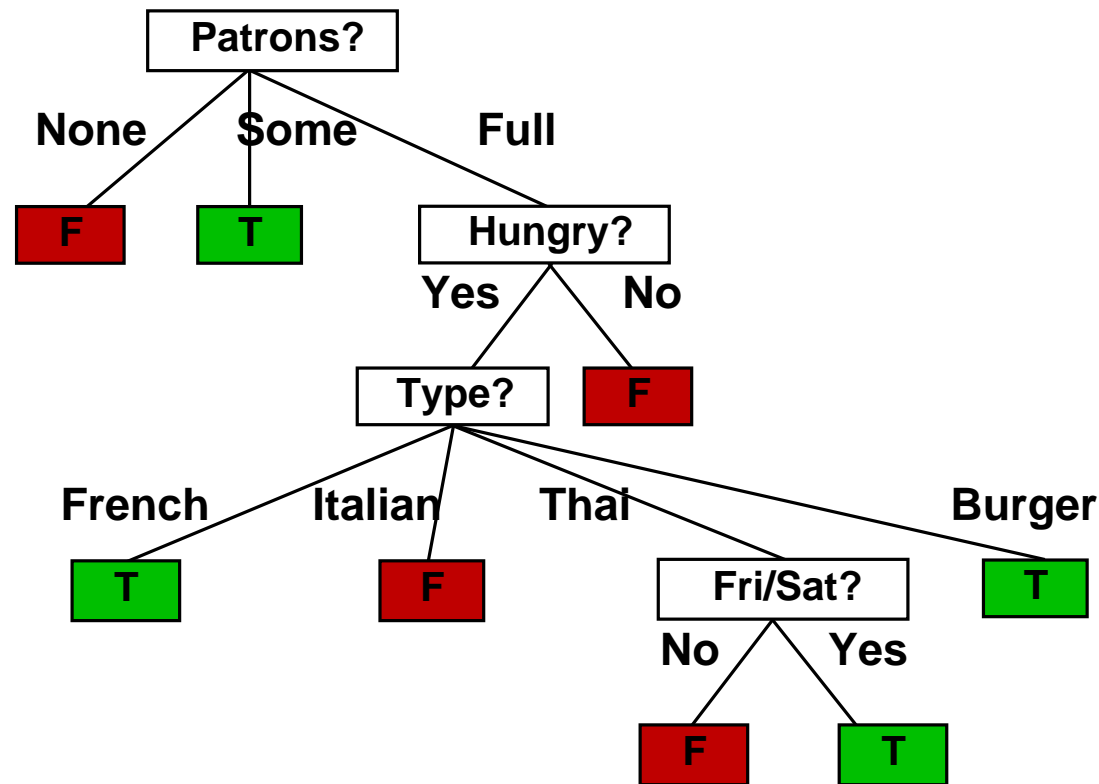
$$\sum_i \frac{p_i + n_i}{p + n} H(\langle p_i/(p_i + n_i), n_i/(p_i + n_i) \rangle).$$

Za *Patrons?*, to je 0.459 bitova, za *Type* to je (još uvijek) 1 bit.

⇒ treba izabrati atribut koji minimizira potrebne preostale informacije

Primjer — nastavak

Stablo odlučivanja dobiveno iz 12 primjera:



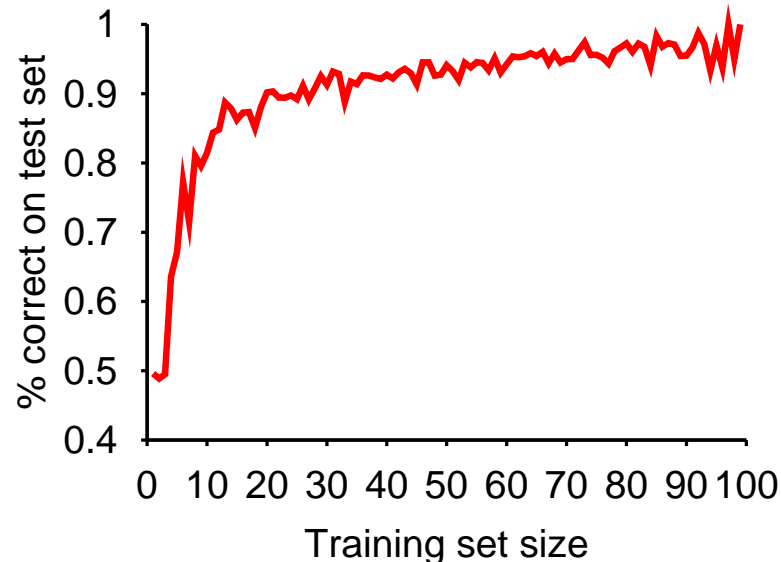
Znatno jednostavnije nego “pravo” stablo—složenije hipoteze nisu opravdane malom količinom podataka

Mjerenje uspjeha

Kako znamo da vrijedi $h \approx f$? (Humeov **Problem of Induction**)

- 1) Korištenjem teorema o računalnoj/statističkoj teoriji učenja
- 2) Iskoristi h na novom **testnom skupu** primjera
(upotrijebiti **istu distribuciju za cijeli prostor uzoraka** kao skupa za treniranje)

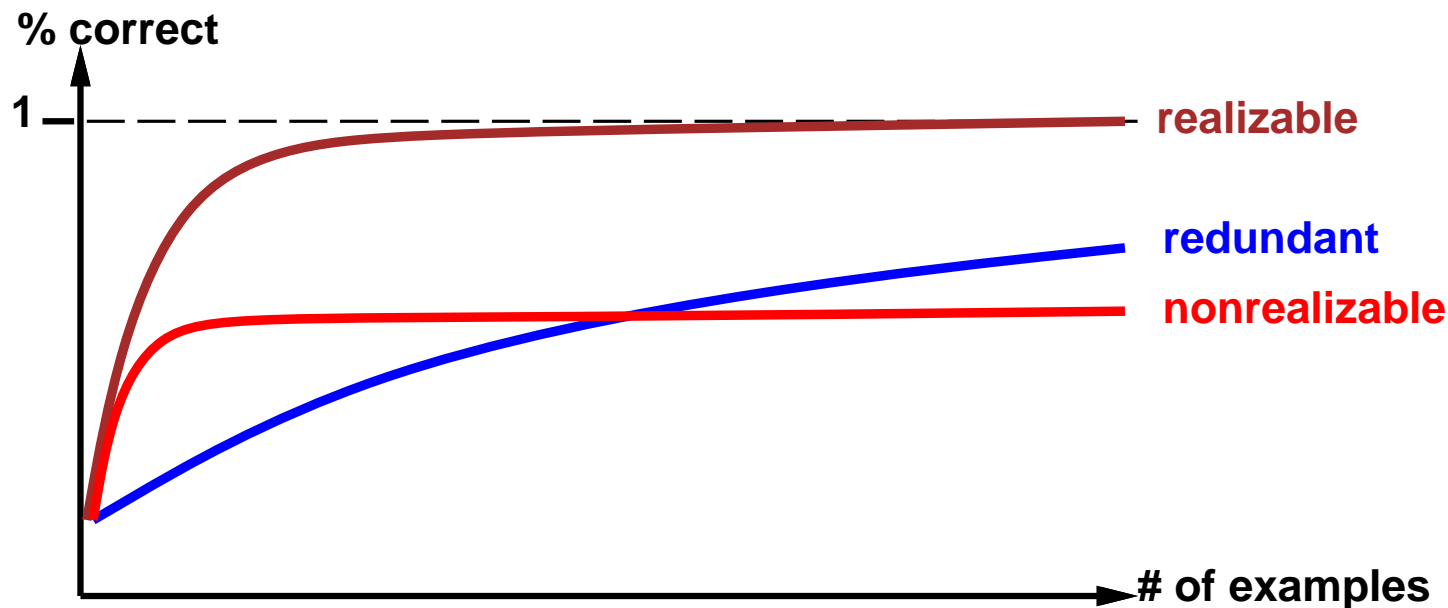
Krivulja učenja = % korektnosti na testnom skupu kao funkciji ovisnoj o veličini skupa za treniranje



Mjerenje uspjeha nastavak

Krivulja učenja ovisi o

- **ostvarljivost** (može izraziti funkciju cilja) vs. **neostvarljivost**
neostvarljivost može biti nedostatak atributa, ili posljedica restrikcije klasa hipoteza (na primjer, linearna funkcija s pragom)
- redundantna izražajnost (na primjer, hrpe nebitnih atributa)



Sažetak

Učenje je potrebno u nepoznatim sredinama, lijeni dizajneri

Agent za učenje = element izvedbe + element učenja

Metoda učenja ovisi o

- tipu elementa izvedbe, dostupnoj povratnoj informaciji,
- tipu komponente koja se može poboljšati, i njezinoj reprezentaciji

Za učenje pod nadzorom, cilj je naći jednostavnu hipotezu koja je približno konzistentna sa skupom primjera za treniranje

Učenje korištenjem stabla odlučivanja koristi dobivene informacije

Mjerenje uspjeha učenja = predviđanje točnosti mjereno na testnom skupu