Gépi tanulás

Tanulás fogalma

- Egy algoritmus akkor tanul, ha egy feladat megoldása során olyan változások következnek be a működésében, hogy később ugyanazt a feladatot vagy ahhoz hasonló más feladatokat jobb eredménnyel, illetve jobb hatékonysággal képes megoldani, mint korábban.
- A tanulás során változhat a feladat
  - reprezentációja (logikai formulák, valószínűségi hálók)
  - megoldó algoritmusa (mély hálók, genetikus programozás)
  - heurisztikája (B’ algoritmus)

Tanulási modellek

- Ha a megoldandó problémát egy $\varphi : X \to Y$ leképezés modellezi, akkor ehhez azt az $f : X \to Y$ leképezést kiszámító algoritmust keressük (tanuljuk meg), amelyre $f \approx \varphi$
  - sokszor egy rögzített $f : P \times X \to Y$ leképezést használunk, és annak azon $\Theta \in P$ paraméterét keressük, amelyre $f(\Theta, x) \approx \varphi(x)$
- *Induktív tanulási modell*
  - f leképezést (illetve annak paraméterét) $x_n \in X$ $(n=1..N)$ bemenetek (minták) alapján tanuljuk
- *Adaptív (inkrementális) tanulás*
  - Egy már megtanult f leképezést egy új minta anélkül módosít, hogy a korábbi mintákat újra meg kell vizsgálnunk.

Induktív modellek tanulási módjai

- *Felügyelt tanulás*: ismeri a tanuláshoz használt minták elvárt kimenetét is, azaz az $(x_n, \varphi(x_n))$ $(n=1..N)$ input-output párok alapján tanul.
- *Felügyelet nélküli tanulás*: nem ismeri a tanuláshoz használt minták elvárt kimenetét, csak x_n $(n=1..N)$ lehetséges inputokat; a minták illetve az azokra kiszámolt kimenetek közötti összefüggéseket próbálja felismerni, azokat osztályozni.
- *Megerősítéses tanulás*: nem ismeri ugyan a tanuláshoz használt minták elvárt kimenetét, de képes az x_n $(n=1..N)$ inputokra kiszámolt eredményt minősíteni, hogy az mennyire megfelelő.

1. Felügyelt tanulás

- A problémát modellező $\varphi : X \to Y$ leképezés közelítéséhez választunk egy $f : P \times X \to Y$ paraméteres leképezést, majd ennek azon $\Theta \in P$ paraméterét keressük (*paraméteres tanulás*), amelyre az (x_n, y_n) $(n=1..N)$ tanító minták mellett (ahol $y_n = \varphi(x_n)$) az alábbi hiba már elég kicsi (ettől reméljük, hogy $f(\Theta, x) \approx \varphi(x)$)

$$\frac{1}{N} \sum_{n=1}^{N} \ell\left(f(\Theta, x_n), y_n \right)$$

hiba függvény

elvárt kimenet

t_n

számított kimenet

- $\ell : Y \times Y \to \mathbb{R}$ hibafüggvény:
  - $\ell(t_n, y_n)$ lehet például $\|t_n - y_n\|_1$, $\|t_n - y_n\|_2^2$, vagy $-\Sigma_i\, y_{n\,i} \cdot \log t_{n\,i}$.

Megjegyzés

- Fontos, hogy az $f(\Theta, x)$ kiszámítása gyors legyen; nem baj, ha a megfelelő Θ megtalálása lassú, hiszen ezt a tanító minták segítségével előre számoljuk ki.
- A Θ megtanulása akkor működik jól, ha
  - N elég nagy (Ugyanakkor számolni kell azzal, hogy a mintákat drága összegyűjteni, a $\varphi(x_n)$-eket költséges kiszámolni.)
  - f és ℓ megfelelőek (ehhez tapasztalat, sok próbálkozás kell)
  - Θ közel esik a paraméter globális optimumához
- A Θ megtalálása egy nemkonvex optimalizálási feladat: a Θ globális optimumát megtalálni NP-teljes probléma. Szerencsére ez nem is cél, mert ezzel túl mohó módszert kapnánk (túltanulás), amely a tanító mintákra tökéletes, de egyébként nem.

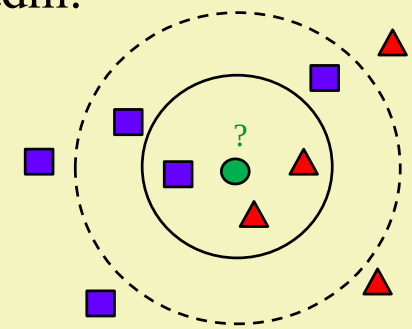
1.1. K legközelebbi szomszéd

- Az f függvény veszi a minták közül az $x \in X$ bemenethez legközelebb eső bemenettel rendelkező K darab mintát, és ezek kimenetei alapján (pl. átlagolással) határozza meg az x kimenetét:

igaz állításra 1-et, különben 0-t ad

$$f(\Theta, x) = \sum_{n=1}^{N} \frac{\mathbb{I}(x_n \text{ az egyike az } x\text{-hez legközelebb eső } K \text{ darab tanító minta inputjainak})}{K} \cdot y_n$$

  - a Θ paramétert (ami egyrészt a mintákból, másrészt a $K \in \mathbb{N}$ számból áll) nem kell optimalizálni, hanem előre meg kell adni.
  - a legközelebbi szomszédokat az $\|x_n - x\|_2^2$ távolságok sorba rendezésével választjuk ki
  - előny: egyszerű leprogramozni, a „tanulás” gyors
  - *hátrány*: ha N nagy, a tárolás, és a minták sorba rendezése erőforrásigényes, az f kiszámítása lassú



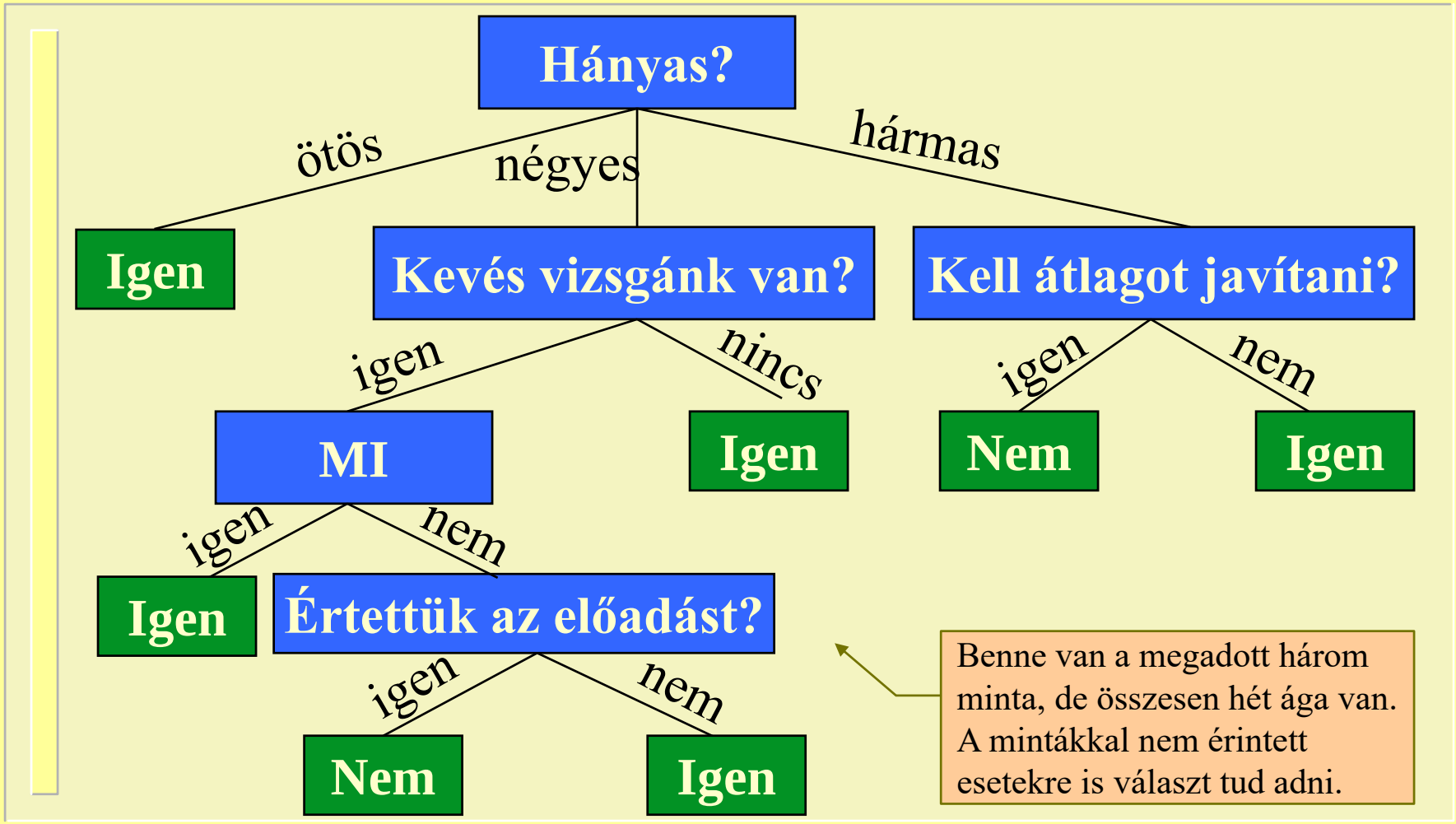
1.2. Döntési fa

- Tegyük fel, hogy az $x \in X$ bemeneteknek ugyanazon tulajdonságait (adott attribútumok értékeit) ismerjük, azaz egy bemenet attribútum-érték párok halmazával jellemezhető.
- Képzeljük el azt az irányított fát, amelynek
  - belső csúcsai egy-egy attribútumot szimbolizálnak, és az abból kivezető éleket ezen attribútum lehetséges értékei címkézik
  - ágai attribútum-érték párok halmazát jelölik ki
  - leveleihez azon tanító minták rendelhetők, amelyeket a levélhez vezető út attribútum-érték párjaival rendelkeznek.
- Egy x bemenet az attribútum-értéke párjai alapján a döntési fa egyik levelére képezhető le, és ekkor a levélhez tartozó minták kimenetei alapján számítható az x-hez tartozó kimenetet.

Példa: Elfogadjuk-e a megajánlott vizsgajegyet?

- Minták (attribútum-érték párok és a válasz):
  - Ha az ötös, akkor feltétlenül.
  - Ha négyes és kevés vizsgánk van és értettük az előadást, akkor nem; feltéve, hogy a tárgy nem a mesterséges intelligencia.
  - Ha hármas és az átlagot kell javítanunk, akkor nem.
- Attribútumok és lehetséges értékeik:
  - hányast ajánlottak meg (1, 2, 3, 4, 5)
  - kevés vizsgánk van-e (igen, nem)
  - kell-e átlagot javítani? (igen, nem)
  - az MI tárgyról van-e szó? (igen, nem)
  - értettük-e az előadást? (igen, nem)

A példa egy döntési fája



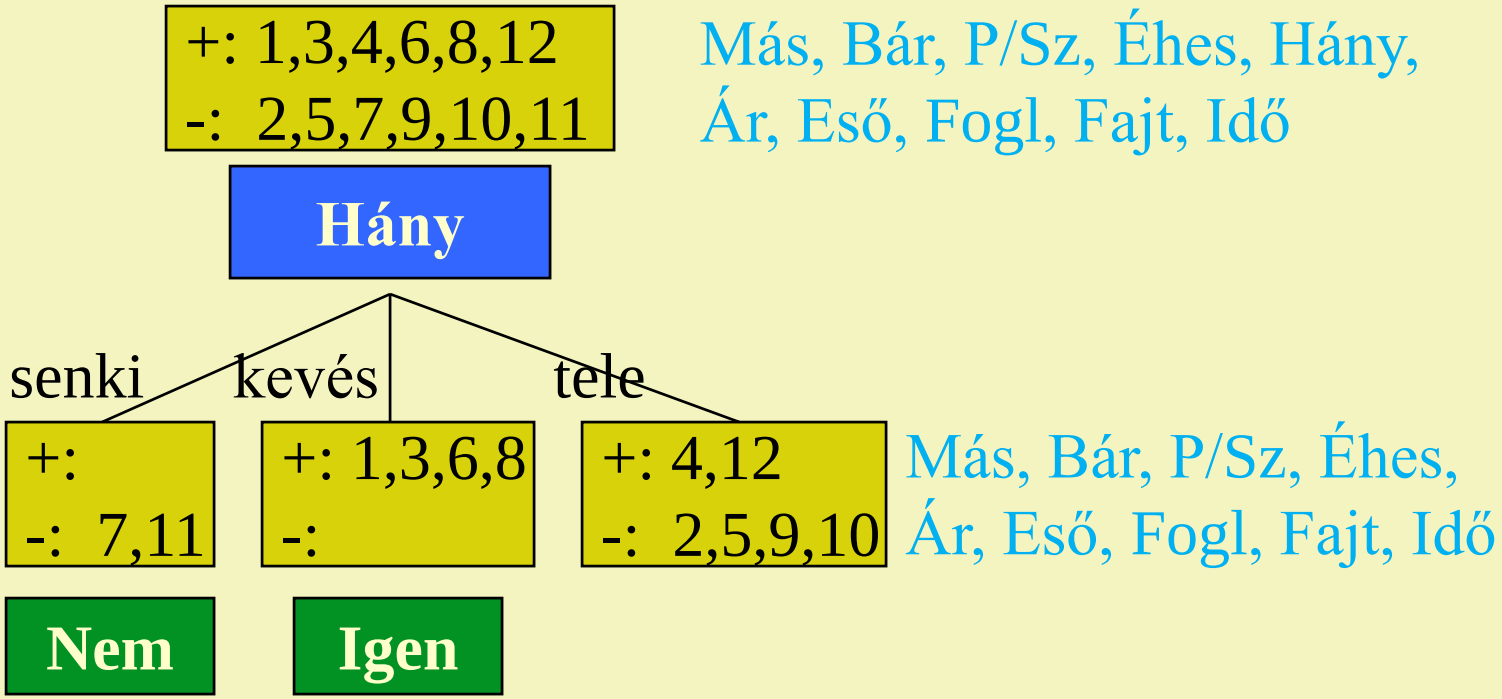
Döntési fa felépítése

- A döntési fát egy (x_n, y_n) $(n=1..N)$ tanító mintahalmaz segítségével építjük fel (ahol $y_n = \varphi(x_n)$).
  - Az építés során egy csúcshoz a tanító minták egy részhalmaza tartozik, amelyet a csúcshoz választott attribútum diszjunkt részekre vág szét, és e részeket a csúcs gyermekei kapják meg.
  - Egy levélcsúcs értékét ezen csúcshoz tartozó tanító minták kimenetei adják: ez lehet az átlaguk vagy leggyakoribb értékük. (Ha ez nem dönt, akkor a levélcsúcs szülőcsúcsának mintáit vizsgáljuk.)
- Egy tanító mintahalmazhoz több döntési fa is megadható.
- A legkisebb (legtömörebb) döntési fa megadása egy NP-teljes probléma.

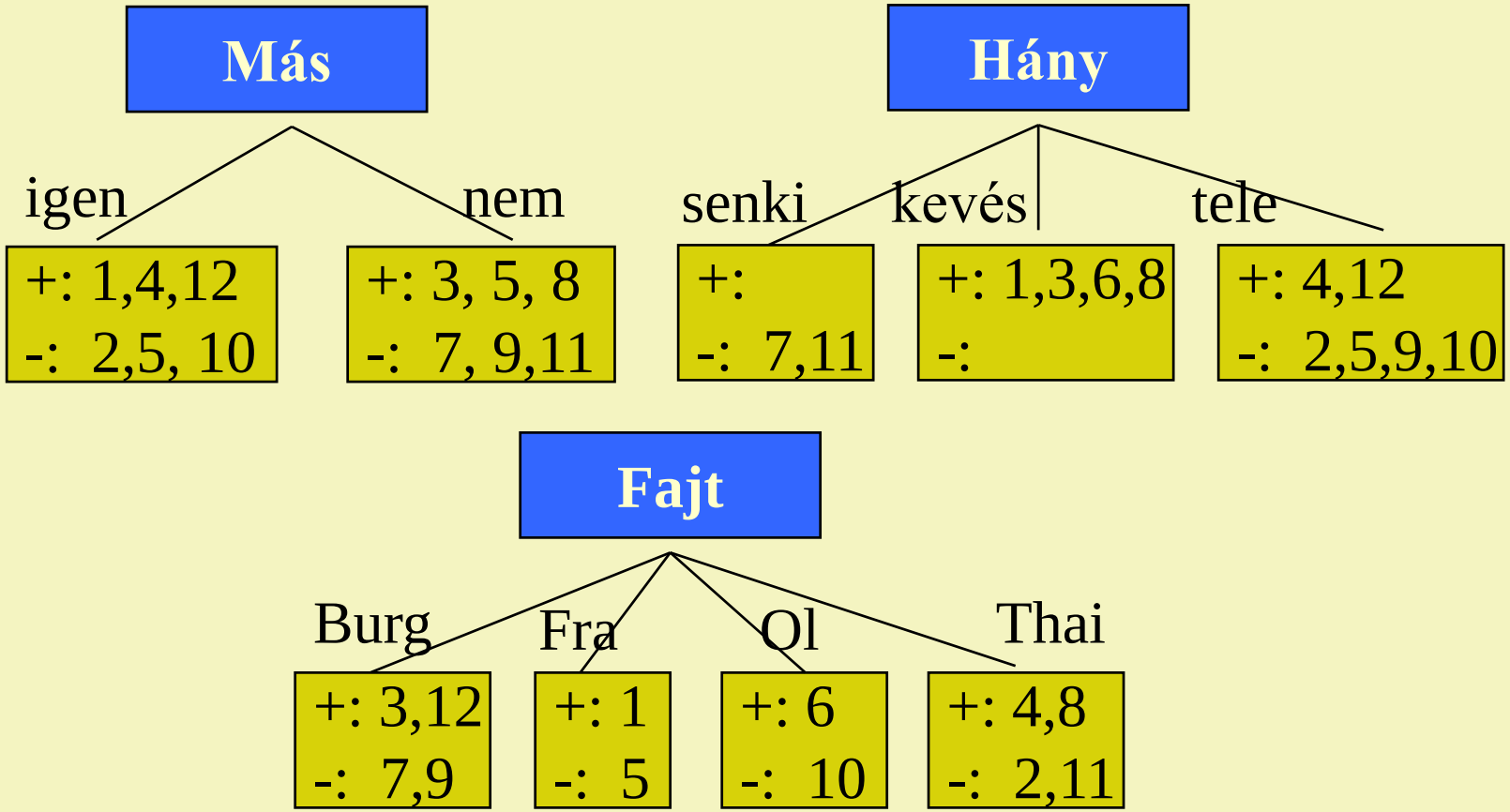
Étterem probléma (Russel-Norvig)

Pl.	Más	Bár	P/Sz	Éhes	Hány	Ár	Eső	Fogl	Fajt	Idő	Marad
1	I	N	N	I	kevés	drá	N	I	Fra	10	I
2	I	N	N	I	tele	olcs	N	N	Tha	60	N
3	N	I	N	N	kevés	olcs	N	N	Bur	10	I
4	I	N	I	I	tele	olcs	N	N	Tha	30	I
5	I	N	I	N	tele	drá	N	I	Fra	sok	N
6	N	I	N	I	kevés	köz	I	I	Ol	10	I
7	N	I	N	N	senki	olcs	I	N	Bur	10	N
8	N	N	N	I	kevés	köz	I	I	Tha	10	I
9	N	I	I	N	tele	olcs	I	N	Bur	sok	N
10	I	I	I	I	tele	drá	N	I	Ol	30	N
11	N	N	N	N	senki	olcs	N	N	Tha	10	N
12	I	I	I	I	tele	olcs	N	N	Bur	60	I

Döntési fa építésének első lépése



Alternatív lépések



Heurisztika

- A döntési fa minél tömörebb, egy-egy ága minél rövidebb lesz, ha
  - egy csúcshoz kiválasztott attribútum (a) a csúcshoz tartozó mintákat olyan részhalmazokra vágja szét, amelyeken belül a minták minél homogénebbek, minél kevésbé különböznek,
  - ezt valamilyen távolság fogalom (2-es norma, kereszt entrópia) alapján vizsgálhatjuk
    - Pl.: a szétvágás információs előnyét – a szétvágás előtti minta-halmaz információ tartalmának (entrópiájának) és az utána kapott minta-részhalmazok információ tartalmának (számosságuk szerinti súlyozott) összege közti különbséget – maximalizáljuk.

Információ tartalom (Entrópia)

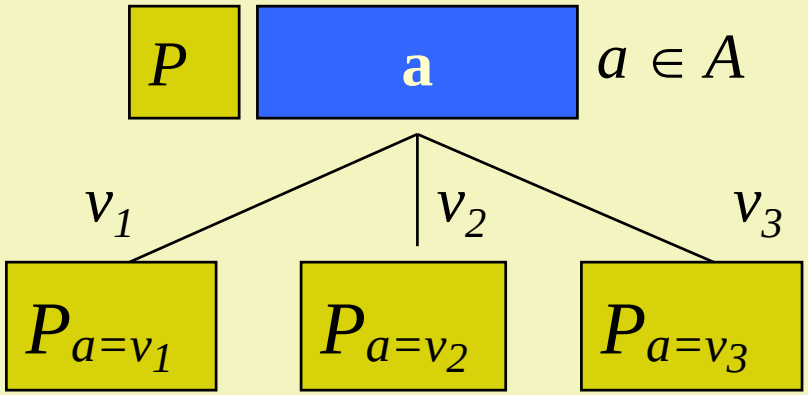
- A *P*-beli minták információtartalma (entrópiája), ha csak kétféle kimenetű minta van:
  - $E(P) = E(p^+, p^-) = - p^+ \log_2 p^+ - p^- \log_2 p^-$
  - ahol p^+ a *P*-beli pozitív, p^- a negatív minták aránya ($p^+ + p^- = 1$)
- Példa:
  - Ha *P*-ben 2 pozitív és 3 negatív minta van:

    $$E(P) = E(2/5, 3/5) = 0.97$$

  - Ha *P*-ben 0 pozitív és 3 negatív minta van:

    $$E(P) = E(0/3, 3/3) = 0$$

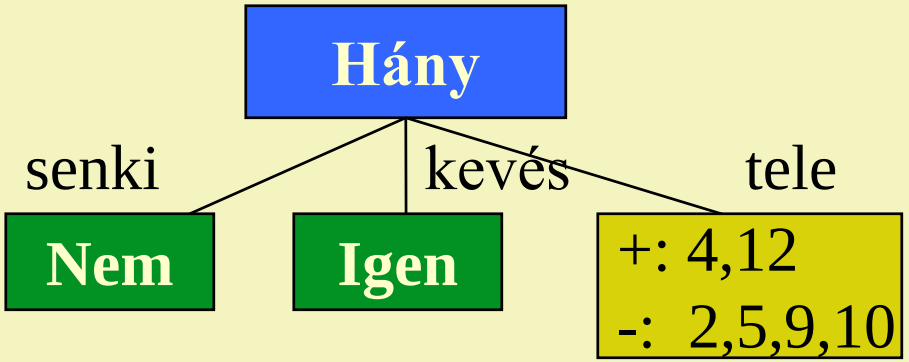
Információs előny számítása



$$C(P,a) = E(P) - \sum_{v \in Érték(a)} \frac{|P_{a=v}|}{|P|} E(P_{a=v})$$

- ahol P a szülő csúcs mintái, a a választott attribútum,
- az *Érték(a)* az a attribútum által felvett értékek, és
- a $P_{a=v} = \{ p \in P \mid p.a=v \}$

Egy csúcs attribútumának kiválasztása 1.



$E(\{2,4,5,9,10,12\}) =$
$= E(2/6,4/6) = 0.92$

Más, Bár, P/Sz, Éhes,
Ár, Eső, Fogl, Fajt, Idő

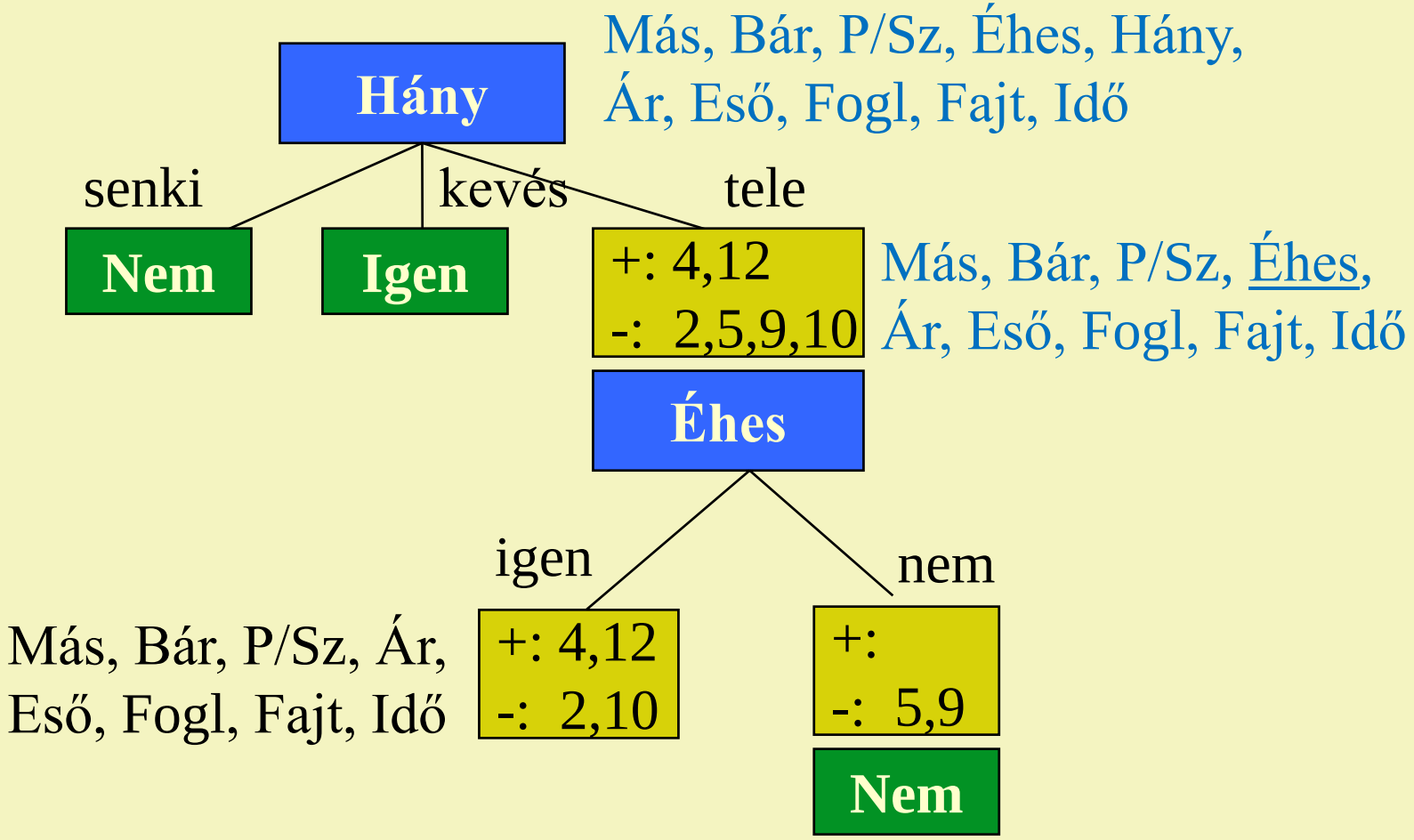
- Ha a *Más* attribútumot választjuk, akkor a minták 1:5 arányban ketté válnak: {9} (*Más= hamis*), és {2, 4, 5, 10, 12} (*Más=igaz*),
  - $E(\{9\}) = E(0/1, 1/1) = 0$
  - $E(\{2,4,5,10,12\}) = E(2/5, 3/5) = 0.97$
- Az információs előny: $C(\{2,4,5,9,10,12\}, Más) =$
  $E(\{2,4,5,9,10,12\}) - (1/6\ E(\{9\}) + 5/6\ E(\{2,4,5,10,12\})) =$
  $E(2/6,4/6) - (1/6\ E(0/1,1/1) + 5/6\ E(2/5,3/5)) = 0.92 - 0.81 = 0.11$

Egy csúcs attribútumának kiválasztása 2.

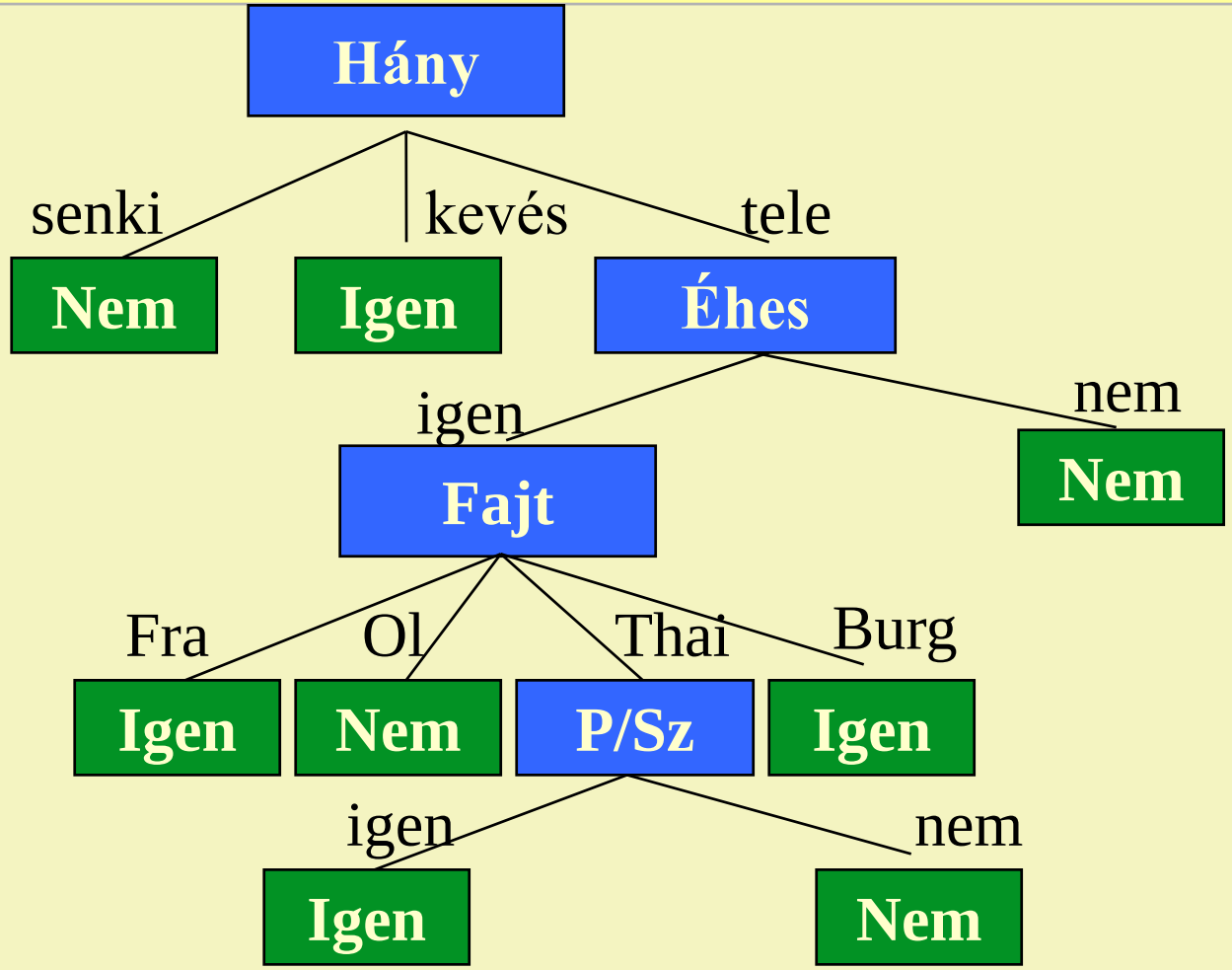
$C(\{2,4,5,9,10,12\},a) = 0.92$ -

Más:	$1/6\ E(0/1,1/1) + 5/6\ E(2/5,3/5) =$	*0.81*
Bár:	$3/6\ E(1/3,2/3) + 3/6\ E(1/3,2/3)) =$	*0.92*
P/Sz:	$1/6\ E(0/1,1/1) + 5/6\ E(2/5,3/5) =$	*0.81*
Éhes:	$4/6\ E(2/4,2/4) + 2/6\ E(0/2,2/2) =$	*0.67*
Ár:	$4/6\ E(2/4,2/4) + 0/6\ E(0,0) + 2/6\ E(0/2,2/2) =$	*0.67*
Eső:	$5/6\ E(2/5,3/5) + 1/6\ E(0/1,1/1) =$	*0.81*
Fog:	$4/6\ E(2/4,2/4) + 2/6\ E(0/2,2/2) =$	*0.67*
Fajt:	$2/6\ E(1/2,1/2) + 1/6\ E(0/1,1/1) + 1/6\ E(0/1,1/1) + 2/6\ E(1/2,1/2) =$	*0.67*
Idő:	$0/6\ E(0,0) + 2/6\ E(1/2,1/2) + 2/6\ E(1/2,1/2) + 2/6\ E(0/2,2/2) =$	*0.67*

További lépések



Étterem probléma döntési fája



Készítsünk algoritmust

- Egy fokozatosan épülő döntési fában a csúcsokhoz a tanító minták egy részhalmaza, valamint a még választható (a csúcshoz vezető út csúcsainak címkéiben nem szereplő) attribútumok tartoznak. Ezek a csúcsok lehetnek
  - attribútummal címkézett belső csúcsok, amelyekből kivezető élek az attribútum lehetséges értékeit képviselik
  - kiértékelt vagy értékkel nem rendelkező levélcsúcsok
- Minden lépésben egy értékkel még nem rendelkező levélcsúcsról kell eldönteni, hogy kaphat-e értéket vagy belső csúcs legyen-e.
  - Előbbi esetben az értéke a csúcshoz tartozó minták értékei alapján (átlag vagy leggyakoribb érték) számolható.
  - Utóbbi esetben egy attribútumot választunk címkéjének, és generáljuk a gyerekeit.

Algoritmus

- Kezdetben a fa egyetlen címkézettlen csúcsból áll (ez lesz majd a gyökér), amelyhez az összes mintát és attribútumot rendeljük.
- Veszünk egy értékeletlen levélcsúcsot:
  1. Ha $A=\varnothing$, akkor a mintái alapján kiértékeljük.
  2. Ha $P=\varnothing$, akkor a szülőcsúcsának mintái alapján kiértékeljük.
  3. Ha P csupa azonos kimenetű mintából áll, akkor a mintái alapján kiértékeljük.
  4. Egyébként ...

Algoritmus (folytatás)

4. Egyébként a legnagyobb információs előnnyel járó $a \in A$ attribútummal címkézzük az adott csúcsot, majd generáljuk a gyerekeit:

   a) Ezekhez az *a* lehetséges értékeivel címkézett élek vezetnek.

   b) Ha az *a* címkéjű csúcsból egy gyerekcsúcsába a *v* címkéjű él vezet, akkor a gyerekcsúcshoz rendelt

   - minták: $P_{a=v} = \{ p \in P \mid p.a = v \}$
   - választható attribútumok: $A = A - \{a\}$

   c) Végül minden gyerekre ismételjük meg rekurzív módon az 1-4 pontokat.

Megjegyzés

- Zaj: Két vagy több eltérő besorolású minta attribútum-értékei megegyeznek.
  - Ilyenkor a minták válaszainak átlagolása félrevezethet
- Túlzott illeszkedés: A bemenetek olyan attribútumait is figyelembe veszünk, amelyek a kimenetre nincsenek hatással. (Például egy kocka dobás eredményére annak színe és dátuma alapján értelmetlen szabályszerűségeket találunk.)
  - A lényegtelen attribútumokat ($C(P,a) \sim 0$) állítsuk félre.
- Általánosítások:
  - Hiányzó adatok (attribútum értékek) problémája
  - Folytonos értékű attribútumok

Tanulás döntési fával

- Egy $x \in X$ bemenethez azon tanító minták kimenetei alapján számol kimenetet, amely minták az előzetesen felépített döntési fában az x-re kiszámolt levélcsúcshoz tartoznak

$$f(\Theta, x) = \sum_{n=1}^{N} \frac{\mathbb{I}(\text{az } x\text{-re kiszámolt levélcsúcs } K' \text{ darab tanító mintájának egyike az } x_n)}{K'} \cdot y_n$$

amikor x kimenete a hozzá kiszámolt levélcsúcs mintái kimeneteinek átlaga

  - Itt a Θ a döntési fa, optimalizálása annak mohó felépítése
  - előny: jól értelmezhető (a mintákra tökéletes eredményt, a mintákhoz hasonló inputokra többnyire jó eredményt ad);
    a tanító minták helyett csak a döntési fát kell tárolni;
    x-re adott eredmény gyorsan számolható
  - hátrány: a faépítés NP-teljes, mohó módszerrel csak lokálisan optimális

1.3. Véletlen erdő

- *K* darab döntési fát építünk a tanító minták alapján úgy, hogy egy-egy fa építéséhez a tanító mintáknak is, és a minták attribútumainak is csak egy-egy véletlen kiválasztott részhalmazát használjuk fel. Ez lesz a véletlen erdő.
- Egy véletlen erdő minden fájában külön-külön megállapíthatjuk, hogy egy $x \in X$ bemenet a döntési fa melyik levelére képződik le. Ezen levelekhez tartozó tanító mintahalmazok kimeneteinek súlyozott átlagával becsüljük az *x* kimenetét.

Tanulás véletlen erdővel

- Egy x bemenethez tartozó kimenetet a minták kimeneteinek súlyozott átlaga, ahol a súlyok attól függenek, hogy egy minta a véletlen erdő döntési fáinak x-re kiszámolt levélcsúcsaihoz tartozó mintahalmazok közül hányba esik bele, és az a halmaz hány elemű:

$$f(\Theta, x) = \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{\mathbb{I}(\text{az } x_n \text{ a } k\text{-adik fa } x\text{-re kiszámolt levélcsúcsához t artozó } K_k(x) \text{ darab mintának az egyike})}{K \cdot K_k(x)} \cdot y_n$$

  - a Θ maga a véletlen erdő, optimalizálása az erdő felépítése
  - előny: a tanító minták helyett csak az erdőt kell tárolni; a véletlen generálás miatt kevésbé mohó, elkerüli a túltanulást; az x-re adott eredmény számolása párhuzamosítható
  - hátrány: az eredmény kevésbé értelmezhető; az erdő-építés NP-teljes