Osztályozás és regresszió II.

Benke Tibor

Tartalom

- Döntési szabályok
- Döntési fák
- Bayes hálózatok
- Szupport Vektor Gépek (SVM)

Döntési szabályok (Tartalom)

- Szabályhalmazok és szabálysorozatok
- Döntési táblázatok
- Az 1R algoritmus
- A Prism módszer

Szabályok

- Felépítés:
  - feltételrész -> következményrész
- Példa:
  - HŐMÉRSÉKLET = magas AND SZÉL = nincs → IDŐ_JÁTÉKRA alkalmas
- Valószínűségi döntési szabály:

  nem = férfi AND gyerek száma = 0 AND autó teljesítmény > 150LE → kockázatos = (80%, 20%)

Logikai műveletek

- a feltételrészben **és**, **vagy**, **tagadás** műveletek használhatók
- általában állítások **és** kapcsolatait használjuk
- azonos következménnyel rendelkező szabályok **vagy** kapcsolattal összevonhatók

Döntési szabály

Def.: **döntési szabály**: Az A attribútumhalmaz felett értelmezett döntési szabály alatt egy $R : \varphi(A) \rightarrow Y = y$ logikai implikációt értünk, amelyek feltételrészében szereplő φ egy logikai formula, amely az A -beli attribútumokra vonatkozó feltételek logikai kapcsolataiból áll. A szabály következményrészében az osztályattribútumra (magyarázott változóra) vonatkozó ítélet szerepel.

Illeszkedés, fennállás

- Def.: Az R : φ(A) → Y = y szabályra **illeszkedik** a **t** objektum, ha a feltételrész attribútumváltozóiba a **t** megfelelő értékeit helyettesítve igaz értéket kapunk.
- Ha a szabály következményrésze is igazra értékelődik ki az objektumon, akkor a szabály **fennáll** vagy más néven **igaz**.

Fedés

- Def.: Az R : φ(A) → Y = y szabály lefedi a T objektumhalmazt, ha minden objektum illeszkedik a szabályra. Adott T tanító halmaz esetén az R által fedett tanítópontok halmazát **cover $_T$(R)** -rel jelöljük.
- **helyes fedés**: R fedi T-t és minden objektum y-ba tart.
- **helytelen fedés** (rossz osztályozás): R fedi T-t, de nem minden objektum tartozik y-ba.

Jelölések

- $cover_T(R)$: fedett pontok halmaza
- $cover^+_T(R)$: helyesen fedett pontok halmaza
- $cover^-_T(R)$: helytelenül fedett pontok halmaza
- relatív fedési hiba: $ErT(R) = \frac{cover^-_T(R)}{cover_T(R)}$

Példa

- Objektum:

HŐMÉRSÉKLET	PÁRATARTALOM	SZÉL	ESŐ	IDŐ_JÁTÉKRA
magas	alacsony	nincs	van	alkalmatlan

- Szabály:

$$\texttt{HŐMÉRSÉKLET} = \texttt{magas AND SZÉL} = \texttt{nincs} \rightarrow \texttt{IDŐ\_JÁTÉKRA alkalmas}$$

- Illeszkedik-e?
- Helyes-e a fedés?

Példa

- Objektum:

HŐMÉRSÉKLET	PÁRATARTALOM	SZÉL	ESŐ	IDŐ_JÁTÉKRA
magas	alacsony	nincs	van	alkalmatlan

- Szabály:

$$\texttt{HŐMÉRSÉKLET} = \texttt{magas AND SZÉL} = \texttt{nincs} \rightarrow \texttt{IDŐ\_JÁTÉKRA alkalmas}$$

- Illeszkedik-e? - IGEN
- Helyes-e a fedés? – NEM (más a következmény)

Döntési szabályok kifejezőereje

- Ítéletkalkulus alapú
  - A feltételrészben predikátumok logikai kapcsolata szerepel
  - Kategória: A = a, a ϵ A
  - Sorrend, intervallum: a’ < A < a’’
  - Algoritmus: Iteratívan fedjük le a tanítóhalmazt, először nagy
  - Példa: $\text{MAGASSÁG} \leq 170 \text{ AND HAJSZÍN} = \text{barna AND SZEMSZÍN} \in \{\text{kék, zöld}\}$
- Reláció alapú
  - A relációk mindkét oldalán állhat változó
  - A kifejezőerejük megegyezik az ítéletkalkulus alapú szabályokéval (véges értékkészletű attribútumok esetén)

Döntési szabályok kifejezőereje 2.

- Induktív logikai programozás
  - Rekurzív kifejezéseket használ
  - Példa:

```
szélesség < magasság → ALAK = álló
```

    - Építőelem álló, ha a szélessége kisebb mint a magassága
    - Torony: csúcs + maradék

```
szélesség(torony.csúcs) < magasság(torony.csúcs) AND
        álló(torony.maradék) → álló(torony).
torony = üres → álló(torony)
```

    - Miért kell az utolsó szabály?
  - A rekurzív szabályokat is tartalmazó szabályhalmaz neve logikai

Szabályrendszerek

- Szabálysorozatok
  - A szabályok sorrendje számít
  - Az első illeszkedő szabály fogja meghatározni az objektum osztályát
  - Alapértelmezett szabály szerepe a sorozat végén
- Szabályhalmazok
  - A szabályok függetlenek egymástól
  - Egyértelmű, ha bármely objektum csak egy szabályra illeszkedik
  - Képezhető sorozatból: szabály + előtte álló szabályok negáltjainak és kapcsolata elé fűzve
- Teljes, ha tetszőleges objektum illeszthető egy szabályra (mindig születik döntés)

Döntési táblázatok

időjárás	hőmérséklet	páratartalom	szél	játékidő?
napos	meleg	magas	nincs	nem
napos	meleg	magas	van	nem
borús	meleg	magas	nincs	nem
esős	enyhe	magas	nincs	igen
esős	hideg	magas	nincs	igen
esős	hideg	magas	nincs	igen
esős	hideg	magas	nincs	igen

- Mik a fontos attribútumok?
- Hogyan diszkretizáljuk a folytonos attribútumokat?

Az 1R algoritmus

- Az egyik legegyszerűbb osztályozó algoritmus
- Kiválaszt egy A attribútumot
- Legyárt |A| darab szabályt: $A = a \rightarrow Y = c$;
- A **c** az adott **a** érték mellett leggyakrabban előforduló osztály
- A legkevesebb rosszul osztályozott tanítópontot adó A attribútumot választjuk
- Sorrend és intervallum típusú $a''' \leq A \rightarrow y$ $A \leq a \rightarrow y,\ a' \leq A < a'' \rightarrow y$
- Példa:

$$\text{hőmérséklet} = \text{meleg} \rightarrow \text{játékidő} = \text{nem}$$
$$\text{hőmérséklet} = \text{enyhe} \rightarrow \text{játékidő} = \text{igen}$$
$$\text{hőmérséklet} = \text{hideg} \rightarrow \text{játékidő} = \text{igen}$$

- Egy kutatás szerint alig marad el az újabb, bonyolultabb osztályozók hatásfokától

Prism módszer (bevezető)

Prism módszer

- Feltételezi, hogy a tanító adatbázisban nincs két olyan elem, melyek fontos magyarázó attribútumai megegyeznek, de más osztályba tartoznak
- Ha vannak ilyenek, csak a leggyakoribb osztályba tartozót tartsuk meg
- Fedő algoritmus:
  - Szabályokat állít elő

Algoritmus Prism

Require: $\mathcal{T}$: tanítópontok halmaza,
Y: osztályattribútum változó,

for all $y \in$ osztályattribútum értékre **do**
  $E \leftarrow$ az y osztályba tartozó tanítópontok
  $\phi \leftarrow \emptyset$
  while $E \neq \emptyset$ **do**
    $R \leftarrow (\phi \rightarrow Y = y)$
    while $Er_{\mathcal{T}}(R) \neq 0$ **do**
      hiba $\leftarrow 1$
      for all (A, a) attribútum-érték párra **do**
        if $Er(\phi \text{ AND } A = a \rightarrow Y = y) < hiba$ **then**
          hiba $\leftarrow Er(\phi \text{ AND } A = a \rightarrow Y = y)$
          $A* \leftarrow A$
          $a* \leftarrow a$
        end if
      end for
      $\phi \leftarrow (\phi \text{ AND } A* = a*)$
    end while
    $\mathcal{T} \leftarrow \mathcal{T} \setminus cover(R)$
  end while
end for

- E: pontok a -ből
- E: $Y = y$ pontok a T -ből
- R: a feltételrésze üres, a következményrésze y
- Míg a relatív fedési hiba nem nulla
- Minden (A, a) párra kiszámoljuk, hogy mennyi lenne a helytelenül osztályozott tanítópontok száma, ha az $A = a$ részfeltételt adnánk a feltételhez
- A legkisebb relatív fedési hibájú részfeltételt választjuk
- Töröljük a lefedett szabályokat a tanítóhalmazból

- E: pontok a -ből
- R: a feltételrésze üres, a következményrésze y
- Míg a relatív fedési hiba nem nulla
- Minden (A, a) párra kiszámoljuk, hogy mennyi lenne a helytelenül osztályozott tanítópontok száma, ha az $A = a$ részfeltételt adnánk a feltételhez
- A legkisebb relatív fedési hibájú részfeltételt választjuk
- Töröljük a lefedett szabályokat a tanítóhalmazból

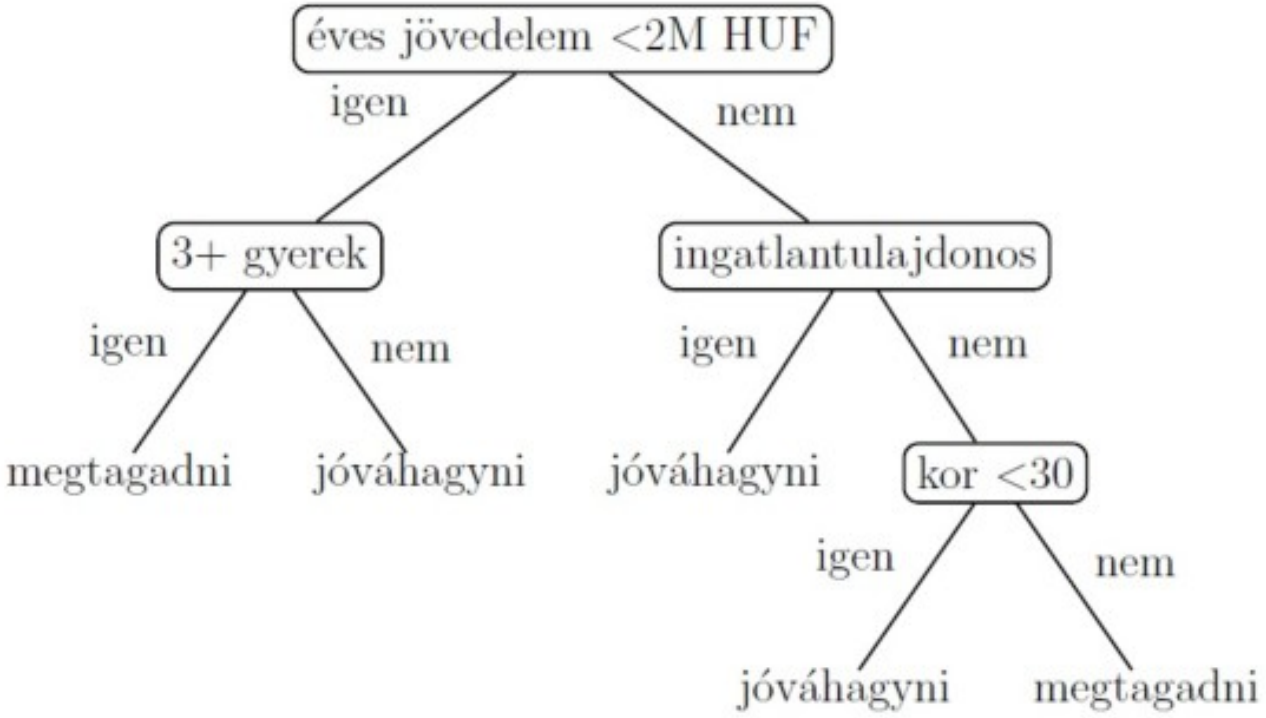
Prism módszer 2.

- A létrehozott szabályokat szabálysorozatként kell értelmezni
- Csak 100%-os pontosságú szabályokat állít elő (Példa)
- Javítás:
  - Ne $\phi \text{ AND } A = a \rightarrow Y = y$ ján válasszunk attribútumot, hanem az információnyereség alapján:
  - $$\text{hiba}^* = cover^+(R) \cdot \left[\log(Er(R)) - \log(Er(\phi \rightarrow Y = y))\right]$$

Döntési fák (Tartalom)

- Döntési fák és döntési szabályok
- A döntési fa előállítása
- Feltételek a csomópontokban
- Vágási függvények
- Döntési fák metszése
- Regressziós fák és modell fák

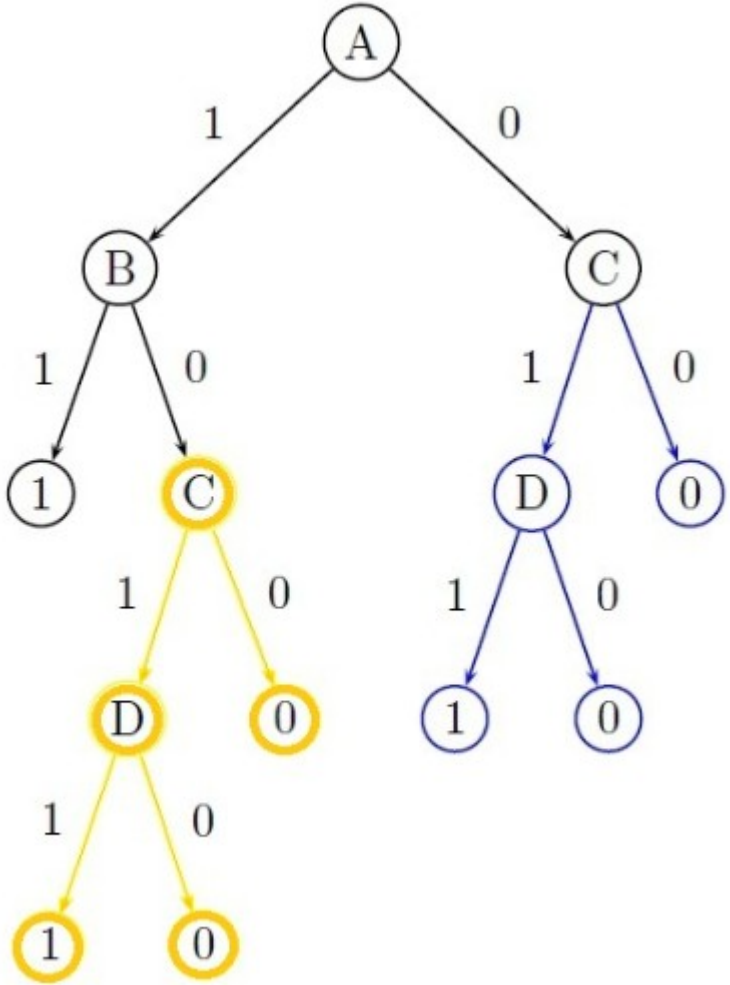
Döntési fák


éves jövedelem <2M HUF
igen
nem
3+ gyerek
ingatlantulajdonos
igen
nem
igen
nem
megtagadni
jóváhagyni
jóváhagyni
kor <30
igen
nem
jóváhagyni
megtagadni

Döntési fák jellemzői

- A lényegtelen változókat nem tesztelik
- A fontos változók, melyek jól szeparálnak a gyökérhez közel helyezkednek el
- Nagyméretű adathalmazokra is hatékonyan felépíthetők
- Egy olyan fa, melynek pontjainak kettőnél több gyermeke van mindig átrajzolható bináris fává
- **Észrevétel**: a döntési fákból nyert szabályhalmazok **egyértelműek**.
- Van, hogy a fa bo[illegible]ályok (ismétlődő részfa probléma):

1. A=1 AND B=1 → Y=1
2. C=1 AND D=1 → Y=1
3. → Y=0


A
1
0
B
C
1
0
1
0
1
C
D
0
1
0
1
0
D
0
1
0
1
0
1
0

Döntési fák előállítása

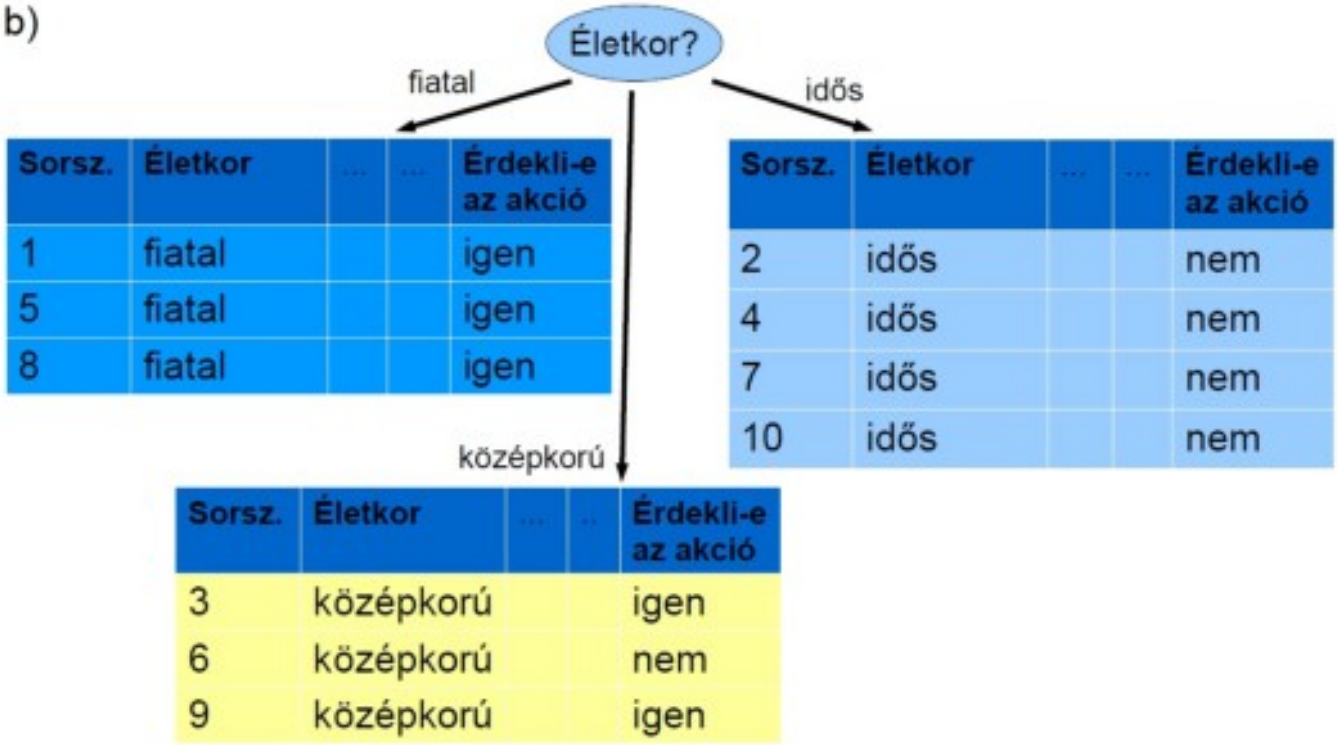
- Rekurzív módon
- Minden csúcsban olyan kérdést keresünk, hogy a magyarázott változó kevésbé legyen szórt az eredményezett halmazban
- Gyakran alkalmazunk feltételeket a levelekre (pl. tanítópontok száma)
- Leállási feltételek:
  - Nincs több vizsgálható attribútum
  - Elértünk egy bizonyos mélységet
  - Nincs olyan vágás, mellyel javítani lehetne a jelenlegi osztályozáson*
- A levelekhez döntést kell rendelni: általában többségi szavazás alapján

Példa tanítóhalmaz

Sorsz.	Életkor	Testsúly	Sportol	Érdekli-e az akció
1	fiatal	alacsony	igen	igen
2	idős	közepes	nem	nem
3	középkorú	magas	nem	igen
4	idős	közepes	igen	nem
5	fiatal	magas	nem	igen
6	középkorú	alacsony	nem	nem
7	idős	alacsony	nem	nem
8	fiatal	közepes	nem	igen
9	középkorú	magas	igen	igen
10	idős	közepes	igen	nem

Döntés az Életkor alapján

b)



Sorsz.	Életkor	...	...	Érdekli-e az akció
1	fiatal			igen
5	fiatal			igen
8	fiatal			igen

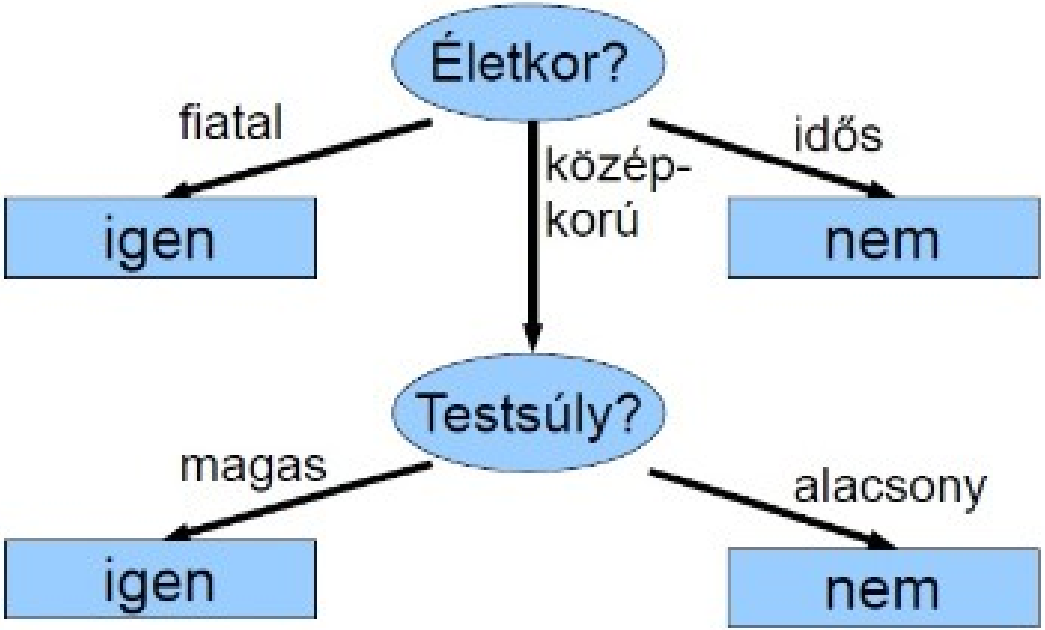
Sorsz.	Életkor	...	...	Érdekli-e az akció
2	idős			nem
4	idős			nem
7	idős			nem
10	idős			nem

Sorsz.	Életkor	...	..	Érdekli-e az akció
3	középkorú			igen
6	középkorú			nem
9	középkorú			igen

Döntés a Testsúly alapján

Sorsz.	Életkor	Testsúly	Sportol	Érdekli-e az akció
3	középkorú	magas	nem	igen
6	középkorú	alacsony	nem	nem
9	középkorú	magas	igen	igen

A végleges döntési fa



Faépítő algoritmusok

1. Algoritmusok:
    1. Iterative Dichotomizer 3 (ID3)
        1. Entrópiát számol
        2. A magyarázó és magyarázott attribútumok közötti kölcsönös információt maximalizálja
        3. Szereti azokat az attribútumokat, amik sokfelé ágaznak el
            1. Tipikusan egyedi azonosítók...
    2. Classification and Regression Trees (CART)
    3. Chi-squared Automatic Interaction Detection (CHAID)
2. Mikor jó egy vágás?
    1. Az ID3 a kölcsönös információt használja, de az miért jó?
    2. A válasz a Taylor-Silverman elvárások és a vágás jóságának fogalma

Vágás jósága

Taylor-Silverman elvárások

4.6.1. Definíció *Legyen X egy olyan diszkrét valószínűségi változó, amely k-értéket vehet fel. Az eloszlásfüggvény értékei legyenek $P = (p_1, p_2, \ldots, p_k)$. A Φ vágási függvény a $p_1, p_2, \ldots, p_k$ értékekhez rendel egy valós számot, amint látni fogjuk, Φ segítségével a vágás jóságát jellemezhetjük számszerűen. A $\Phi : [0,1]^k \longmapsto R$ vágási függvénnyel szemben támasztott Taylor-Silverman elvárások a következők:*

1. $\Phi(P) \geq 0$
2. $\Phi(P)$ *akkor veszi fel a minimumát, ha* $\exists j : p_j = 1$
3. $\Phi(P)$ *akkor veszi fel a maximumát, ha* $\forall j : p_j = 1/k$
4. $\Phi(P)$ *a P komponenseire nézve szimmetrikus, tehát a $p_1, p_2, \ldots, p_k$ értékek tetszőleges permutációjára ugyanazt az értéket adja.*
5. $\Phi(P)$ *differenciálható az értelmezési tartományában mindenhol*

Döntési fák metszése

- Feltételezzük, hogy az építés során a fát nyesszük, ezért egyszerűsíteni kell rajta
- A metszést egy a tanítóhalmaztól független adathalmazzal (T_1) szokás elvégezni
- Előmetszés:
  - A fa építése során alkalmazzuk
  - Valamilyen intelligens megállási feltétel (általában statisztikai alapokon)
- **Utómetszés** (preferált):
  - Nagy fát növesztünk, majd azt zsugorítjuk
  - **Részfa helyettesítés**: belső pontból induló részfát egy levéllel helyettesítünk, ha az jobban osztályoz a T_1 halmazon
  - **Részfa felhúzás**: Az A csúcs alatti egyik részfával helyettesítjük az A csúcsot, ha az jobban osztályoz T_1-en

Regressziós és modell fák

- Regressziós fa:
  - folytonos a magyarázandó változó, milyen értékre döntsünk?
  - vegyük a tanító adatbázisból az ezen levélhez tartozó pontok magyarázott változóinak átlagát
- Modell fa:
  - A leveleket egy-egy egyedi lineáris regressziós modell tartalmaznak
  - Levelenként nagyon eltérő lehet: tartalmazzanak a csúcsok is regressziós függvényeket
  - Ahogy a levelet megkeressük, kiértékeljük az érintett csúcsok függvényeit is

Bayes hálózatok (Tartalom)

- Naív Bayes-hálók
- Bayes hihetőségi hálók

Bayes-hálózatok

- Elv: azt az osztályt választjuk, aminek a legnagyobb a valószínűsége a megfigyelt attribútumok alapján
- Bayes-tétel: néhány valószínűség ismeretében feltételes valószínűségeket számolhatunk
- Jelölések:
  - Y_i: az osztályozandó attribútum az i. osztályba esik
  - $\vec{X}$: magyarázó attribútumok
  - $\vec{x} = (x_1, x_2, \ldots, x_k)$: egy objektum konkrét értékei
- A cél a következő valószínűség maximalizálása:

$$\mathbb{P}\left(Y_i|\vec{X}=\vec{x}\right) = \frac{\mathbb{P}\left(\vec{X}=\vec{x}, Y_i\right)}{\mathbb{P}\left(\vec{X}=\vec{x}\right)} = \frac{\mathbb{P}\left(\vec{X}=\vec{x}|Y_i\right)\mathbb{P}\left(Y_i\right)}{\mathbb{P}\left(\vec{X}=\vec{x}\right)}$$

Bayes-hálózatok

$$\mathbb{P}\left(Y_i|\vec{X}=\vec{x}\right)=\frac{\mathbb{P}\left(\vec{X}=\vec{x},Y_i\right)}{\mathbb{P}\left(\vec{X}=\vec{x}\right)}=\frac{\mathbb{P}\left(\vec{X}=\vec{x}|Y_i\right)\mathbb{P}\left(Y_i\right)}{\mathbb{P}\left(\vec{X}=\vec{x}\right)}$$

- A nevező konstans -> elég a számlálót maximalizálni
- Adott, vagy egyszerűen becsülhető relatív gyakoriságokkal: $\mathbb{P}\left(Y_i\right)$
- Cél a $\mathbb{P}\left(\vec{X}=\vec{x}|Y_i\right)$ maximalizálása
- Komplexitás:
  - k darab bináris X attribútum, Y l különböző értéket vehet fel: $\sim l*2^k$
  - k darab tetszőleges X attribútum, Y l különböző értéket vehet fel: $\sim l*n^k$
- Ez túl sok tanulópontot igényel, kell valami egyszerűsített modell

Naiv Bayes-hálók

- Feltételezik, hogy az egyes X_i attribútumok feltételesen függetlenek egymástól
- Így már csak $l * k$ darab paramétert kell megbecsülni

$$\mathbb{P}(X_1, X_2|Y_i) = \mathbb{P}(X_1|X_2, Y_i)\,\mathbb{P}(X_2|Y_i) = \mathbb{P}(X_1|Y_i)\,\mathbb{P}(X_2|Y_i)$$

$$\mathbb{P}\left(\vec{X} = \vec{x}|Y_i\right) = \mathbb{P}\left((X_1, X_2, \ldots, X_k) = (x_1, x_2, \ldots, x_k)\,|Y_i\right) = \prod_{j=1}^{k} \mathbb{P}(X_j = x_j|Y_i)$$

- A valószínűség a mintapol becsülneto
  - Kategória típusú attribútumok esetén meghatározzuk az X_j attribútumában x_j értéket felvevő elemek arányát az Y_i osztályú elemek között
- Probléma lehet, ha valamelyik relatív gyakoriság 0
- **Laplace estimation**: az adott attribútum minden előfordulásához adjunk 1-et

Példa

Sorsz.	Életkor	Testsúly	Sportol	Érdekli-e az akció
1	fiatal	alacsony	igen	igen
2	idős	közepes	nem	nem
3	középkorú	magas	nem	igen
4	idős	közepes	igen	nem
5	fiatal	magas	nem	igen
6	középkorú	alacsony	nem	nem
7	idős	alacsony	nem	nem
8	fiatal	közepes	nem	igen
9	középkorú	magas	igen	igen
10	idős	közepes	igen	nem

Valószínűségek

Jelölések:

Magyarázó attribútumok:
X: életkor
Y: testsúly
Z: sportol

Magyarázandó attribútum (osztályváltozó):
C: érdekli-e az akció?

P(X=fiatal | C=igen) = 3/5
P(X=idős | C=igen) = 0
P(X=középk. | C=igen) =2/5

P(X=fiatal | C=nem) = 0
P(X=idős | C=nem) = 4/5
P(X=középk. | C=nem) = 1/5

P(Y=alacsony | C=igen) = 1/5
P(Y=közepes | C=igen) = 1/5
P(Y=magas | C=igen) =3/5

P(Z=igen | C=igen) = 2/5
P(Z=nem | C=igen) = 3/5

P(Z=igen | C=nem) = 2/5
P(Z=nem | C=nem) = 3/5

P(C=igen) = 5/10
P(C=nem) = 5/10

Alkalmazás

Új (ismeretlen osztályba tartozó) példány osztályozása

Sorsz.	X	Y	Z	C
11	fiatal	közepes	igen	?

$$P(C=igen \mid X=fiatal, Y=közepes, Z=igen) =$$

$$= \frac{P(C=igen)*P(X=fiatal|C=igen)*P(Y=közepes|C=igen)*P(Z=igen|C=igen)}{P(X=fiatal, Y=közepes, Z=igen)} =$$

$$= \frac{5/10 * 3/5 * 1/5 * 2/5}{P(X=fiatal, Y=közepes, Z=igen)} = \frac{0{,}024}{P(X=fiatal, Y=közepes, Z=igen)}$$

Hasonlóképpen: $P(C=nem \mid X=fiatal, Y=közepes, Z=igen) =$

$$= \frac{5/10 * 0 * 3/5 * 2/5}{P(X=fiatal, Y=közepes, Z=igen)} = \frac{0}{P(X=fiatal, Y=közepes, Z=igen)}$$

Bayes hihetőségi hálók (Bayesian networks)

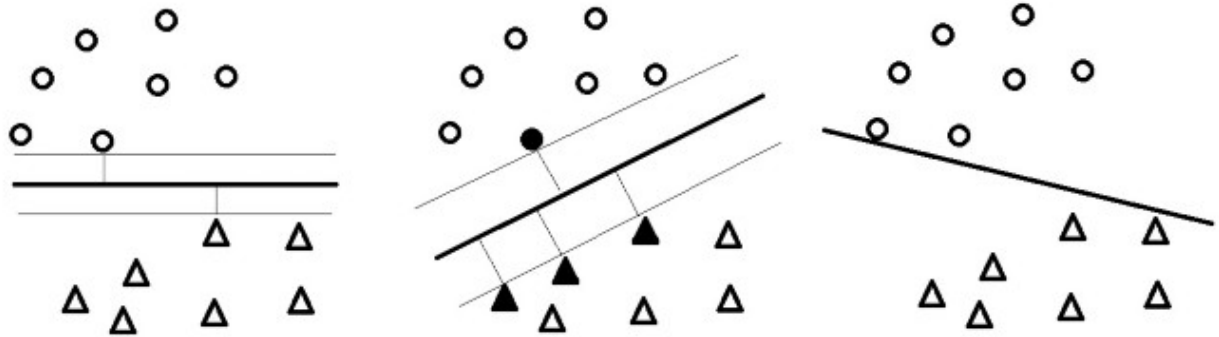
- Függetlenségre tett feltételeket enyhítik
- A változók közötti függőségeket egy DAG-gal lehet megadni
- A függőségeket leírja a következő képlet:

$$\mathbb{P}((Z_1, Z_2, \ldots, Z_s) = (z_1, z_2, \ldots, z_s)) = \prod_{j=1}^{s} \mathbb{P}(Z_j = z_j | par(Z_j))$$

- A $par(Z_j)$ a Z_j csúcs szüleit jelenti (hisz több is lehet)
- Tetszőleges csúcsokat kijelölhetünk előrejelzendőnek
- Ha nincsenek rejtett változók, akkor a $\mathbb{P}(Z_j = z_j | par(Z_j))$ valószínűségek becsülhetők a mintákból
- Rejtett változók esetén az ún. gradiens módszer alkalmazható
- Valószínűségi modellek megtervezésekor a Bayes hálók topológiáját kialakítani

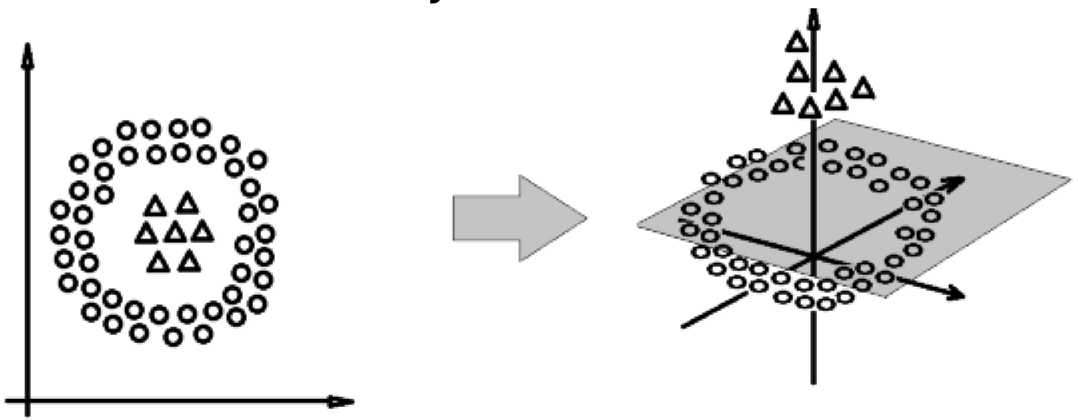
Szupport Vektor Gépek

- Bináris osztályozási feladatra használhatók
- Az attribútumoknak szám típusúnak kell lennie, hogy az objektumokat térbeli vektorokként ábrázolhassuk
- Elválasztó hipersíkot keresnek a két osztály pontjai között
  - Az elválasztó hipersíkhoz közeli pontok távolsága a lehető legnagyobb legyen
  - **Maximal marg**
- A hipersík dime
  objektumokénál
- Szupport vektor



SVM-ek

- Mi van, ha nincs tökéletes 1 dimenziós hipersík?
  - Kis hibát megengedhetünk, vagyis néhány pont a rossz oldalra kerülhet
  - Magasabb dimenziójú térben találhatunk megfelelő síkot
- A C (*complexity*) paraméterrel állíthatjuk az SVM-ek hibák súlyát
- Magasabb dimenzióba vetítés:



SVM-ek

- Nem feltétlenül kell egyenesen magasabb dimenzióba vetítenünk, elég ha ki tudjuk számolni a magasabb dimenziós távolságukat a pontoknak
- Ehhez egy megfelelően távolságfüggvény, úgynevezett $kernel$ szükséges
- (x_1, x_2) vetítése: $(x_1, x_1 * x_2)$
- Ekkor a magasabb dimenzióban lévő távolság egy alacsonyabb függvénye
- Kernel trükknek hívják a vetítés nélküli távolság számítást