

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Агенти способни да учат (машинско учење)

Машинско учење

- Целта на машинското учење е вградувањето на способностите за учење кај сметачите.
- Предмет на машинското учење се студиите и компјутерското моделирање на процесите на учење и нивните манифестации.
- Машинското учење се занимава со создавањето и развојот на алгоритмите и техниките кои му овозможуваат на сметачот да учи.

Потреба од машинското учење

- Развивање системи кои во случај рачно да се развиваат би биле тешки или скапи, а притоа бараат специфични знаења или вештини.
- Развивање системи кои автоматски се приспособуваат и прилагодуваат (adapt and customize), како на пример:
 - Персонализирани филтри за вести или пораки
 - Персонализирано подучување
- Откривање нови знаења од големи податочни бази, како на пример:
 - Анализа на пазарот
 - Текстуално рударење на медицински податоци

Што е учење?

- Промени во даден систем кои водат кон тоа при следната појава на слична задача, системот да ја извршува поефикасно и поефективно
- Herbert Simon: “Учење е било кој процес преку кој еден систем го подобрува своето однесување врз основа на стекнатото искуство.”
- Од тривијално меморирање на факти ($2+2=4$) до развивање на потполно нови теории (откривање нови знаења)

Учењето опфаќа

- Усвојување ново декларативно знаење
- Развивање моторни и спознајни вештини по пат на инструкции или пракса
- Организација на новото знаење во општи, ефективни репрезентации
- Откривање нови факти и теории по пат на набљудувања или експерименти.

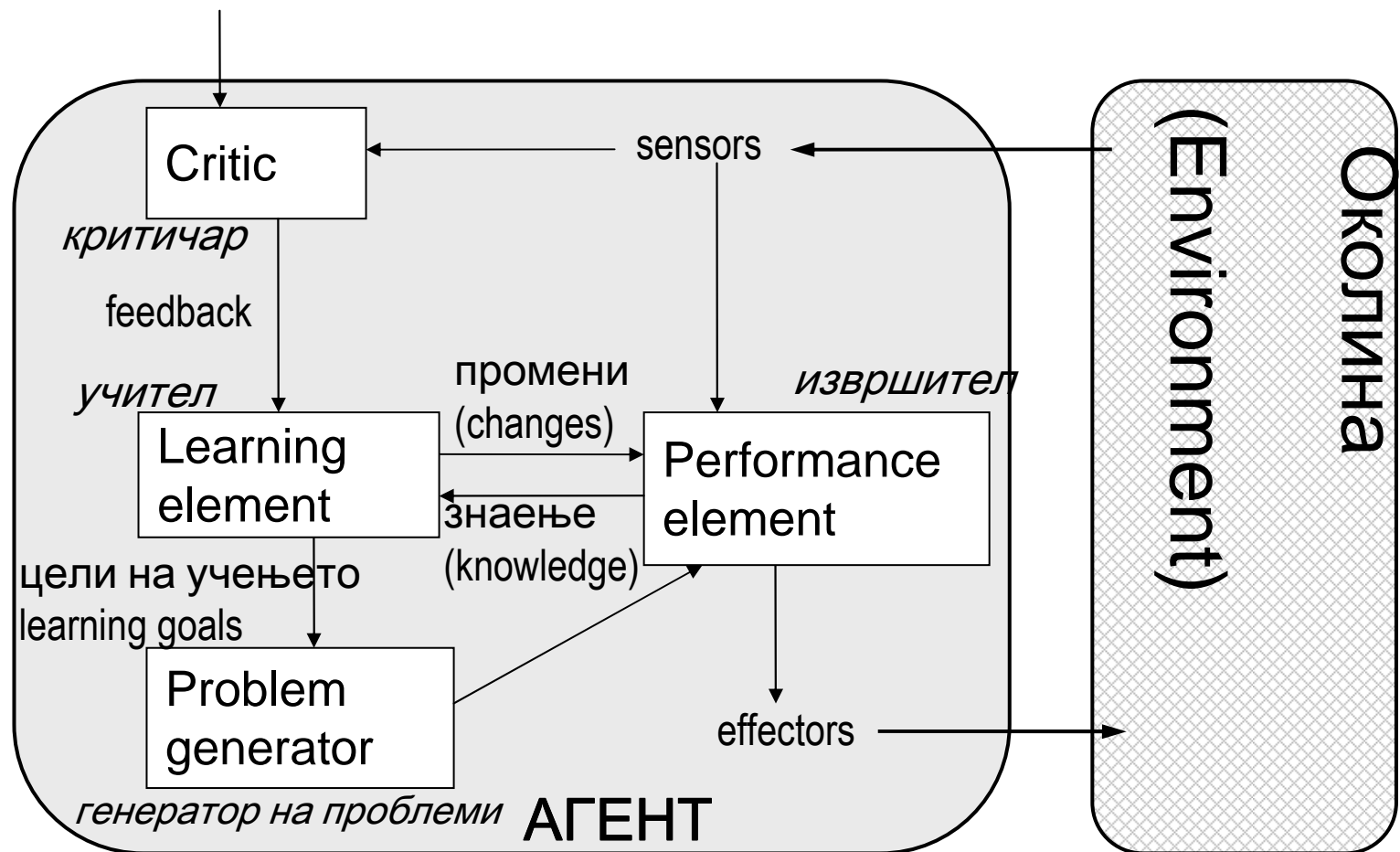


Како учи еден агент?

- Со механичко учење или бубање (rote learning)
- Учење според инструкции (learning from instruction)
- Учење по аналогичја (learning by analogy)
- Учење од набљудување и откривање (learning from observation and discovery)
- Учење од примери (learning from examples)

Општ модел на агент кој учи

надворешен стандард за изведбата
(external performance standard)



Учител (Learning element)

- изајнот на учителот зависи од:
 - *компонентите* на извршителот кои треба да се научат;
 - *повратната информација* која е на располагање за да се научат овие компоненти;
 - која е *репрезентацијата* на овие компоненти;

Видови учење врз основа на повратната информација

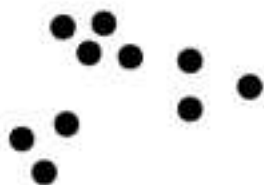
- Учење со учител – надгледувано (supervised learning): со секој примерок се проследува и точниот одговор;
- Учење без учител – самоучење, ненадгледувано (unsupervised learning): не се проследува точниот одговор;
- Учење со поттикнување (reinforcement learning) (наметнато учење): повремени награди и/или казни.

Надгледувано учење

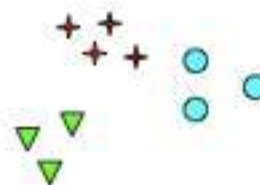
- Надгледуваното учење е техника со која се создаваат функции само врз основа на множество за обука (training set).
- Целта на ова учење е да ја предвиди вредноста на функцијата за секој влезен објект откако ќе научи како да ја предвиди вредноста на функцијата по прикажувањето примери за обука (training examples)
- Значи, станува збор за примена на обопштувањето.
- Пример: ако на системот му се претстават илјадници именки во еднина и множина, тој се обидува да најде правила за формирање на множината од еднината (споредете со слајд 55).

Ненадгледувано учење

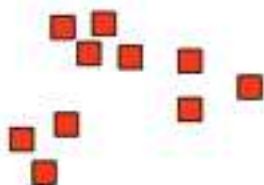
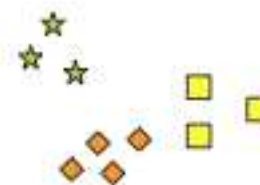
- Ненадгледуваното учење се занимава со класификација или дефинирање кластери во кои припааат влезовите.
- Тоа проучува како да се претстават влезовите за да ја одразат статистичката структура на сите можни збирки од влезни обележја.
- Пронајдените објекти кои припааат на даден кластер се слични меѓу себе (или во некаква релација), но различни од објектите кои припааат на други групи.
- Примена:
 - а се воочи некоја зависност / поврзаност (пр. групирање на слични документи на Интернет)
 - Сумаризација т.е да се намали големината на множеството податоци



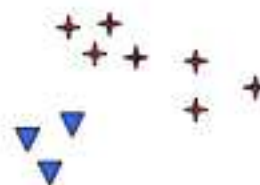
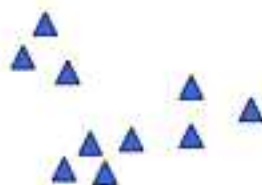
Initial points.



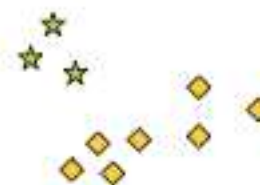
Six Clusters



Two Clusters



Four Clusters



ројот и големината на кластерите може да се одреди според неколку параметри

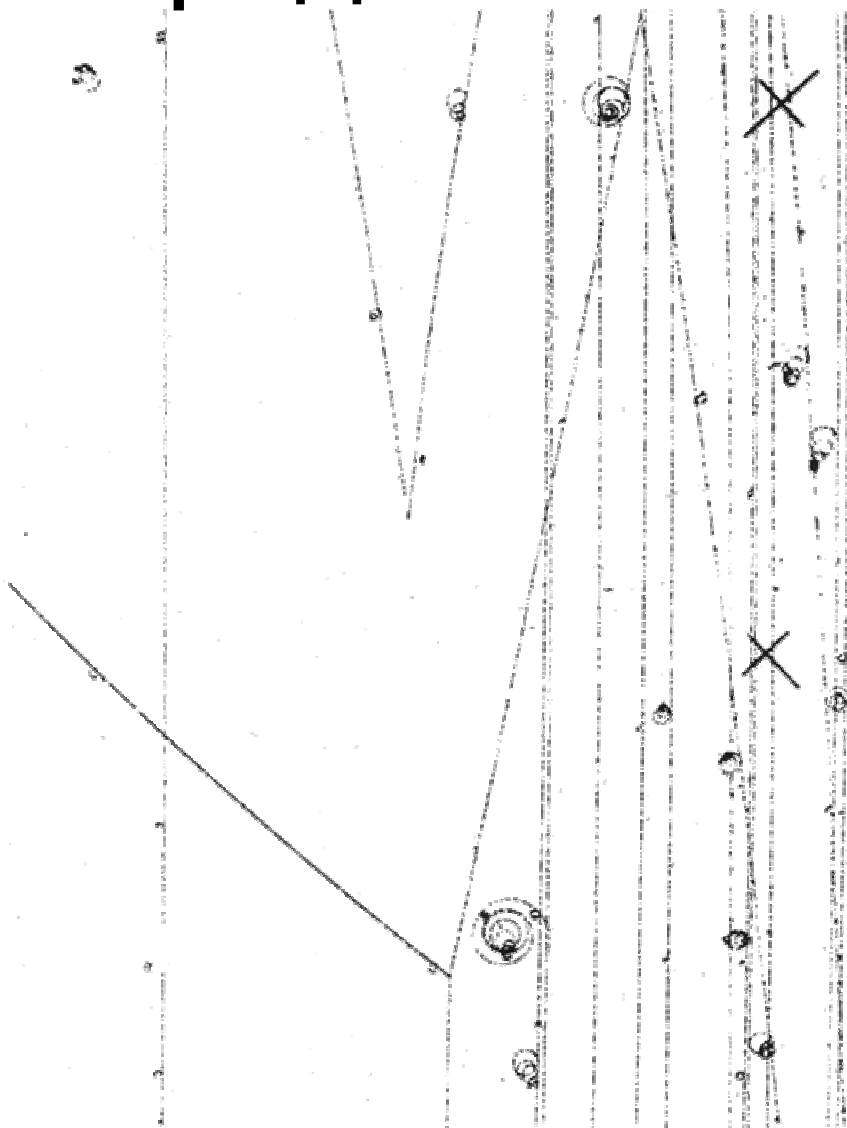
Учење со поттикнување

- Учењето со поттикнување се занимава со тоа кои активности треба да ги преземе еден автономен агент во околината во која делува за тоа да биде оптимално.
- Се базира на доделување награди за оние активности што доведуваат до подобро однесување, и се стреми кон тоа да ги максимизира наградите.
- Исто така може да се доделуваат и казни за оние активности кои водат до полошо однесување и се стреми кон тоа да ги минимизира казните.
- Отсуство на награда исто така може да се смета како еден вид на казна.
- Алгоритмите што тука се применуваат прават обид да најдат т.н. политики (полиси) кои оптимално ги пресликуваат состојбите на светот во активности на агентот.

Учење со поттикнување

- Кај игрите како што е шахот, поттикот се добива дури на крајот на играта. Кај други околина, поттиците се почести.
 - Во пинг-понг, секој постигнат поен може да се смета за награда. Кога учиме како да ползиме, секое движење нанадпред е постигнување.
- Недостатокот на бакшиш на крајот на патувањето или големата сметка за поправка заради можен судир со колата пред него, му дава на агентот (таксистот) некаков показател дека таквото однесување е непосакувано.

Предзнаење



- Мноштвото на ML истражувања во AI се концентрираат на обука на агент кој на почетокот нема никакво знаење за тоа што се обидува да научи. Тој има пристап само до примероци презентирани од неговото искуство.
- Во реалноста најголем дел од учењето кај човекот се изведува во контекст на одредено предзнаење.
- Bubble-chamber photo. Како ја гледа
 - уметник (слика од апстрактен експресионист)
 - нуклеарен физичар (постоење на нова честица со одредена маса и полнеж)

Некои аспекти поврзани со процесот на учење

Парови спротивставени поими при учењето:

- индукција наспроти дедукција
- генерализација наспроти спецификација
- апстракција наспроти конкретизација
- конзистентност наспроти комплетност

Индуктивно учење (1)

Кај надгледуваното учење на учителот му се проследуваат (приближно) точни вредности на функцијата за одредени влезови, и учителот се обидува да ја измени репрезентацијата на функцијата со цел нејзината вредност да се совпадне со доставената повратна информација.

f е целната функција

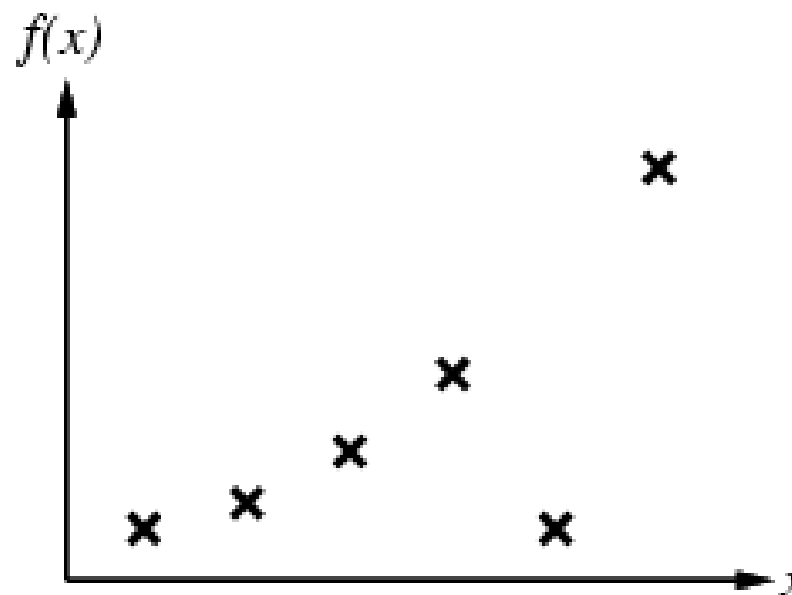
примерок е парот $(x, f(x))$

каде x е влезот, а $f(x)$ е вредноста на функцијата f за влезот x

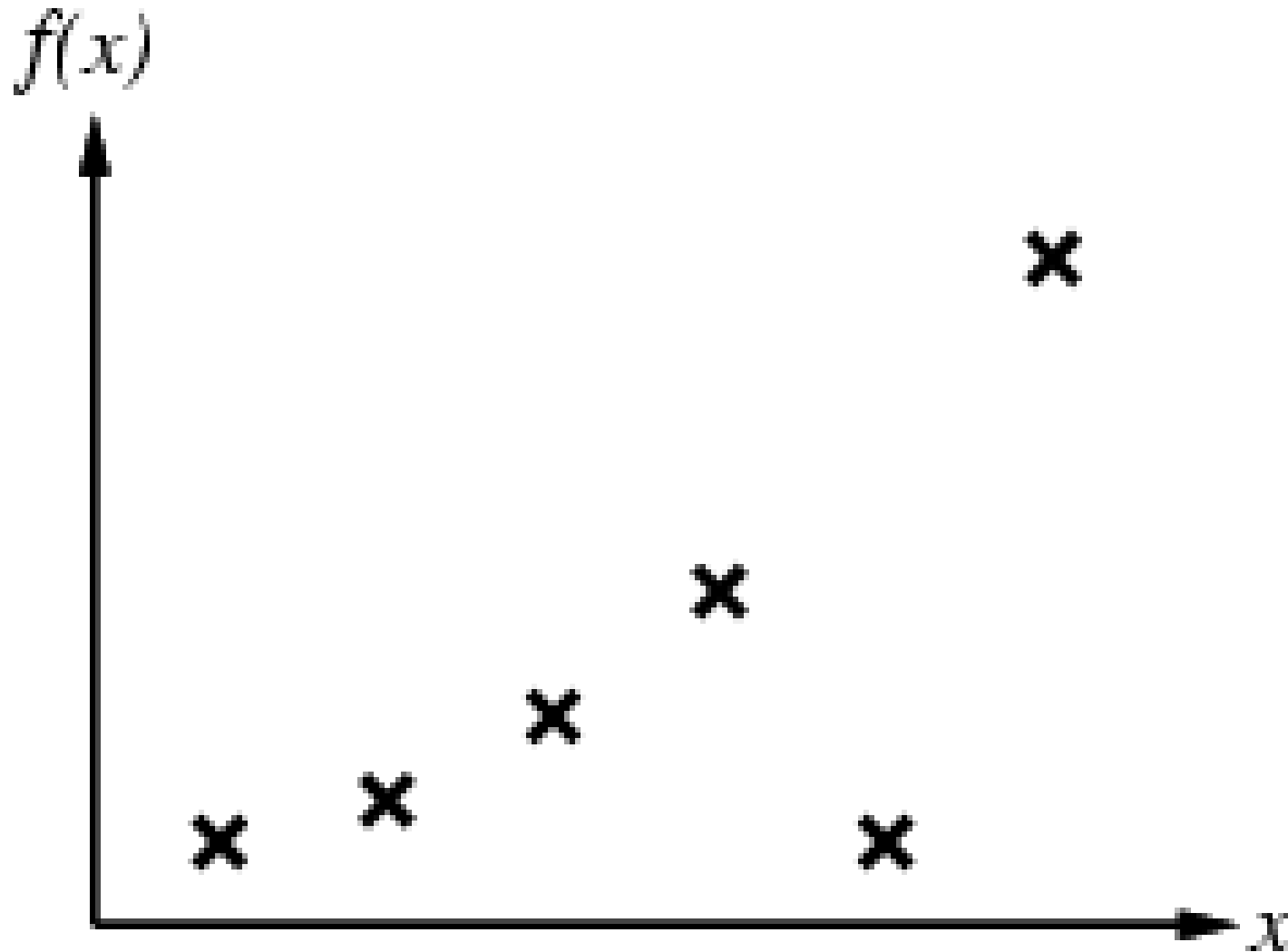
Проблем: да се пронајде хипотеза h така да $h \approx f$, за дадено множество примероци за обука

Индуктивно учење (2)

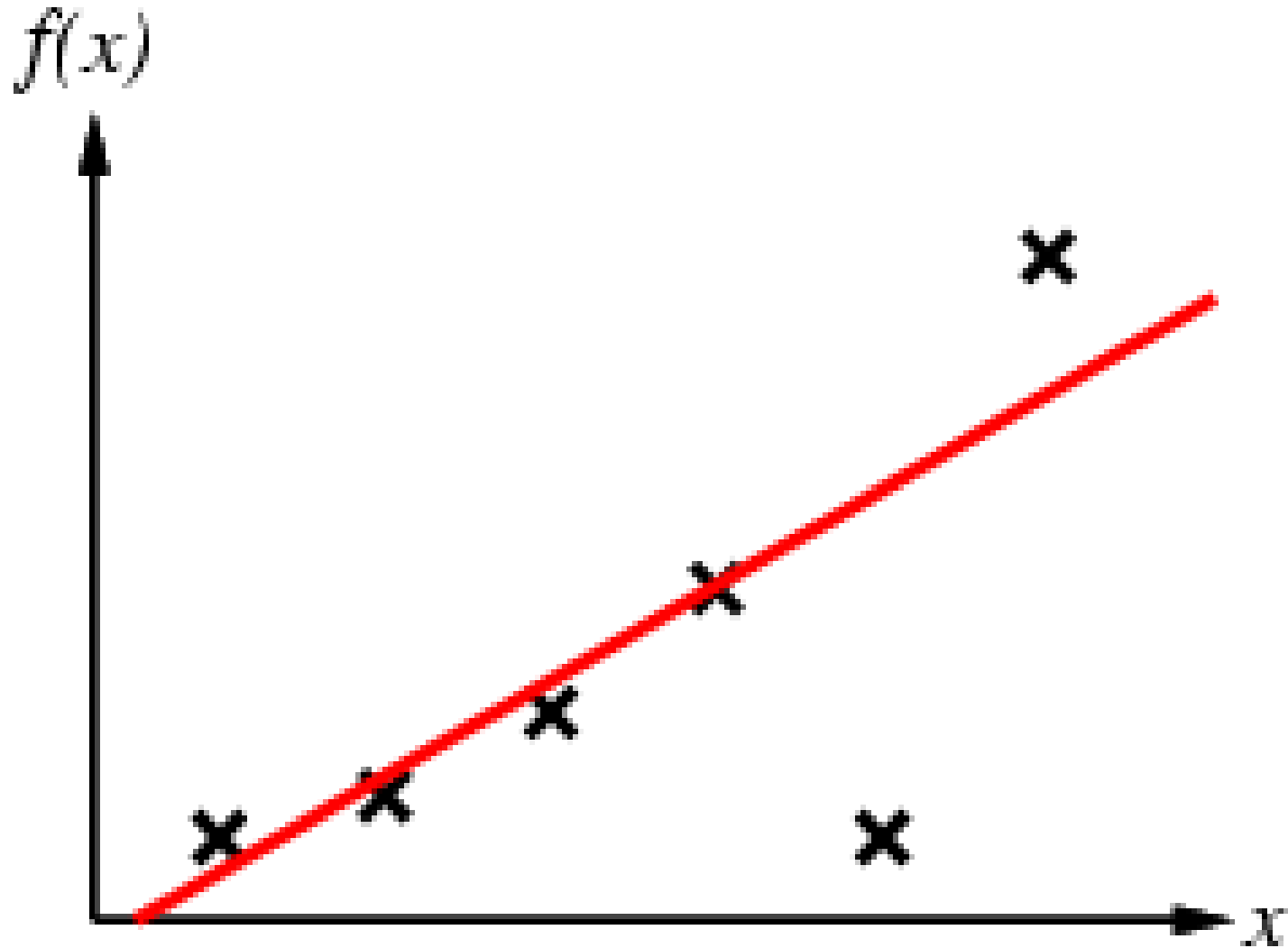
- Пронајди / нагоди функција h која ќе се совпаа со f на примероците од множеството за обука.
- h е конзистентна доколку се совпаа со f за сите примероци
- Пр. curve fitting.



Индуктивно учење (3)

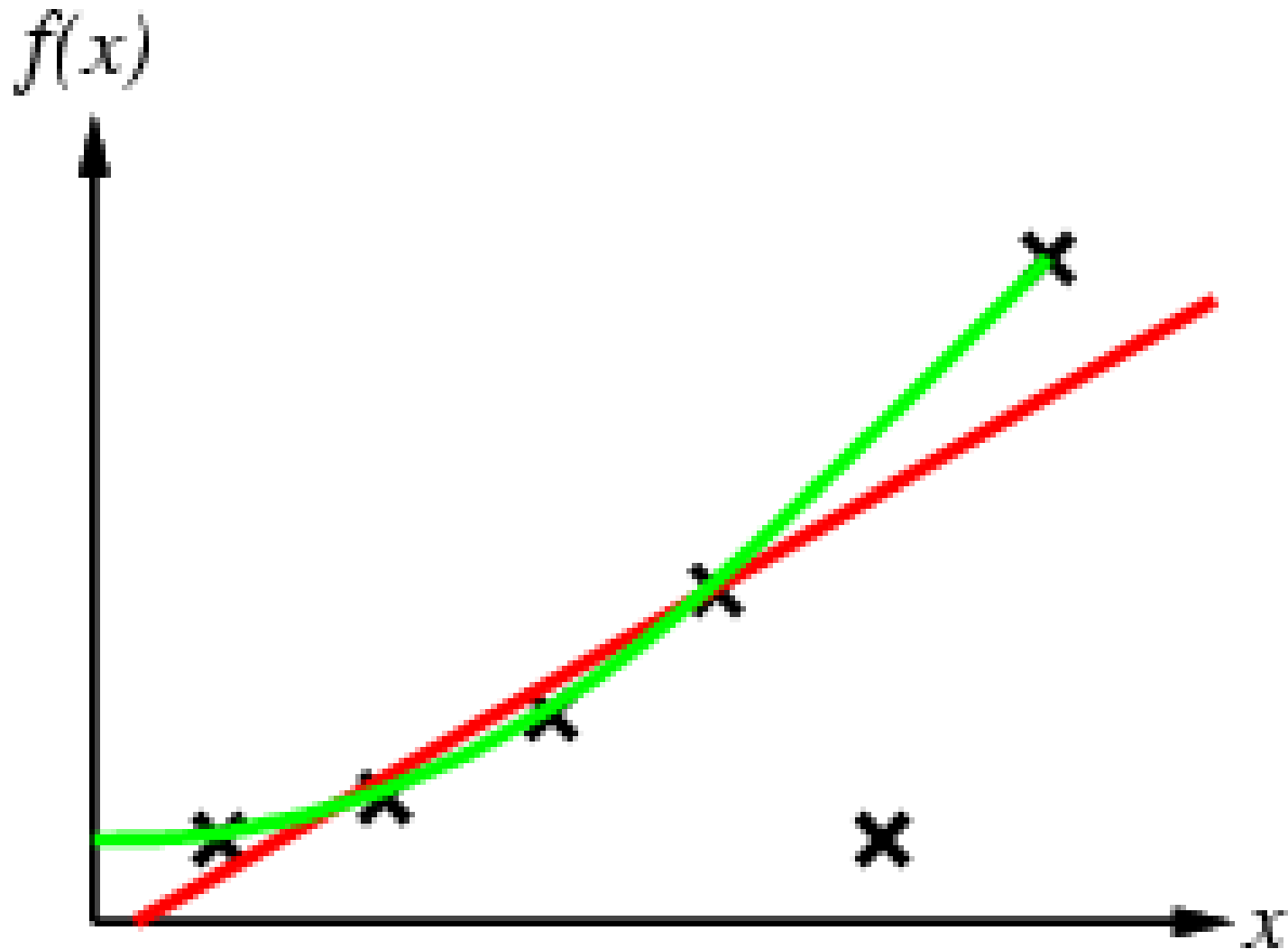


Индуктивно учење (4)

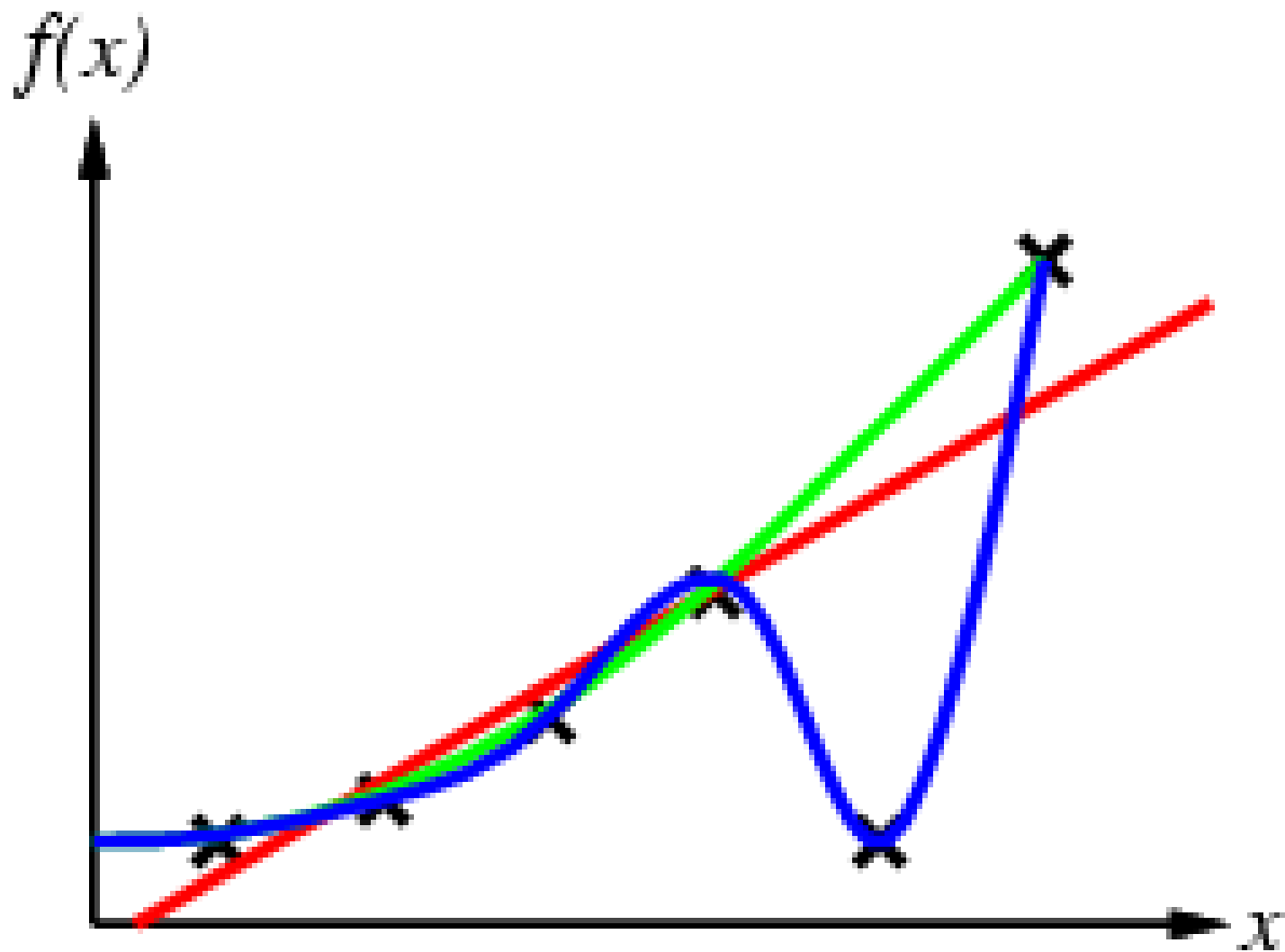




Индуктивно учење (5)

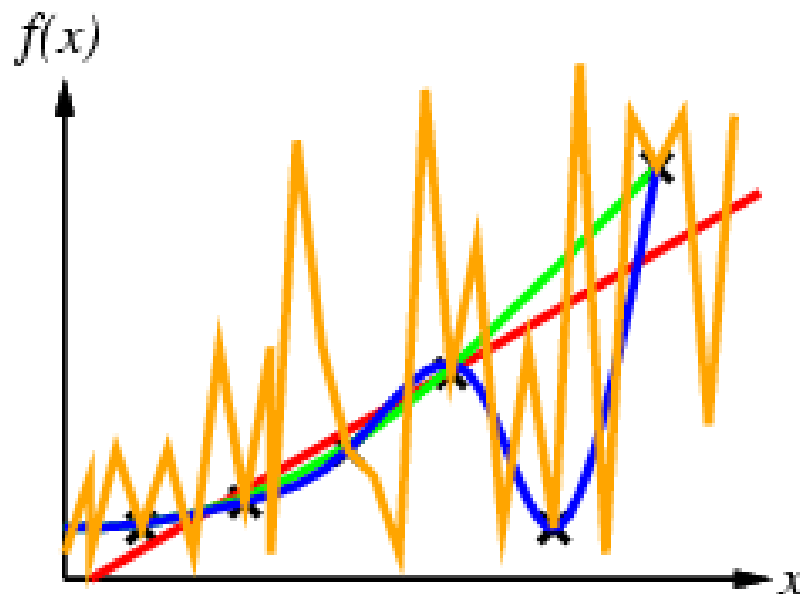


Индуктивно учење (6)



Индуктивно учење (7)

- bias – наклон. Сите алгоритми за учење изразуваат одреден вид на наклонетост кон една или друга хипотеза од множеството можни хипотези.



ричот на Окам (Ockham's razor): Преферирање на наједноставната хипотеза конзистентна со податоците.

Инкрементално учење

- Наместо примена на алгоритмот за учење над целото множество на примероци за обука одеднаш, учење еден по еден примерок.
- Со презентацијата на секој нов примерок, агентот се обидува да ги измени постоечките хипотези (да бидат конзистентни и со новиот примерок и со сите поранешни). Агентот може да добива и одредени информации во врска со квалитетот на акциите кои ги презема.
- Учењето стихотворби е комбинација на инкрементално и целосно учење
- рва за одлучување (Decision Trees)
- Невронски мрежи (Neural Networks)
- SVM (Support Vector Machines) (машини со носечки / потпорни / поткрепувачки / поддржувачки вектори)

AQ алгоритам (1)

- предложен од Michalski (1969).
- Влез: табела од примери кај кои се зададени вредностите на атрибутите.
- Излез: опис на концептот обично зададен со логички израз.

| Табела Т. 6.1 Почетна табела за алгоритмот AQ | | | | |
|---|--------|--------|--------|------------|
| пример | атриб1 | атриб2 | атриб3 | припадност |
| E1 | x | r | m | + K |
| E2 | y | r | n | + K |
| E3 | y | s | n | + K |
| E4 | x | s | m | - K |
| E5 | z | t | n | - K |
| E6 | z | r | n | - K |

AQ алгоритам (2)

1. аденото множество примери подели го на множество PE позитивни и NE негативни примери.
2. Одбери случајно пример од PE. Нека е тоа пример почетник, наречен семе.
3. Најди најгенерални описи (кои покриваат најмногу примери) на семето, при што границата е дефинирана со NE. Множеството на тие описи се нарекува везда.
4. Одбери некој опис од множеството описи, според некој критериум.
5. Ако така добиениот опис заедно со претходно одбраните описи ги покрива сите позитивни примери, запри; инаку најди друго семе и оди на 3.

AQ алгоритам (3)

Табела Т. 6.1 Почетна табела за алгоритмот AQ

| пример | атриб1 | атриб2 | атриб3 | припадност |
|--------|--------|--------|--------|------------|
| E1 | x | r | m | + K |
| E2 | y | r | n | + K |
| E3 | y | s | n | + K |
| E4 | x | s | m | - K |
| E5 | z | t | n | - K |
| E6 | z | r | n | - K |

| итерација | чекор | акција |
|-----------|-------|--|
| 1 | 2 | нека семе биде E2 |
| 1 | 3 | најгенерален опис на E2 е (atrib1=y) |
| 1 | 4 | нема потреба |
| 1 | 5 | атриб1 не го покрива примерот E1. Нека е ново семе E1 |
| 2 | 3 | најгенерален опис на E1 даваат описите (atrib1=x)&(atrib2=r) и (atrib2=r)&(atrib3=m) |
| 2 | 4 | преферираме (atrib1=x)&(atrib2=r) |
| 2 | 5 | добиваме опис (atrib1=y) V [(atrib1=x)&(atrib2=r)] Тој опис ги покрива сите позитивни примери Стоп. |

Според примерот, концептот K ќе биде репрезентиран со изразот $K = (atrib1=y) \vee [(atrib1=x) \wedge (atrib2=r)]$

рва за одлучување

- Влез – објект или ситуација опишана со множество особини (обележја)
- Излез – А/НЕ одлука
- Секој внатрешен јазел претставува испитување на вредноста на една од особините (и разгранување во зависност од вредноста)
- Секој лист претставува булова вредност која треба да се врати како резултат ако се стигне до него

Пример Russell & Norvig (1)

Problem: decide whether to wait for a table at a restaurant, based on the following attributes:

1. Alternate: is there an alternative restaurant nearby?
2. Bar: is there a comfortable bar area to wait in?
3. Fri/Sat: is today Friday or Saturday?
4. Hungry: are we hungry?
5. Patrons: number of people in the restaurant (None, Some, Full)
6. Price: price range (\$, \$\$, \$\$\$)
7. Raining: is it raining outside?
8. Reservation: have we made a reservation?
9. Type: kind of restaurant (French, Italian, Thai, Burger)
10. WaitEstimate: estimated waiting time (0-10, 10-30, 30-60, >60)

Пример Russell & Norvig (2)

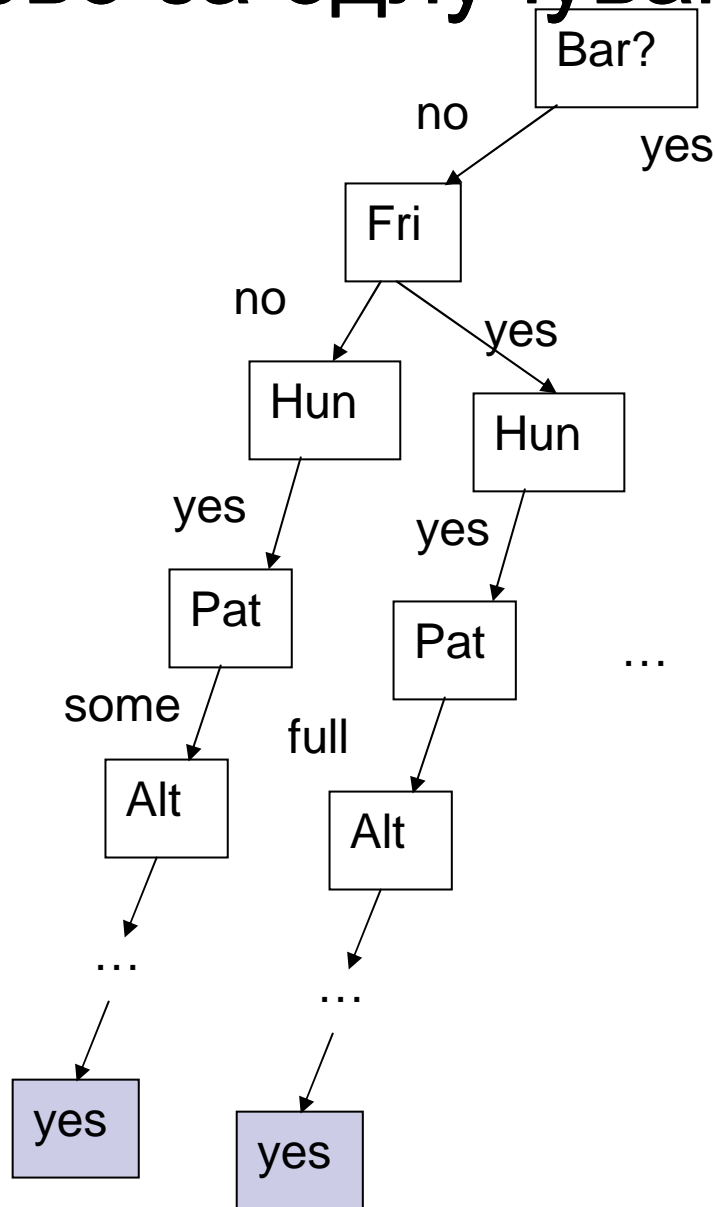
- Проблем: одлука дали да се чека за маса во некој ресторан, врз основа на следните атрибути:
 1. Алтернатива: дали има алтернативен ресторан во близина?
 2. бар: дали има удобен бар за чекање?
 3. Пет/Саб: дали денот е петок или сабота?
 4. Гладни: дали сме гладни?
 5. Стални гости: бројот на познати стални гости во ресторанот (ниеден, неколку, сите)
 6. Цена: опсег на цени (\$, \$\$, \$\$\$)
 7. Врне: дали надвор врне?
 8. Резервација: дали сме направиле резервација?
 9. Тип: тип на ресторанот (француски, италијански, тајландски, сендвичара)
 10. ПроценкаЧекање: проценето време на чекање (0-10, 10-30, 30-60, >60)

Пример Russell & Norvig (3)

- Претстава на примероците преку вредности на атрибутите кои може да се дискретни, континуирани или улови
- Класификација на примероците на позитивни и негативни

| 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>Wait</i> |
| X_1 | T | F | F | T | Some | \$\$\$ | F | T | French | 0–10 | T |
| X_2 | T | F | F | T | Full | \$ | F | F | Thai | 30–60 | F |
| X_3 | F | T | F | F | Some | \$ | F | F | Burger | 0–10 | T |
| X_4 | T | F | T | T | Full | \$ | F | F | Thai | 10–30 | T |
| X_5 | T | F | T | F | Full | \$\$\$ | F | T | French | >60 | F |
| X_6 | F | T | F | T | Some | \$\$ | T | T | Italian | 0–10 | T |
| X_7 | F | T | F | F | None | \$ | T | F | Burger | 0–10 | F |
| X_8 | F | F | F | T | Some | \$\$ | T | T | Thai | 0–10 | T |
| X_9 | F | T | T | F | Full | \$ | T | F | Burger | >60 | F |
| X_{10} | T | T | T | T | Full | \$\$\$ | F | T | Italian | 10–30 | F |
| X_{11} | F | F | F | F | None | \$ | F | F | Thai | 0–10 | F |
| X_{12} | T | T | T | T | Full | \$ | F | F | Burger | 30–60 | T |

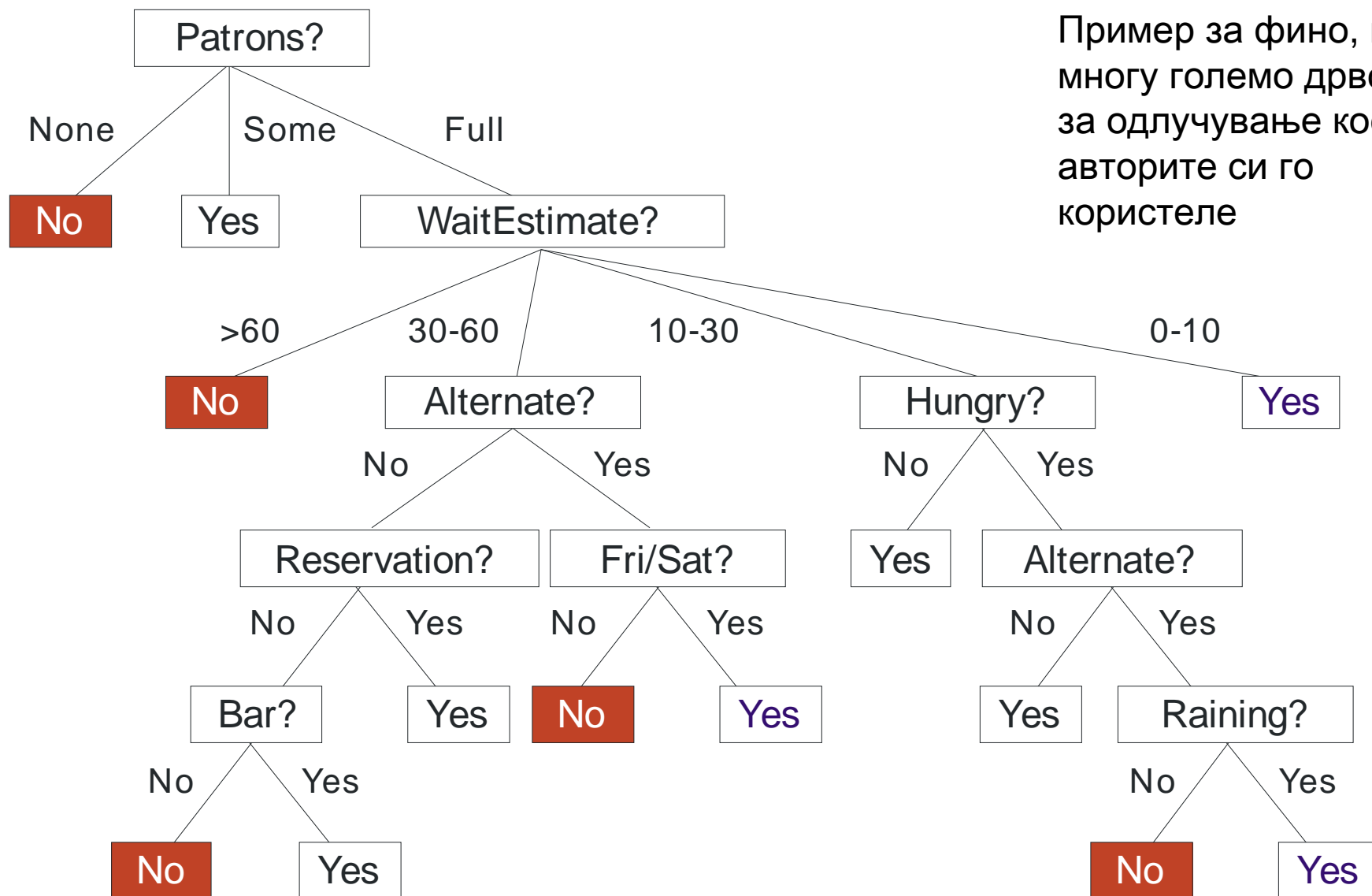
Пример на некое случајно избрано дрво за одлучување



Проблем: дрвото може да е многу големо, т.е. неоптимално

Колку различни дрва на одлука може да се креираат со n улови атрибути?
 $n \Rightarrow$ број на атрибутите
 $2^n \Rightarrow$ комбинации влезови
 2^n на степен $2^n \Rightarrow$ број на различни функции
(пр. со 6 улови атрибути постојат
18,446,744,073,709,551,616 дрва $\approx 2 \times 10^{19}$)

Пример Russell & Norvig (4)



Алгоритам за учење на дрвото за одлучување *

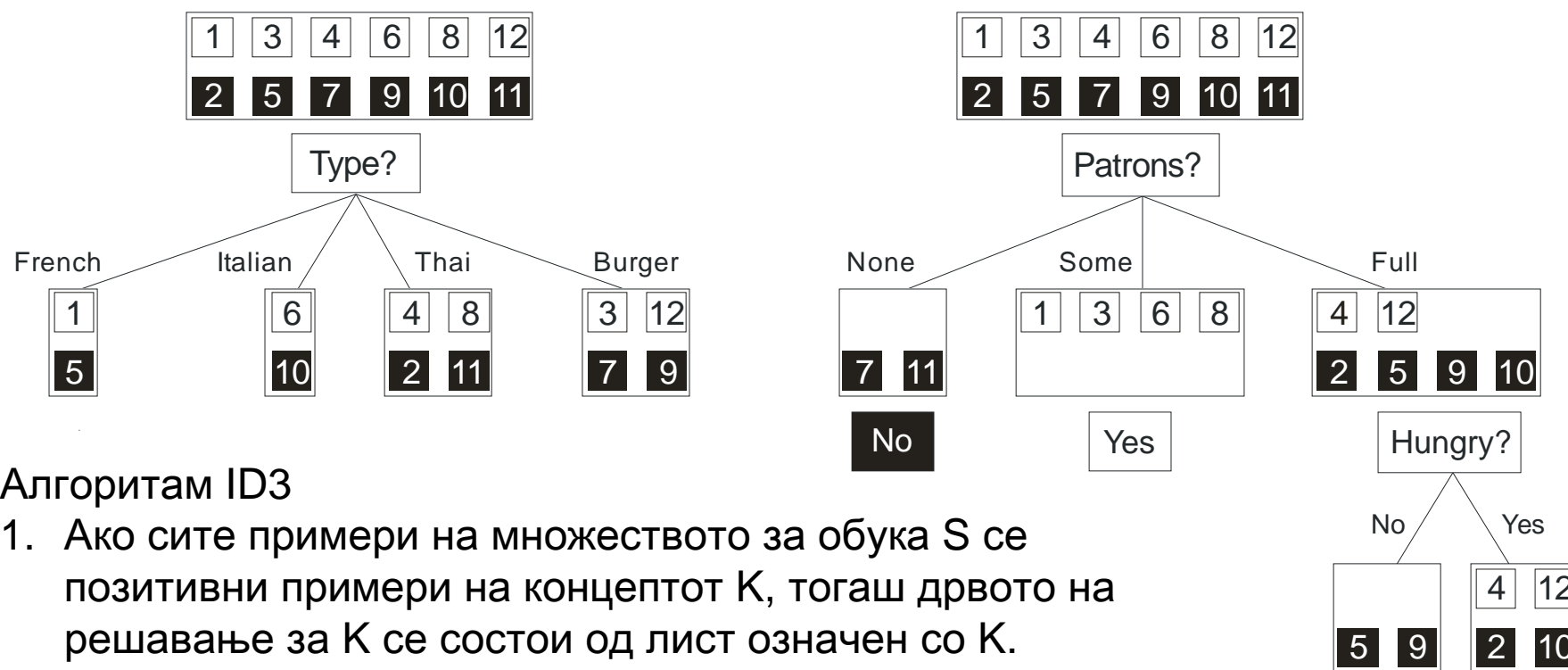
- Цел: да се пронајде мало дрво конзистентно со примероците за обука
- Идеја: рекурзивно избирање на „најзначајниот“ атрибут како корен на (под)дрво

```
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  $\leftarrow$  CHOOSE-ATTRIBUTE(attributes, examples)
    tree  $\leftarrow$  a new decision tree with root test best
    for each value  $v_i$  of best do
      examplesi  $\leftarrow$  {elements of examples with best =  $v_i$ }
      subtree  $\leftarrow$  DTL(examplesi, attributes — best, MODE(examples))
      add a branch to tree with label  $v_i$  and subtree subtree
    return tree
```

Учење на дрвото за одлучување

- Се бира “најдобриот” атрибут, т.е. оној што најмногу придонесува за лесната класификација на примероците.
- Ако се пола примероци негативни, а пола позитивни, тој атрибут не ни кажува многу, па не е добро него да го избереме на почетокот.
- Ако има само неколку позитивни или само неколку негативни примероци според некој атрибут, најдобро е таков некаков атрибут да го избереме погоре во дрвото и според него најпрво да се поделат останатите примероци.
- Ако остатокот се сите позитивни (или сите негативни) примероци тогаш сме дошле до крај (до лист) т.е. може да се одговори со А/НЕ.

ID3 алгоритам



Алгоритам ID3

1. Ако сите примери на множеството за обука S се позитивни примери на концептот K , тогаш дрвото на решавање за K се состои од лист означен со K .
2. Инаку, одбери тест кој ќе го подели множеството за обука S во две или повеќе подмножества S_i , зависно од излезот на тестот (на пр. вредностите на атрибутите). Тестот станува корен на дрвото и за секој излез на тестот изгради поддрво, повикувајќи ја оваа процедура рекурзивно за секое S_i .

Теорија на информации

- а се имплементира Choose-Attribute во DTL алгоритмот
- Количество информација (Entropy):

$$I(P(v_1), \dots, P(v_n)) = \sum_{i=1}^n -P(v_i) \log_2 P(v_i)$$

- За множество за обука со p позитивни и n негативни примери:

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

Информациска добивка

- Избраниот атрибут A го дели множеството за обука E на подмножества E_1, \dots, E_v според вредностите на атрибутот A , каде A прима v различни вредности.

$$remainder(A) = \sum_{i=1}^v \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

- Информациска добивка (Information Gain - IG) - намалувањето на ентропијата по изборот на атрибутот A :

$$IG(A) = I\left(\frac{p}{p + n}, \frac{n}{p + n}\right) - remainder(A)$$

- Избери го атрибутот со најголем IG

Информациска добивка*

За множеството за обука, $p = n = 6$, $I(6/12, 6/12) = 1$ bit

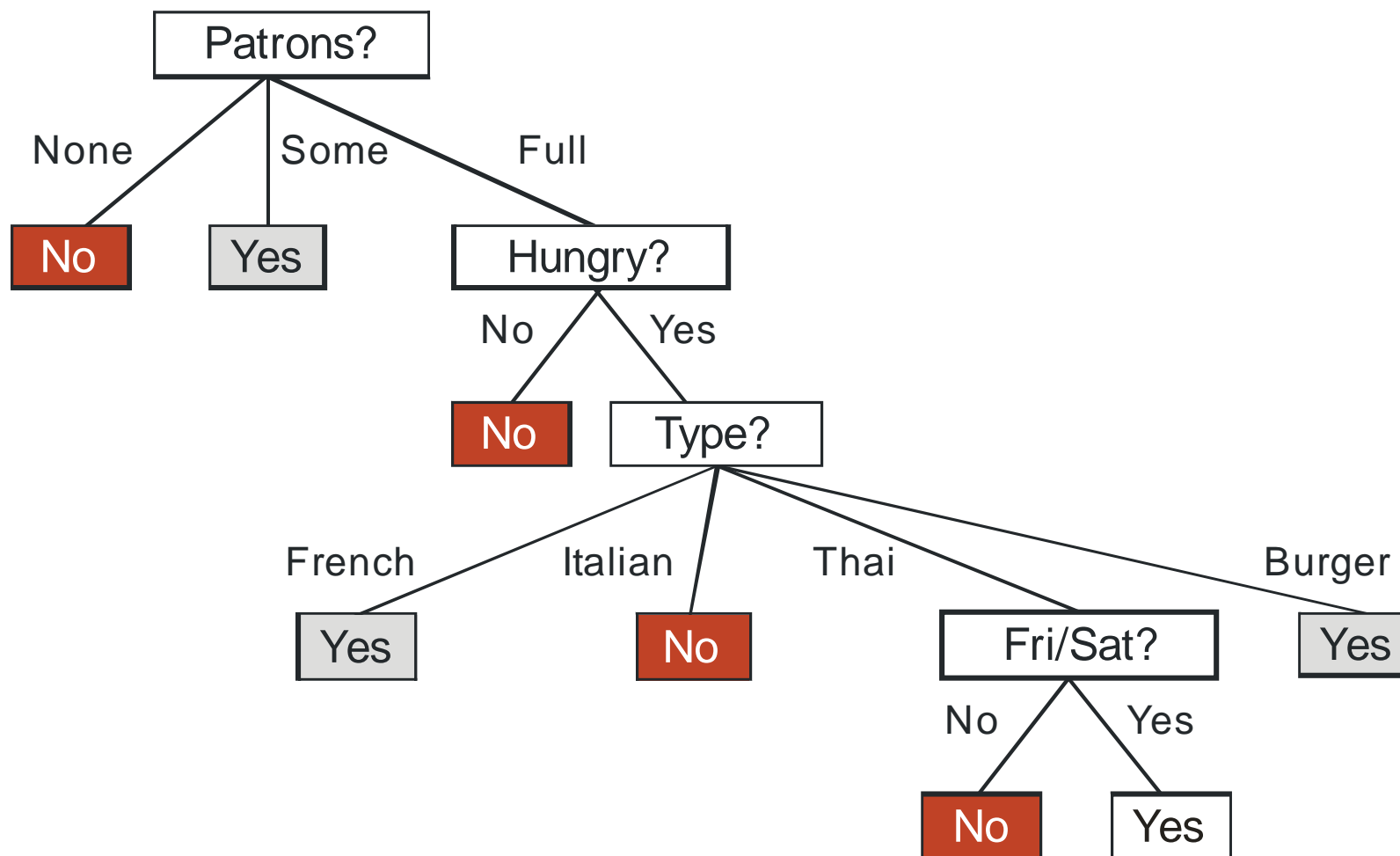
а ги разгледаме атрибутите *Patrons* и *Type* (и останатите):

$$IG(Patrons) = 1 - \left[\frac{2}{12} I(0,1)^{\text{none}} + \frac{4}{12} I(1,0)^{\text{some}} + \frac{6}{12} I\left(\frac{2}{6}, \frac{4}{6}\right) \right] = .0541 \text{ bits}$$

$$IG(Type) = 1 - \left[\frac{2}{12} I\left(\frac{1}{2}, \frac{1}{2}\right)^{\text{French}} + \frac{2}{12} I\left(\frac{1}{2}, \frac{1}{2}\right)^{\text{Italian}} + \frac{4}{12} I\left(\frac{2}{4}, \frac{2}{4}\right)^{\text{Thai}} + \frac{4}{12} I\left(\frac{2}{4}, \frac{2}{4}\right)^{\text{Burger}} \right] = 0 \text{ bits}$$

Patrons има највисок IG од сите атрибути и е избран од DTL алгоритмот како корен

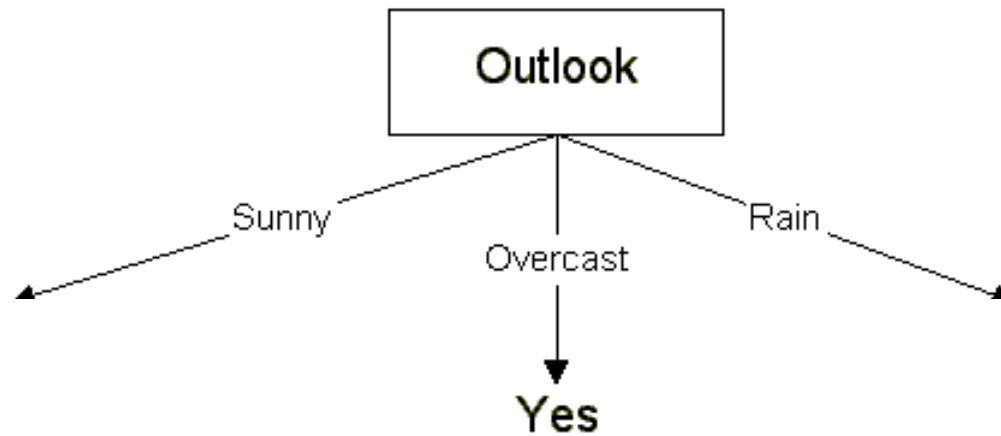
обиено дрво за одлучување



Пример 2: али да се игра со топка? (1)

| Day | Outlook | Temperature | Humidity | Wind | Play ball |
|-----|----------|-------------|----------|--------|-----------|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

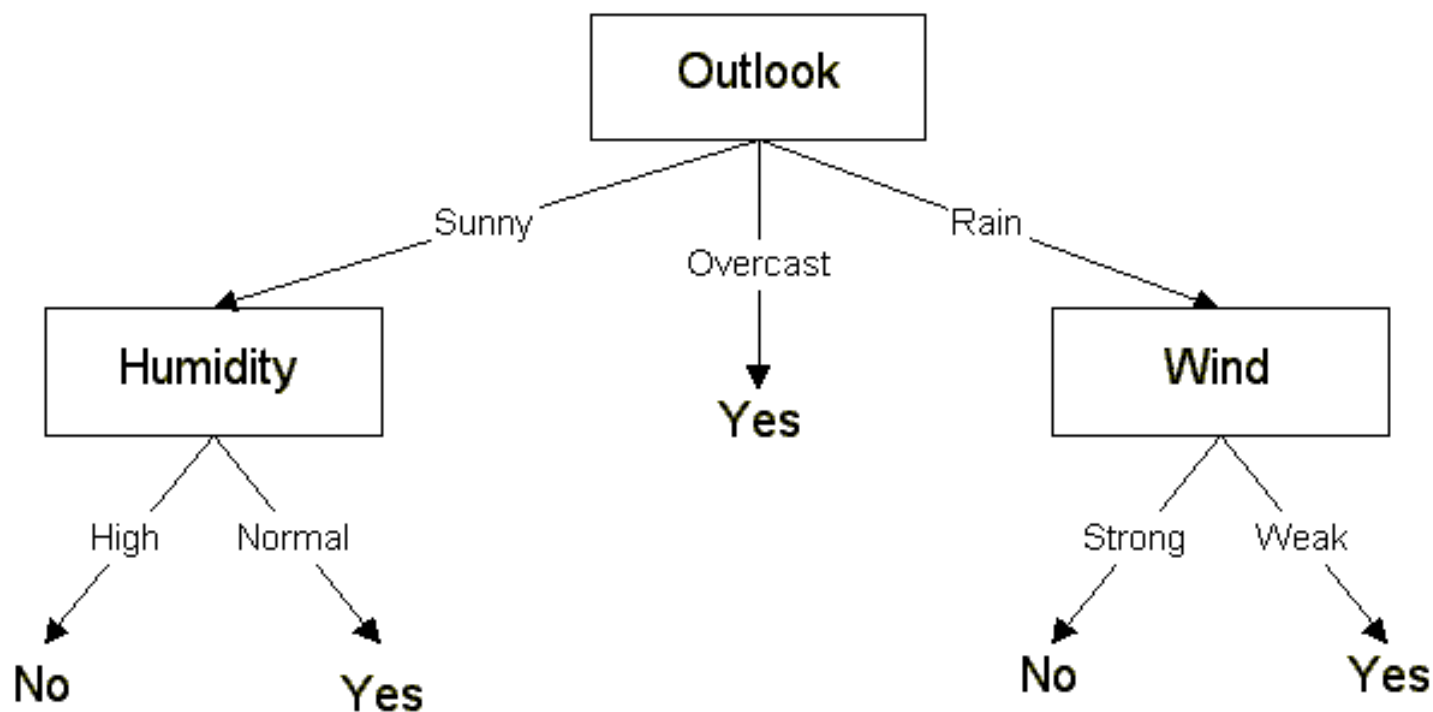
Научено дрво (2)



Пример 2: али да се игра со топка? (3)

| Outlook | Day | Temperature | Humidity | Wind | Play ball |
|----------|-----|-------------|----------|--------|-----------|
| Sunny | D1 | Hot | High | Weak | No |
| Sunny | D2 | Hot | High | Strong | No |
| Sunny | D8 | Mild | High | Weak | No |
| Sunny | D9 | Cool | Normal | Weak | Yes |
| Sunny | D11 | Mild | Normal | Strong | Yes |
| Rain | D4 | Mild | High | Weak | Yes |
| Rain | D5 | Cool | Normal | Weak | Yes |
| Rain | D6 | Cool | Normal | Strong | No |
| Rain | D10 | Mild | Normal | Weak | Yes |
| Rain | D14 | Mild | High | Strong | No |
| Overcast | D3 | Hot | High | Weak | Yes |
| Overcast | D7 | Cool | Normal | Strong | Yes |
| Overcast | D12 | Mild | High | Strong | Yes |
| Overcast | D13 | Hot | Normal | Weak | Yes |

Научено дрво (4)

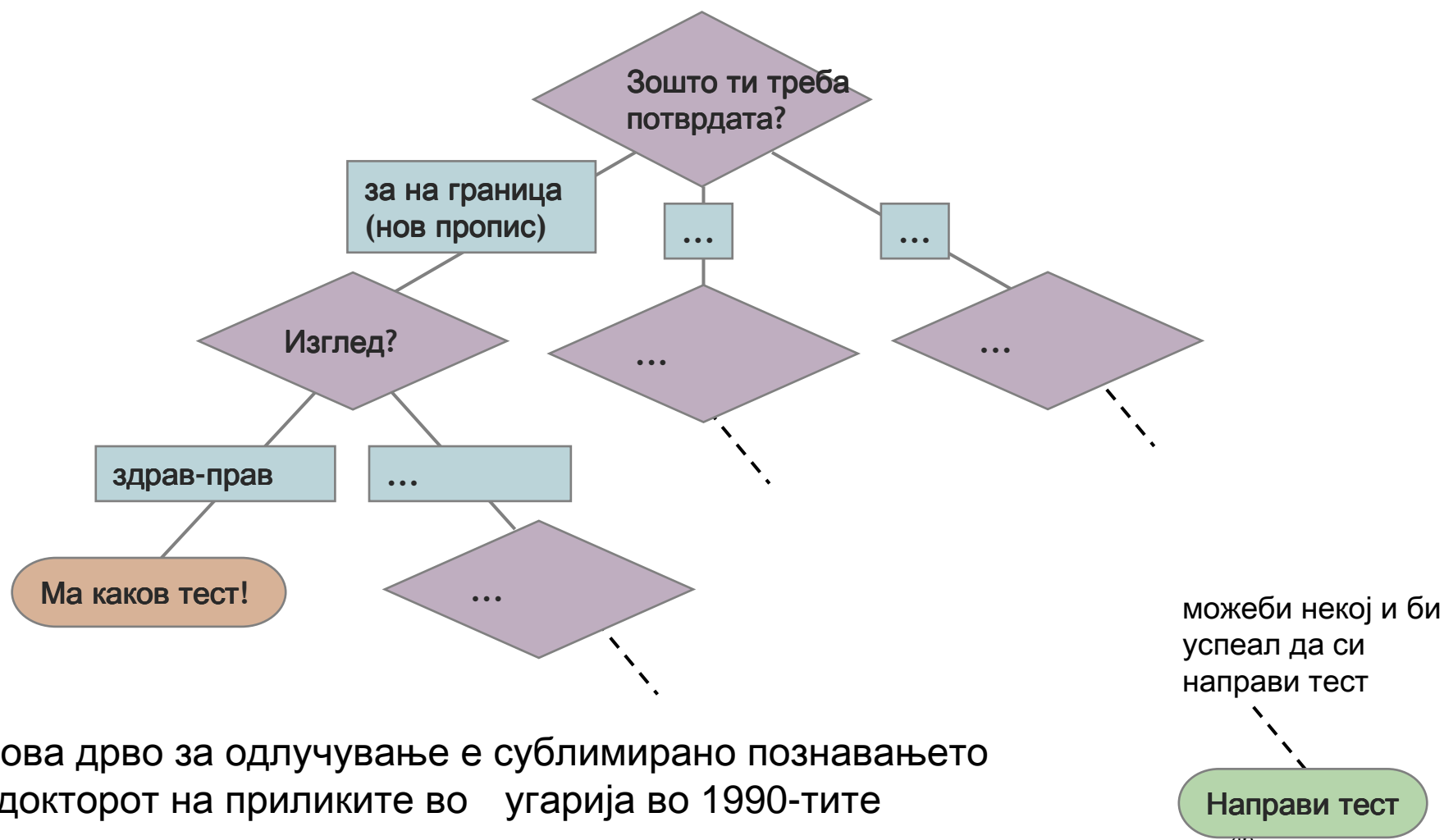




Зошто дрва за одлучување? (1)

- Целта е со што помалку прашања да се одлучи за нешто
- Кога треба да одлучиме за некоја комбинација која до сега не се појавила

Како во угарија прават тест за СИ А 😊



Во ова дрво за одлучување е сублимирано познавањето на докторот на приликите во угарија во 1990-тите

Зошто дрва за одлучување? (2)

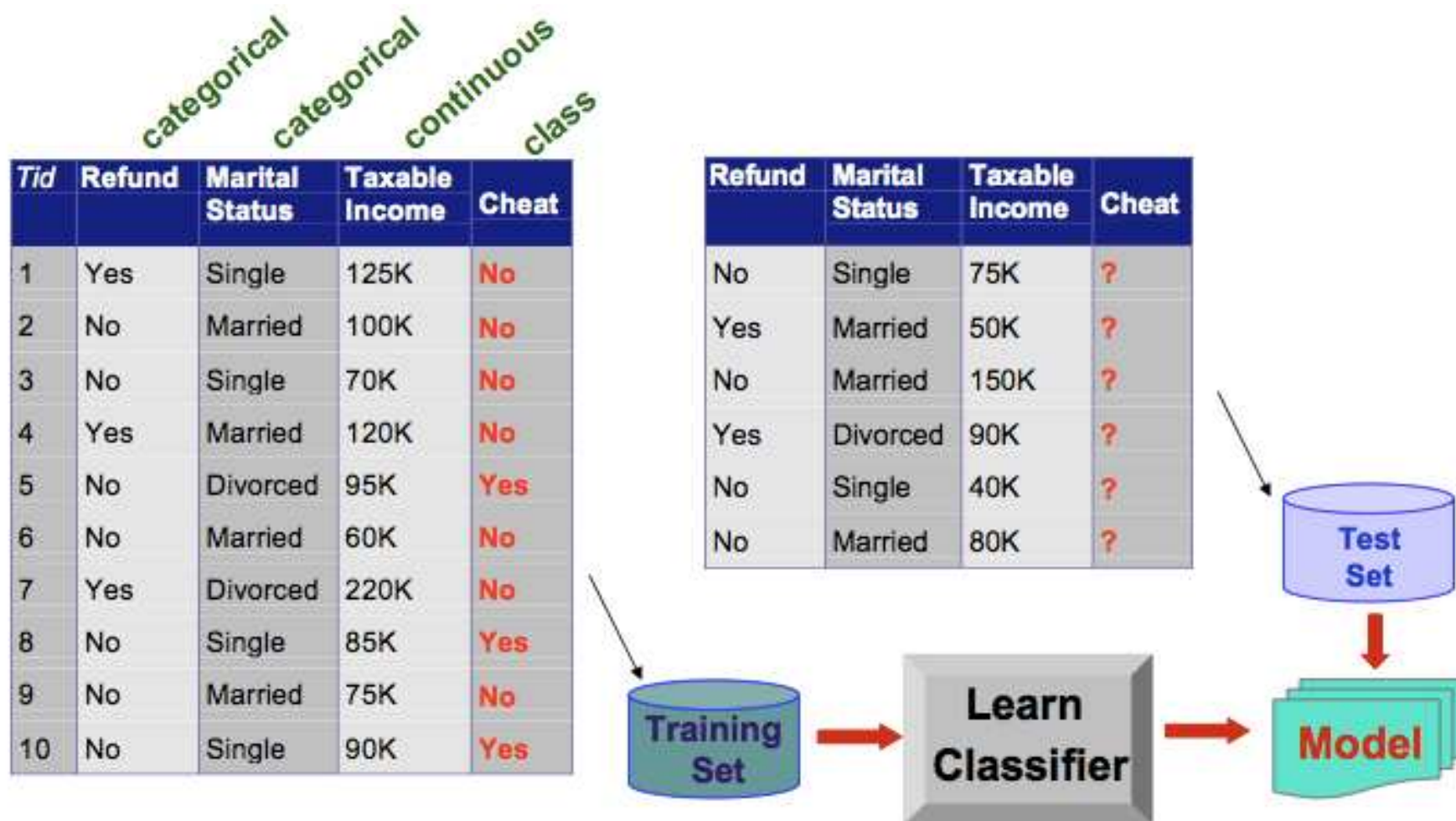
- Лесно интуитивно се толкуваат
- C4.5 (j48) се многу популарни
- Реален пример (од Russell & Norvig)
 - Системот на британскиот Петрол (BP) за одвојување на гас од нафта, што имал рачно направен потсистем со 2500 правила, бил заменет со дрва за одлучување. Системот направен со C4.5 давал подобри перформанси од човечките експерти и му заштедил на BP милиони. (1986)

Проблеми кај дрвата за одлучување

- Ако за некоја комбинација нема примерок во базата, т.е. значи нема опсервација за таков пример, тогаш се враќа некоја предодредена (default) вредност.
- Ако нема веќе атрибути, но имаме сеуште позитивни и негативни примероци, настанува проблем. Тоа значи дека примероците имаат ист опис, но различна класификација. Тоа се случува кога недостасуваат некои од податоците или се погрешни. Може да се случи кога проблемот е недетерминистички или кога не е опишан целосно со дадените примероци. Тогаш може да се користи мнозинско гласање - majority vote.

Трите основни множества кај учењето

- Учењето кое се базира на множества, ги користи овие три множества:
 - Множество за обука (training set)
 - Множество за исправност (validation set)
 - Множество за тестирање (testing set)
- Овие три множества не се неопходни во сите системи способни да учат
- Тие се карактеристични за индуктивното учење и за невронските мрежи



Класификација: множеството за обука се состои од група на записи кои содржат атрибути од кои едниот атрибут е класа. Задачата е да се најде моделот на класниот атрибут како функција од другите атрибути. Множеството за тестирање се користи за валидација на моделот.

Множество за обука

- ефиниција: множество примери кои се користат за да се научи системот да генерира точен одговор.
- Постапка:
 - Од популацијата се одбира примерок кој е претставен во форма на влезен вектор за кој се генерира вектор одговор.
 - Притоа, целта е одговорот да се совпадне со излезот, т.е. да се минимизира средно-квадратното отстапување на излезот и одговорот, кое ја претставува грешката на обучувањето.
 - Обучувањето завршува кога грешката ќе биде помала од однапред дефинираната.

Множество за исправност

- ефиниција: множество примери кои се користат за да се нагодат параметрите дефинирани при обучувањето.
- Постапка:
 - ☐ Множеството за исправност е множество независно од множеството за обука.
 - ☐ Со него се проверува исправноста на предвидените релации воспоставени при обучувањето.
 - ☐ Грешката при генерирањето на одговорот треба да биде помала од вредност што се дефинира однапред.
 - ☐ околку овој критериум не е исполнет, тогаш треба да се нагодат параметрите на обучувањето и да се повтори обучувањето со претходно одбраното множество за обука.

Множество за тестирање

- ефиниција: множество примери со кои се мери однесувањето, односно способноста за генерализација на обучениот систем.
- Постапка:
 - По воспоставувањето на релациите кои се одобрени со тестот за исправност, се одбираат примери различни од двете претходни.
 - За секој влезен пример се генерира одговорот и се споредува со очекуваниот излез.
 - Со тоа се проценува успешноста на способноста за генерализација

Што се прави кога нема доволно податоци

- Во отсуство на доволно податоци, множеството за обучување се разделува на две подмножества, едно кое ќе се користи за обука, а второ за исправност.
- Ова разделување се вика вкрстена исправност (cross-validation).

Техники за вкрстената исправност

- Случајно одбирање (random sampling)
 - основното множество случајно се дели на два дисјунктни дела со различна големина, од кои едното е за обука, а другото за исправност
- К-кратна поделба (K-fold cross validation)
 - основното множество случајно се дели на K дисјунктни делови со сличен обем, од кои $K-1$ се за обука, а последното за исправност
 - притоа, секое од K -те множества еднаш се зема за тестирање, што значи дека постапката се повторува K пати
- Поделба со оставање на еден (leave-one-out cross validation)
 - ако множеството е многу мало за да се дели на K делови, во тој случај, $n-1$ примери од вкупно n -те достапни се користат за обука, а само еден за тестирање и постапката се повторува n пати

Пример за македонскиот јазик

- Цел: учење на правила според кои се збороформираат именките (т.е. како од именката дете да се добијат збороформите: детето, детево, детено, деца, децата, децава, децана)
- Множество за обука и оценка на исправноста: сите именки од една книга.
- Разделување на овие множества: 10fold crossvalidation, т.е. случајно делење на сите именки на 10 групи, од кои едната се користи за обука, а останатите за исправност.
- Тестирање: именките од речникот кои ги нема во книгата.

Користена литература

- Artificial Intelligence, A Modern Approach
2nd edition, Russel and Norvig
- ожиновски С., Вештачката
интелигенција, Гоцмар, Скопје, 1994



Прашања?