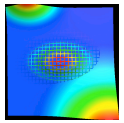
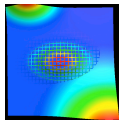


Nagy adathalmazok labor

2018-2019 őszi félév

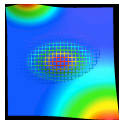
2018.09.19

1. Döntési fák
2. Naive Bayes
3. Logisztikus regresszió



Féléves terv

- Kiértékelés: cross-validation, bias-variance trade-off
- Supervised learning (discr. és generative modellek): nearest neighbour, decision tree (döntési fák), logisztikus regresszió, nem lineáris osztályozás, neurális hálózatok, support vector machine, idős osztályozás és dynamic time warping
- Lineáris, polinomiális és sokdimenziós regresszió és optimalizálás
- Tree ensembles: AdaBoost, GBM, Random Forest
- Clustering (klaszterezés): k-means, sűrűség és hierarchikus modellek
- Principal component analysis (PCA, SVD), collaborative filtering, ajánlórendszerek
- Anomália keresés
- Gyakori termékhalmazok
- Szublineáris algoritmusok: streaming, Johnson-Lindenstrauss, count-min sketch



Következő hetek

Szeptember 12: bevezetés, osztályozás-klaszterezés, kiértékelés, kNN, DT

Szeptember 13: gyakorlat, ipython notebook (akinek van kérem hozzon laptopot), NN

Szeptember 19: DT, Naive Bayes, logisztikus regresszió

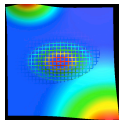
Szeptember 26: logisztikus regresszió, generalizáció

Szeptember 27: gyakorlat, ipython, LR

Október 3: polinomiális szeparálás, SVM

Október 10: ANN, MLP, CNN

Október 11: gyakorlat



NagyHF: Ajánlórendszer

Adat: moviedata.zip
- tanulóhalmaz:
  - ratings.train
- teszthalmaz:
  - ratings.test

Felhasználó/ film	Napoleon Dynamite	Szörny RT.	Cindarella	Life on Earth
Dávid	1	?	?	3
Dóri	5	3	5	5
Péter	?	4	3	?

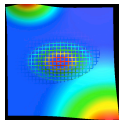
Célunk a ratings.test ismeretlen (“?”) értékeléseit megjósolni a tanulóhalmazon tanított modellünk segítségével!

Két módszer :
1. k Nearest Neighbour:
   - távolság? -> CF vagy CB
2. Mátrix faktorizáció (SGD):
   - faktorok száma
   - centralizáció

+1 Wikipedia-IMDB

+1 kombináció: jóslatok súlyozott összege?

Beküldendő: kitöltött ratings.test (valós, nem egész szám!)



NagyHF: Web Quality

Adat: webq.zip

tanulóhalmaz elemei

train_annotation

teszthalmaz:

test_annotation

Ötféle annotáció:

- megjelenés minősége (presentation)
- megbízhatóság (credibility)
- teljesség (completeness)
- ismeret mélysége (knowledge)
- koncepció (intention)

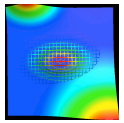
Többféle adathalmaz:

- szöveges tartalom
- numerikus feature-ök
- hálózati (link) statisztikák

Jósoljuk meg, hogy az adott teszt oldalak minősége megfelelő-e!
(az eredeti adatban 1-5 skála)

Módszerek:

- szöveges tartalom alapú osztályozás
- link és content alapú osztályozás
- fentiek kombinációja



NagyHF: Képosztályozás

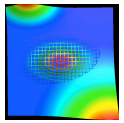
Autómatikus fogalom detekció és lokalizáció

Nem csak vizuális tartalom:

- Exif meta:camera, date gps
- Flickr tag
- kontextus

Objektum vs. fogalom osztályozás

Nappal, hajók, tó (tenger?), emberek, hegyek



NagyHF: Képosztályozás

Többféle adathalmaz (külön):

Pascal VOC:
- tanulóhalmaz: 5011
- teszthalmaz: 4952
- kategóriák: 20
- pl. kutya, birka, autó stb.

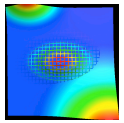
MirFlickr Yahoo!
- tanulóhalmaz: 15000
- teszthalmaz: 10000
- kategóriák: 94
- Objektumok és fogalmak!

Két megoldás:

- BoF: előre kiszámolt alacsony szintű leírók
- CNN: Convolutional Neural Network

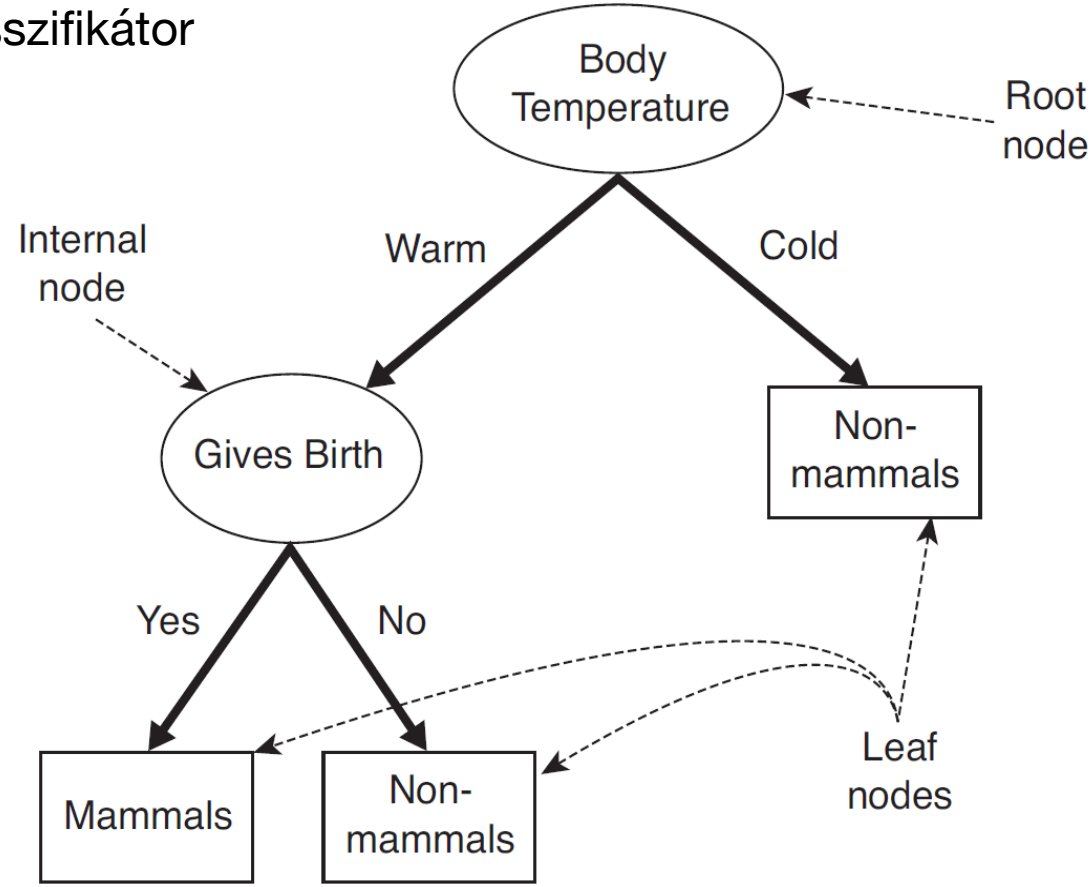
alapú osztályozás a nyers képek alapján
(ajánlott CNN implementációk: TensorFlow vagy DATO)

Nehézségre hasonlóak. (mindkét esetben külső lib-ek)

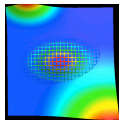


Döntési fák

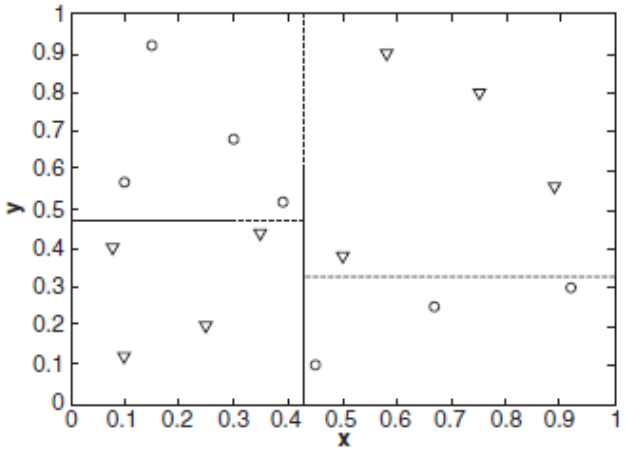
Pl. emlős klasszifikátor

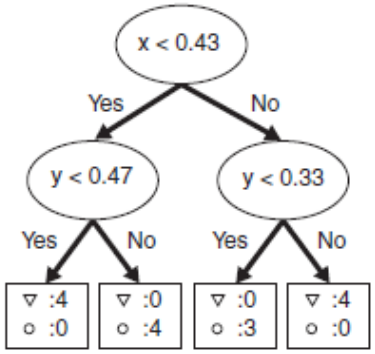


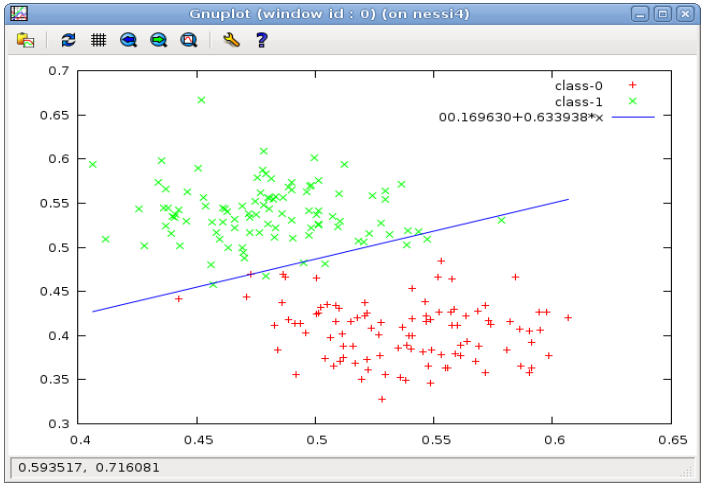
Ábra: TSK



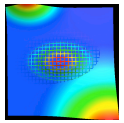
Döntési fák





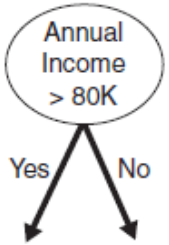


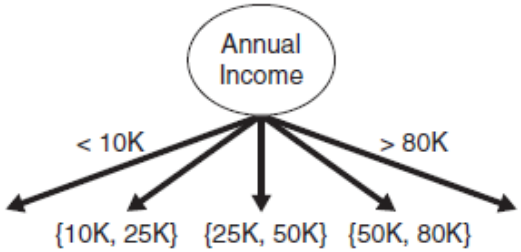
Megfelelő döntési fa?



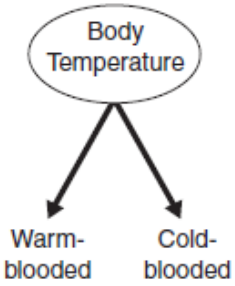
Döntési fák

Vágás attribútumok típusa szerint

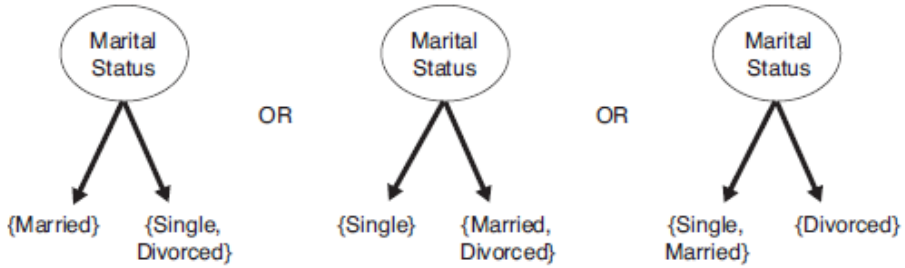




skála szerint

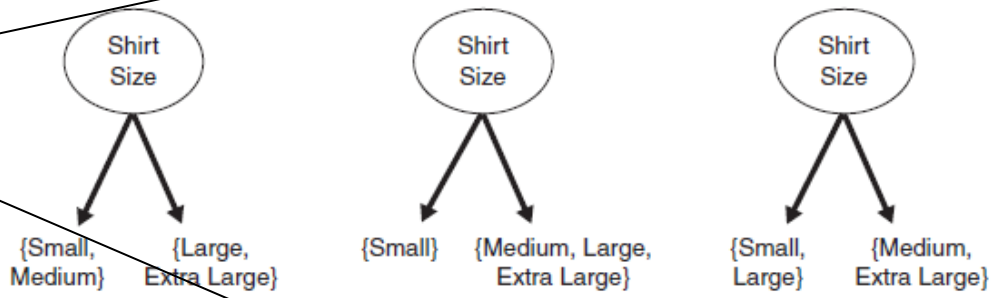


bináris

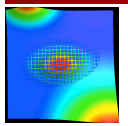


nominális

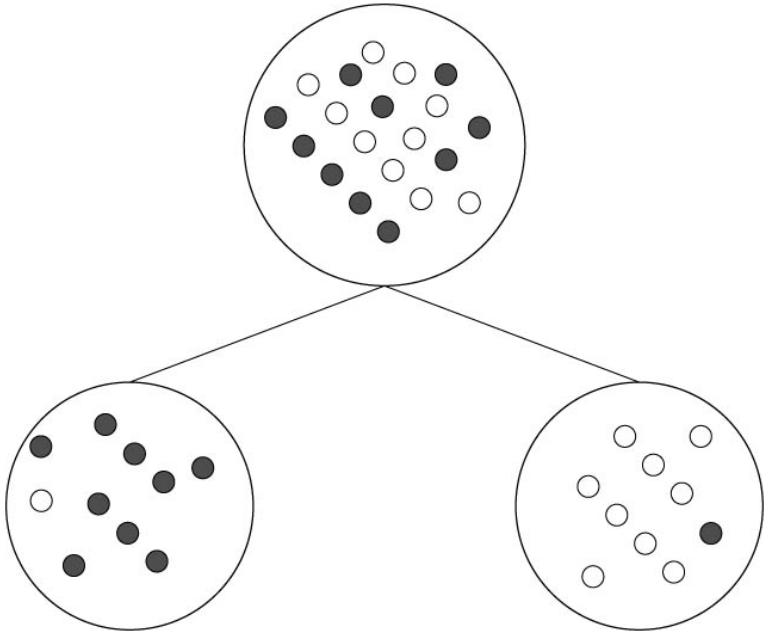
Különbség?



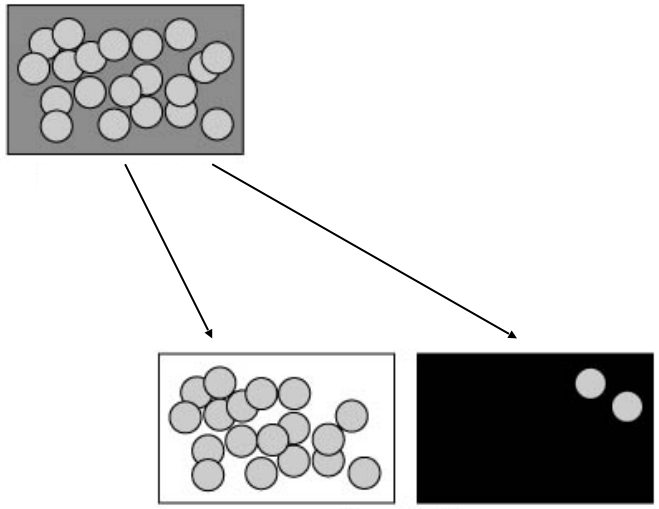
ordinális



Döntési fák

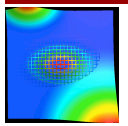


Jó vágás



Rossz vágás

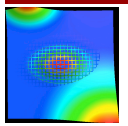
Ábra: TSK



Döntési fa építés (C4.5)

Feltétel: minden attribútum nominális

```
Procedure TreeBuilding (D)
        If (the data is not classified correctly)
                Find best splitting attribute A
                        For each a element in A
                        Create child node N_a
                                D_a all instances in D where A=a
                                TreeBuilding (D_a)
                        Endfor
        Endif
EndProcedure
```



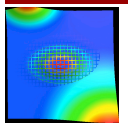
Döntési fák

Jó vágási attribútum:

- jól szeparálja az egyes osztályokhoz tartozó elemeket külön ágakra
  → növeli a tisztaságot (purity)
- kiegyensúlyozott

Purity mértékek lehetnek:

- klasszifikációs hiba
- entrópia
- gini
- vagy ami épp szükséges az adathoz



Döntési fák

Klasszifikációs hiba:

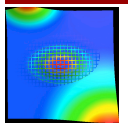
p(i,t) : egy adott t csomópontban az i osztályba tartozó elemek aránya

Classification error: 1-max(p(i,t))

Növekmény(gain):

$$\Delta = I(\text{parent}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j)$$

Itt I(parent) az adott csomópont tisztasága, k a vágás után keletkező ágak száma, $N(v_j)$ az adott ágon található elemek száma, N a csomópontban található elemek száma, $I(v_j)$ pedig a j-dik ág tisztasága

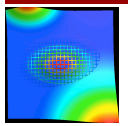


Döntési fák

Példa:

A vagy B attribútum szerint érdemes vágni, ha a tisztasági mérték a klasszifikációs hiba?

A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	−
T	T	+
F	F	−
F	F	−
F	F	−
T	T	−
T	F	−



Döntési fák

Példa:

A vagy B attribútum szerint érdemes vágni, ha a tisztasági mérték a klasszifikációs hiba?

A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	−
T	T	+
F	F	−
F	F	−
F	F	−
T	T	−
T	F	−

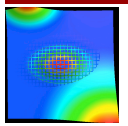
Vágás A alapján:

MCE = 0+3/7

Vágás B alapján:

MCE = 1/4+1/6

Vágjunk B szerint!



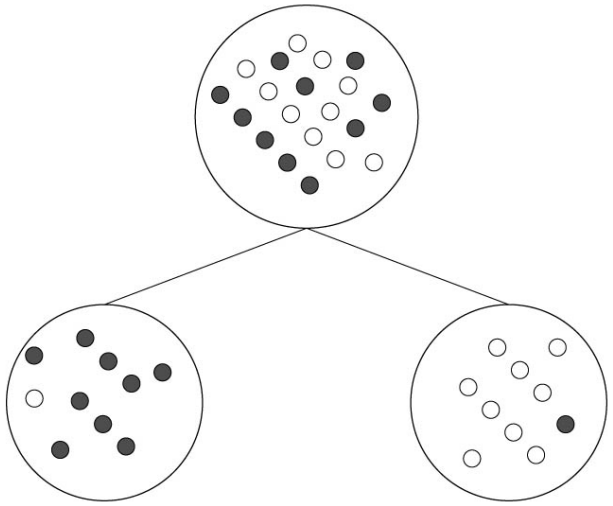
Döntési fák

Gini (population diversity)

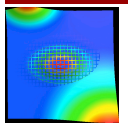
p(i|t) : egy adott t csomópontban az i osztályba tartozó elemek aránya

Gini:

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2$$



Gini a gyökérben?
Gini a leveleknél?



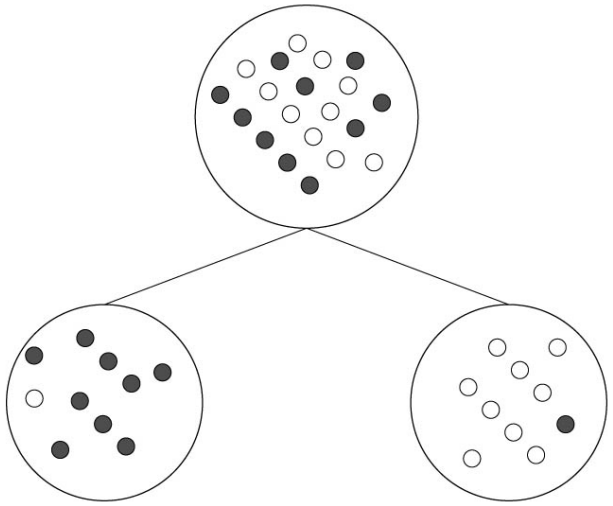
Döntési fák

Gini (population diversity)

p(i|t) : egy adott t csomópontban az i osztályba tartozó elemek aránya

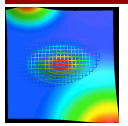
Gini:

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2$$



Gini(gyökér)= 0.5 = 0.5^2 + 0.5^2

Gini(level)= 0.82 = 0.1^2 + 0.9^2



Döntési fák

Entrópia (információ)

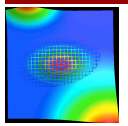
p(i|t) : egy adott t csomópontban az i osztályba tartozó elemek aránya

Entrópia:

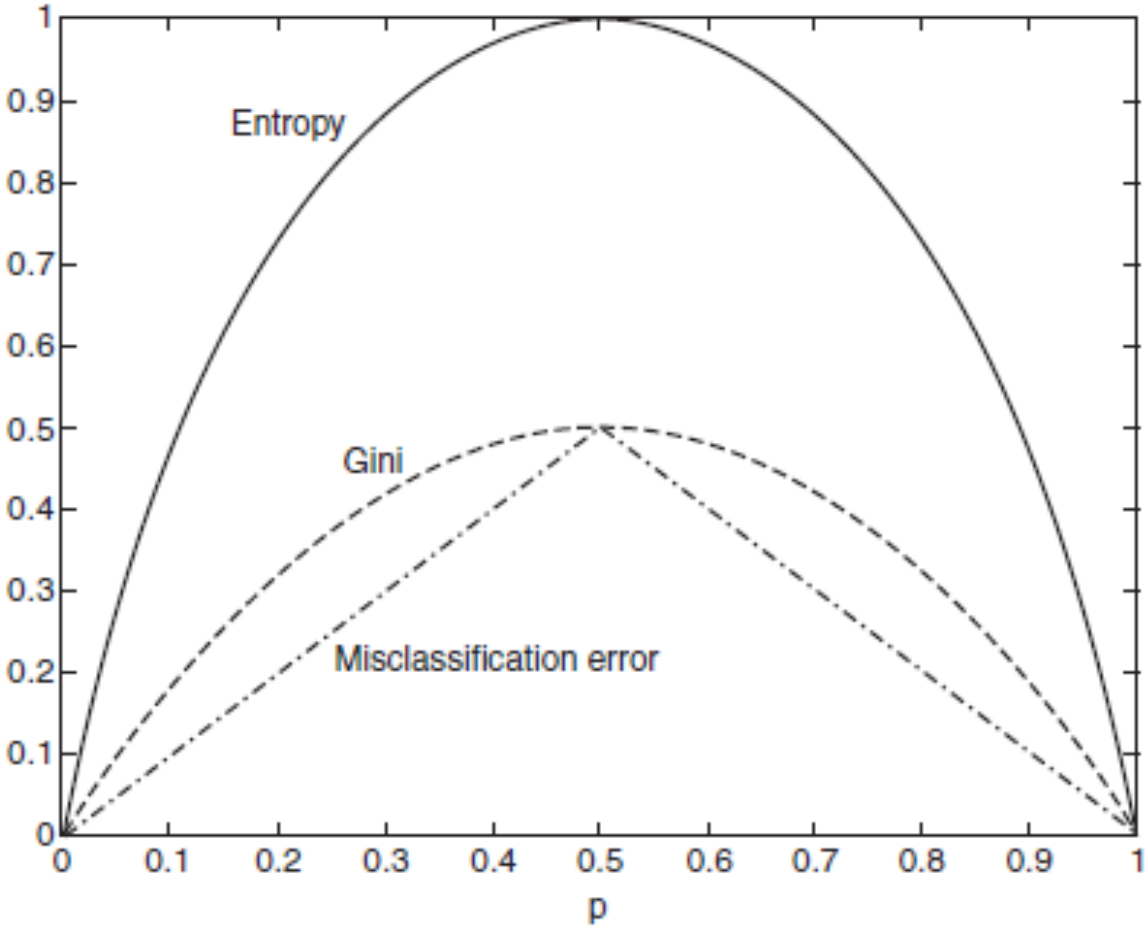
$$I(v_j) - \sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t)$$

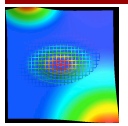
Mindegyik mérték közös tulajdonsága, hogy maximumukat 0.5-nél érik el. Mind a több részre szabdalást preferálja. (nem bináris attribútumoknál)

$$\text{Gain ratio} = \frac{\Delta_{\text{info}}}{-\sum_{i=1}^{k} P(v_i) \log_2 P(v_i)}$$



Döntési fák


Entropy
Gini
Misclassification error
p
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



Döntési fák

$$\text{Entropy}(t) = -\sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t),$$

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2,$$

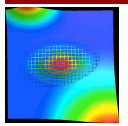
$$\text{Classification error}(t) = 1 - \max_i [p(i|t)],$$

Node N_1	Count
Class=0	0
Class=1	6

Node N_2	Count
Class=0	1
Class=1	5

Node N_3	Count
Class=0	3
Class=1	3

Mennyi az entrópia, klasszifikációs hiba és a gini?



Döntési fák

$$\text{Entropy}(t) = -\sum_{i=0}^{c-1} p(i|t)\log_2 p(i|t),$$

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1}[p(i|t)]^2,$$

$$\text{Classification error}(t) = 1 - \max_i[p(i|t)],$$

Node N_1	Count
Class=0	0
Class=1	6

$\text{Gini} = 1 - (0/6)^2 - (6/6)^2 = 0$
$\text{Entropy} = -(0/6)\log_2(0/6) - (6/6)\log_2(6/6) = 0$
$\text{Error} = 1 - \max[0/6, 6/6] = 0$

Node N_2	Count
Class=0	1
Class=1	5

$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$
$\text{Entropy} = -(1/6)\log_2(1/6) - (5/6)\log_2(5/6) = 0.650$
$\text{Error} = 1 - \max[1/6, 5/6] = 0.167$

Node N_3	Count
Class=0	3
Class=1	3

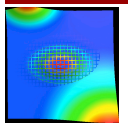
$\text{Gini} = 1 - (3/6)^2 - (3/6)^2 = 0.5$
$\text{Entropy} = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1$
$\text{Error} = 1 - \max[3/6, 3/6] = 0.5$

Pár direktíva:

- ha két fa hasonlóan teljesít mindig válasszuk a kevésbé komplexet
- csak adott threshold feletti javulás esetén vágjunk (early pruning)
- utólag összevonjuk azon leveleket, melyek a legkevesebb hibát okozzák, s a leggyakoribb osztályra döntünk (post-pruning)
- MDL(Minimum Description Length): azt a fát válasszuk, melynek a leírása kisebb

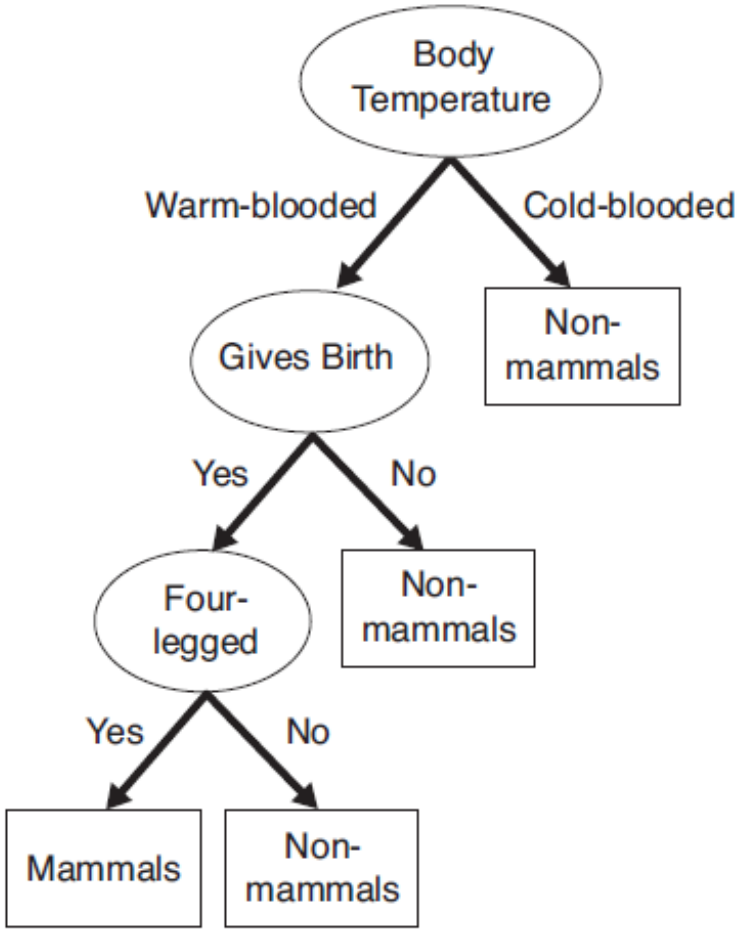
Példa hibára: mi lesz egy delfinnel? (tesztadat)

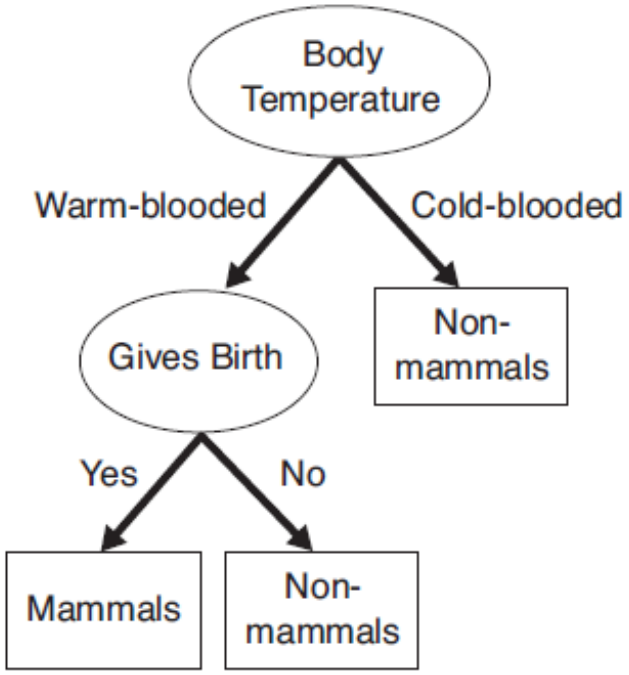
Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
porcupine	warm-blooded	yes	yes	yes	yes
cat	warm-blooded	yes	yes	no	yes
bat	warm-blooded	yes	no	yes	no*
whale	warm-blooded	yes	no	no	no*
salamander	cold-blooded	no	yes	yes	no
komodo dragon	cold-blooded	no	yes	no	no
python	cold-blooded	no	no	yes	no
salmon	cold-blooded	no	no	no	no
eagle	warm-blooded	no	no	no	no
guppy	cold-blooded	yes	no	no	no

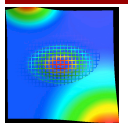


Döntési fák

Zaj?







Döntési fák

Megjegyzések:

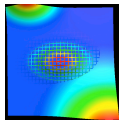
DT-k képesek kezelni mind nominális, mind pedig numerikus adatokat
(dátumok és string-ek?)
könnyen értelmezhetőek
Zajra robusztusak (?)
De a C4.5 algoritmus előállíthatja ugyanazt az alfát ("subtree") többször
Túltanulás ("overfitting")

Miért?

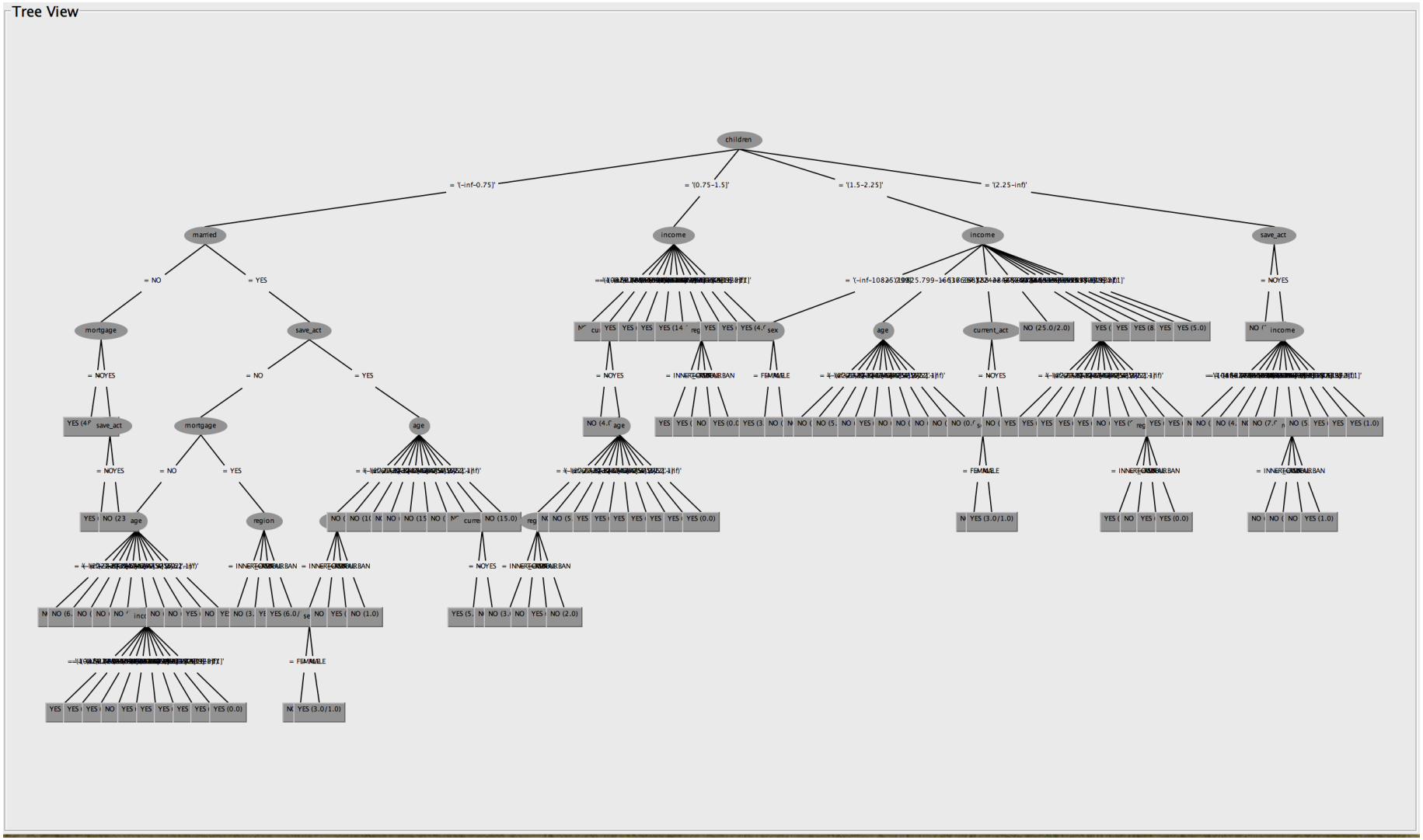
Jellemző okok:

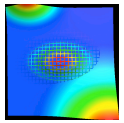
- túl mély és széles a DT kevés tanuló elemmel az egyes levelekben
- kiegyensúlyozatlan ("unbalanced") tanuló halmaz

Megoldás: "pruning"!



Unpruned tree





Subtree raising vs. replacement (rep)

Decision Tree:

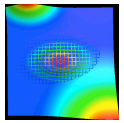
```
depth = 1:
|  breadth> 7 :  class 1
|  breadth<= 7:
|  |  breadth <= 3:
|  |  |  ImagePages> 0.375:  class 0
|  |  |  ImagePages<= 0.375:
|  |  |  |  totalPages<= 6:  class 1
|  |  |  |  totalPages> 6:
|  |  |  |  |  breadth <= 1:  class 1
|  |  |  |  |  breadth > 1:  class 0
|  |  width > 3:
|  |  |  MultiIP = 0:
|  |  |  |  ImagePages<= 0.1333:  class 1
|  |  |  |  ImagePages> 0.1333:
|  |  |  |  breadth <= 6:  class 0
|  |  |  |  breadth > 6:  class 1
|  |  |  MultiIP = 1:
|  |  |  |  TotalTime <= 361:  class 0
|  |  |  |  TotalTime > 361:  class 1
depth> 1:
|  MultiAgent = 0:
|  |  depth > 2:  class 0
|  |  depth <= 2:
|  |  |  MultiIP = 1:  class 0
|  |  |  MultiIP = 0:
|  |  |  |  breadth <= 6:  class 0
|  |  |  |  breadth > 6:
|  |  |  |  |  RepeatedAccess <= 0.322:  class 0
|  |  |  |  |  RepeatedAccess > 0.322:  class 1
|  MultiAgent = 1:
|  |  totalPages <= 81:  class 0
|  |  totalPages > 81:  class 1
```

Subtree Raising

Subtree Replacement

Simplified Decision Tree:

```
depth = 1:
|  ImagePages <= 0.1333:  class 1
|  ImagePages > 0.1333:
|  |  breadth <= 6:  class 0
|  |  breadth > 6:  class 1
depth > 1:
|  MultiAgent = 0:  class 0
|  MultiAgent = 1:
|  |  totalPages <= 81:  class 0
|  |  totalPages > 81:  class 1
```



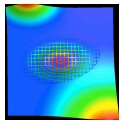
Subtree raising

Levelek: 20
Fa: 39

```
income <= 51284.3
|   children <= 1
|   |   children <= 0
|   |   |   married = NO
|   |   |   |   mortgage = NO: YES (43.0/3.0)
|   |   |   |   mortgage = YES
|   |   |   |   |   save_act = NO: YES (12.0)
|   |   |   |   |   save_act = YES: NO (22.0)
|   |   |   married = YES
|   |   |   |   save_act = NO
|   |   |   |   |   mortgage = NO
|   |   |   |   |   |   income <= 21506.2
|   |   |   |   |   |   |   age <= 41: NO (11.0/1.0)
|   |   |   |   |   |   |   age > 41: YES (5.0/1.0)
|   |   |   |   |   |   income > 21506.2: NO (20.0)
|   |   |   |   |   mortgage = YES: YES (25.0/3.0)
|   |   |   |   save_act = YES: NO (115.0/12.0)
|   |   children > 0
|   |   |   income <= 15538.8
|   |   |   |   age <= 41
|   |   |   |   |   income <= 12683.6: NO (14.0)
|   |   |   |   |   income > 12683.6
|   |   |   |   |   |   income <= 13381: YES (2.0)
|   |   |   |   |   |   income > 13381: NO (6.0)
|   |   |   |   age > 41: YES (2.0)
|   |   |   income > 15538.8: YES (101.0/5.0)
|   children > 1
|   |   income <= 30404.3: NO (124.0/12.0)
|   |   income > 30404.3
|   |   |   children <= 2: YES (36.0/5.0)
|   |   |   children > 2
|   |   |   |   income <= 44288.3: NO (19.0/2.0)
|   |   |   |   income > 44288.3: YES (6.0)
income > 51284.3
|   children <= 0
|   |   age <= 62: YES (4.0)
|   |   age > 62: NO (6.0/1.0)
|   children > 0: YES (27.0)
```

Levelek: 17
Fa: 33

```
children <= 1
|   children <= 0
|   |   married = NO
|   |   |   mortgage = NO: YES (48.0/3.0)
|   |   |   mortgage = YES
|   |   |   |   save_act = NO: YES (12.0)
|   |   |   |   save_act = YES: NO (23.0)
|   |   married = YES
|   |   |   save_act = NO
|   |   |   |   mortgage = NO
|   |   |   |   |   income <= 21506.2
|   |   |   |   |   |   age <= 41: NO (11.0/1.0)
|   |   |   |   |   |   age > 41: YES (5.0/1.0)
|   |   |   |   |   income > 21506.2: NO (20.0)
|   |   |   |   mortgage = YES: YES (25.0/3.0)
|   |   |   save_act = YES: NO (119.0/12.0)
|   children > 0
|   |   income <= 15538.8
|   |   |   age <= 41
|   |   |   |   income <= 12683.6: NO (14.0)
|   |   |   |   income > 12683.6
|   |   |   |   |   income <= 13381: YES (2.0)
|   |   |   |   |   income > 13381: NO (6.0)
|   |   |   age > 41: YES (2.0)
|   |   income > 15538.8: YES (111.0/5.0)
children > 1
|   income <= 30404.3: NO (124.0/12.0)
|   income > 30404.3
|   |   children <= 2: YES (51.0/5.0)
|   |   children > 2
|   |   |   income <= 44288.3: NO (19.0/2.0)
|   |   |   income > 44288.3: YES (8.0)
```



Pruning hatása

Subtree replacement vs. raising

Tanuló halmaz

ID	jármű	szín	gyorsulás
T1	motor	piros	nagy
T2	motor	kék	nagy
T3	autó	kék	nagy
T4	motor	kék	nagy
T5	autó	zöld	kicsi
T6	autó	kék	kicsi
T7	autó	kék	nagy
T8	autó	piros	kicsi

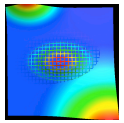
Validációs halmaz

ID	jármű	szín	gyorsulás
Valid 1	motor	piros	kicsi
Valid 2	motor	kék	nagy
Valid 3	autó	kék	nnagy
Valid 4	autó	kék	nagy

Test set

ID	jármű	szín	gyorsulás
Teszt 1	motor	piros	kicsi
Teszt 2	motor	zöld	kicsi
Teszt 3	autó	piros	kicsi
Teszt 4	autó	zöld	kicsi

Építsünk egy két szintű döntési fát
Csökkentsük a fa méretét (pruning) a validációs halmaz alapján
Hogyan változott a modell teljesítménye a teszt halmazon?



Súlyozott hiba és C4.5

A	B	C	“+”	“-”
I	I	I	5	0
H	I	I	0	20
I	H	I	20	0
H	H	I	0	5
I	I	H	0	0
H	I	H	25	0
I	H	H	0	0
H	H	H	0	25

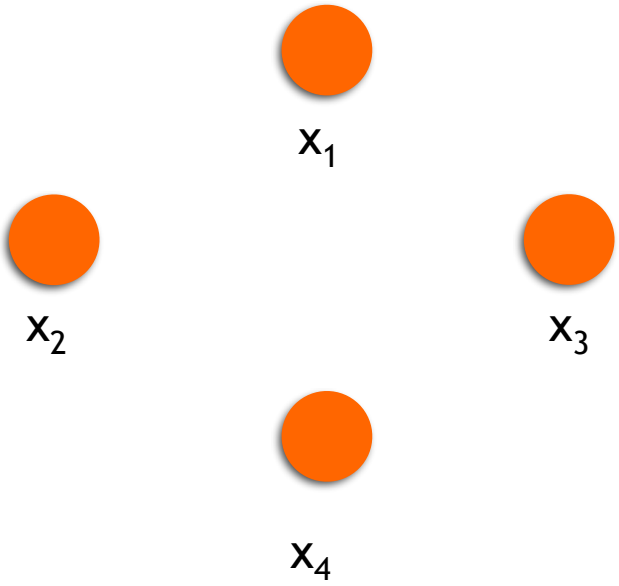
Építsünk egy két szintű döntési fát
Létezik ideális dönntési fa?
Az alábbi "cost" mátrix alapján változtassuk meg a predikciót:

Jósolt/"ground truth"	“+”	“-”
“+”	0	1
“-”	2	0

Bayes Networks

Mielőtt NN ☺️

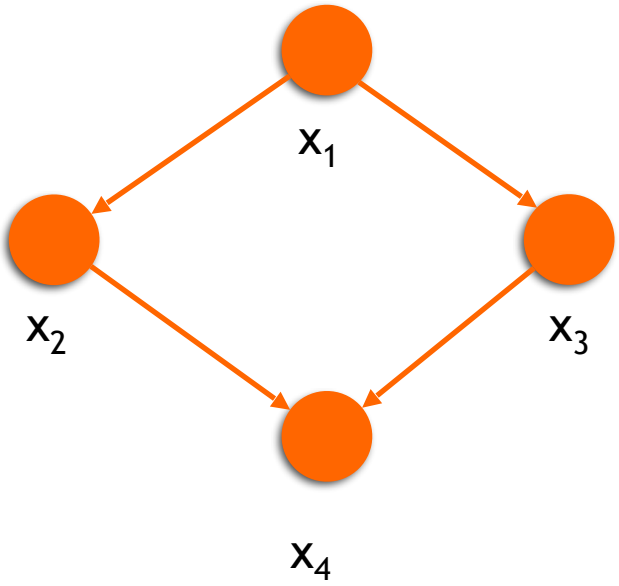
Probabilistic Graphical model -> Random Fields
Adott valószínűségi változók halmaza: $X= \{x_1,\ldots,x_T\}$
Egy él két változ között kapcsolatot jelez



Bayes Networks

Probabilistic Graphical model

Egy él két változ között kapcsolatot jelez (irányított)
Feltételes valószínűségek (A “causes” B, okozat) vs. Markov Random Fields?

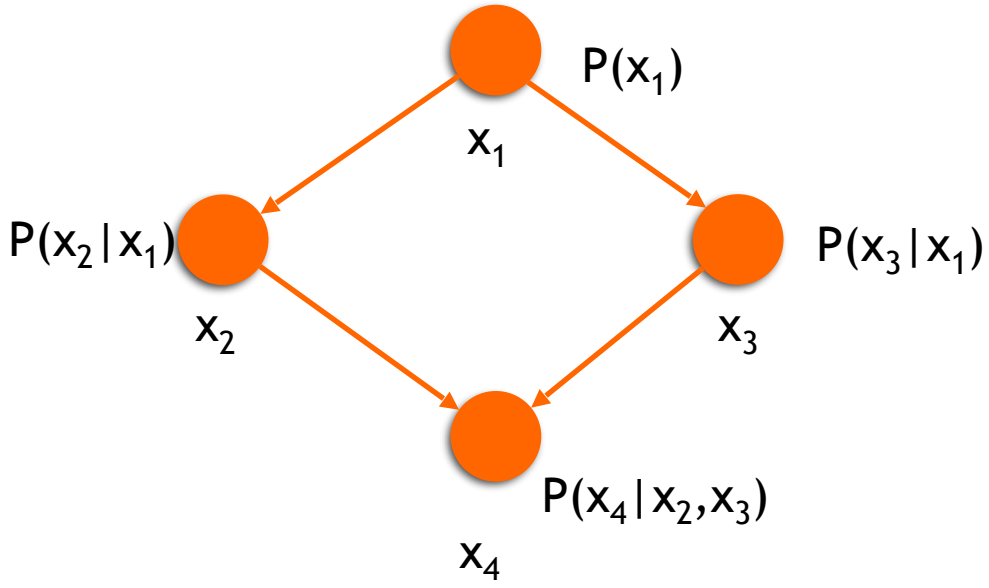


Bayes Networks

Probabilistic Graphical model

Adott valószínűségi változók halmaza: $X= \{x_1,\ldots,x_T\}$

Egy él két változ között kapcsolatot jelez (irányított)

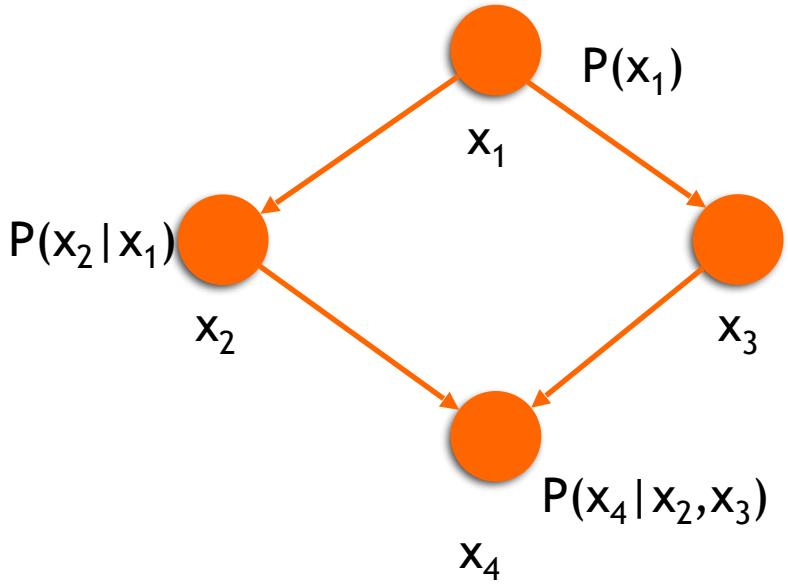


Bayes Networks

Probabilistic Graphical model

Adott valószínűségi változók halmaza: $X= \{x_1,\ldots,x_T\}$

Egy él két változ között kapcsolatot jelez (irányított)



$P(x_1)P(x_2|x_1)P(x_3|x_2,x_1)\ P(x_4|\ x_3,x_2,x_1)$
$= P(x_1)P(x_2|x_1)P(x_3|x_1)\ P(x_4|\ x_3,x_2)$

Bayes Networks

Tanulás:

I) A függőségeket ismerjük -> struktúra

Parameter estimation (prior, posterior)
Analitikus vagy ha nincs más optimalizáció (EM, GD stb.)

II) Nem ismerjük a függőségeket -> nem ismert a struktúra

az összes lehetséges struktúra (DAG) terében optimalizálunk
P(G | D) ahol G egy gráf és D maga az ismert adat

Mint látni fogjuk a legtöbb ismert modell egyértelműen az egyik csoportba sorolható

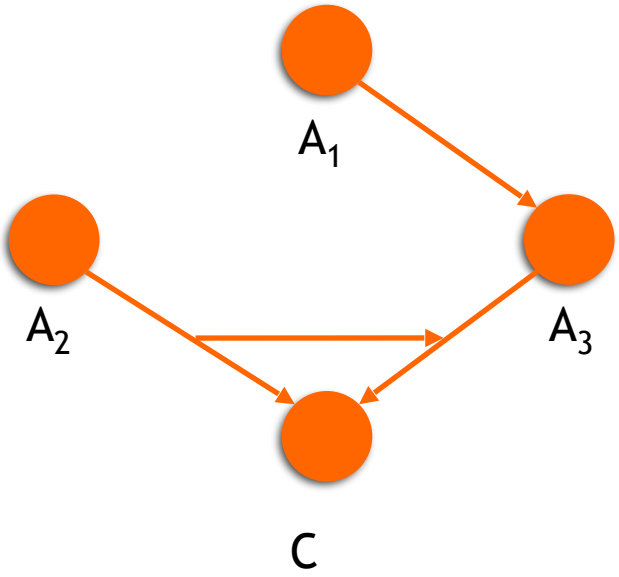
DT?

A legegyszerűbb Bayes Networks: Naïve-Bayes

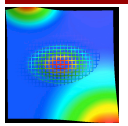
Inference:

$P(C=1 \mid \{A_i\})$ vs. $P(C=0 \mid \{A_i\})$

ahol $\{A_i\}$ az attribútumok halmaza, C az osztályok halmaza



Bayes tétel?



Naïve-Bayes

Bayes-i alapok:

$$p(h \mid d) = \frac{P(d \mid h)P(h)}{P(d)}$$

Priori: P(h)
Posteriori: P(h|d)
Evidence: P(d)
Total probability: P(h)=P(h,d)+P(h,/d)=P(h|d)P(d)+P(h|/d)P(/d)

Feladat: Az épületben megforduló emberek tizede tanár a többi diák. 10-ből 8 diák saját laptoppal rendelkezik míg minden 10 tanárból csak 5-nek van laptopja, mennyi a valószínűsége, hogy egy ember aki bejön az ajtón:

a) tanár
b) diák
c) van laptopja
d) van laptopja és diák
e) van laptopja és tanár

Naïve-Bayes

Szeretnénk attribútumok alapján meghatározni a posteriori valószínüségeket:

P(class=1| A1,A2, ...,An) és P(class=0| A1,A2,...,An)

Amelyik nagyobb arra döntünk. P(A1,A2,...,An | class=1) P(class=1)/P(A1,A2,...,An)

P(A1,A2,...,An | class=1)=πP(Ai|class=1)

konstans!

Saját háza van	Családi állapot	kereset	Van hitele?
van	egyedülálló	125e	nincs
nincs	házas	100e	nincs
nincs	egyedülálló	70e	nincs
van	házas	120e	nincs
nincs	elvált	95e	van
nincs	házas	60e	nincs
van	elvált	220e	nincs
nincs	egyedülálló	85e	van
nincs	házas	75e	nincs
nincs	egyedülálló	90e	van

Naïve-Bayes

Szeretnénk attribútumok alapján meghatározni a posteriori valószínüségeket:

P(class=1| A1,A2, ...,An) és P(class=0| A1,A2,...,An)

Amelyik nagyobb arra döntünk. P(A1,A2,...,An | class=1) P(class=1)/P(A1,A2,...,An)

P(A1,A2,...,An | class=1)=πP(Ai|class=1)

konstans!

Saját háza van	Családi állapot	kereset	Van hitele?
van	egyedülálló	125e	nincs
nincs	házas	100e	nincs
nincs	egyedülálló	70e	nincs
van	házas	120e	nincs
nincs	elvált	95e	van
nincs	házas	60e	nincs
van	elvált	220e	nincs
nincs	egyedülálló	85e	van
nincs	házas	75e	nincs
nincs	egyedülálló	90e	van

P(saját ház=nincs | hitel=nincs)=4/7
P(saját ház=van | hitel=nincs)=3/7
P(saját ház=nincs | hitel=van)=1
P(saját ház=van | hitel=van)=0
P(családi_áll=egyedülálló | hitel=nincs)=2/7
P(családi_áll=házas | hitel=nincs)=4/7
P(családi_áll=elvált | hitel=nincs)=1/7
P(családi_áll=egyedülálló | hitel=van)=2/3
P(családi_áll=házas | hitel=van)=0
P(családi_áll=elvált | hitel=van)=1/3

P(kereset=? | hitel=van)=?
P(kereset=? | hitel=nincs)=?

Naïve-Bayes

Folytonos változók esetében

- diszkretizáljuk az adatatot : 0-90e Ft-ig 91-125e stb
- modellezzük a problémát normál eloszlással!

$$P(X_i = x_i | Y = y_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}}$$

Saját háza van	Családi állapot	kereset	Van hitele?
van	egyedülálló	125e	nincs
nincs	házas	100e	nincs
nincs	egyedülálló	70e	nincs
van	házas	120e	nincs
nincs	elvált	95e	van
nincs	házas	60e	nincs
van	elvált	220e	nincs
nincs	egyedülálló	85e	van
nincs	házas	75e	nincs
nincs	egyedülálló	90e	van

Kereset:

hitel=nincs:

Átlag: 110e
Szórás^2: 2975

hitel=van:

Átlag: 90e
Szórás^2: 25

Naïve-Bayes

Szeretnénk attribútumok alapján meghatározni a posteriori valószínüségeket:

P(class=1| A1,A2, ...,An) és P(class=0| A1,A2,...,An)

Amelyik nagyobb arra döntünk. P(A1,A2,...,An class=1)P(class=1)/P(A1,A2,...,An)

P(A1,A2,...,An | class=1)=πP(Ai|class=1)

konstans!

Saját ház	Családi állapot	kereset	Van hitele?
van	egyedülálló	125e	nincs
nincs	házas	100e	nincs
nincs	egyedülálló	70e	nincs
van	házas	120e	nincs
nincs	elvált	95e	van
nincs	házas	60e	nincs
van	elvált	220e	nincs
nincs	egyedülálló	85e	van
nincs	házas	75e	nincs
nincs	egyedülálló	90e	van

Van-e hitele a következő tulajdonságokkal rendelkező személynek:

- nincs saját háza
- házas
- 120e a keresete

P(hitel=van | saját ház=nincs, családi_áll=házas, kereset=120e)=?
P(hitel=van | saját ház=nincs, családi_áll=házas, kereset=120e)=?

Naïve-Bayes

A 0-a valószínűségek több esetben zajként jelentkeznek (pl. nincs rá elem az eredeti adathalmazban ergo nem is lehetséges)

M-estimate:

Legyen p egy előre meghatározott minimum valószínűség

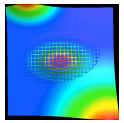
Módosítsuk a feltételes valószínűségek kiszámítását:

$$P(x_i|y_j) = \frac{n_c + mp}{n + m}$$

Ahol m egy előre meghatározott konstans, n_c azon x_i tulajdonsággal rendelkező tanulópontok száma melyek osztályváltozója y_i , n pedig az összes ide tartozó tanulópont száma.

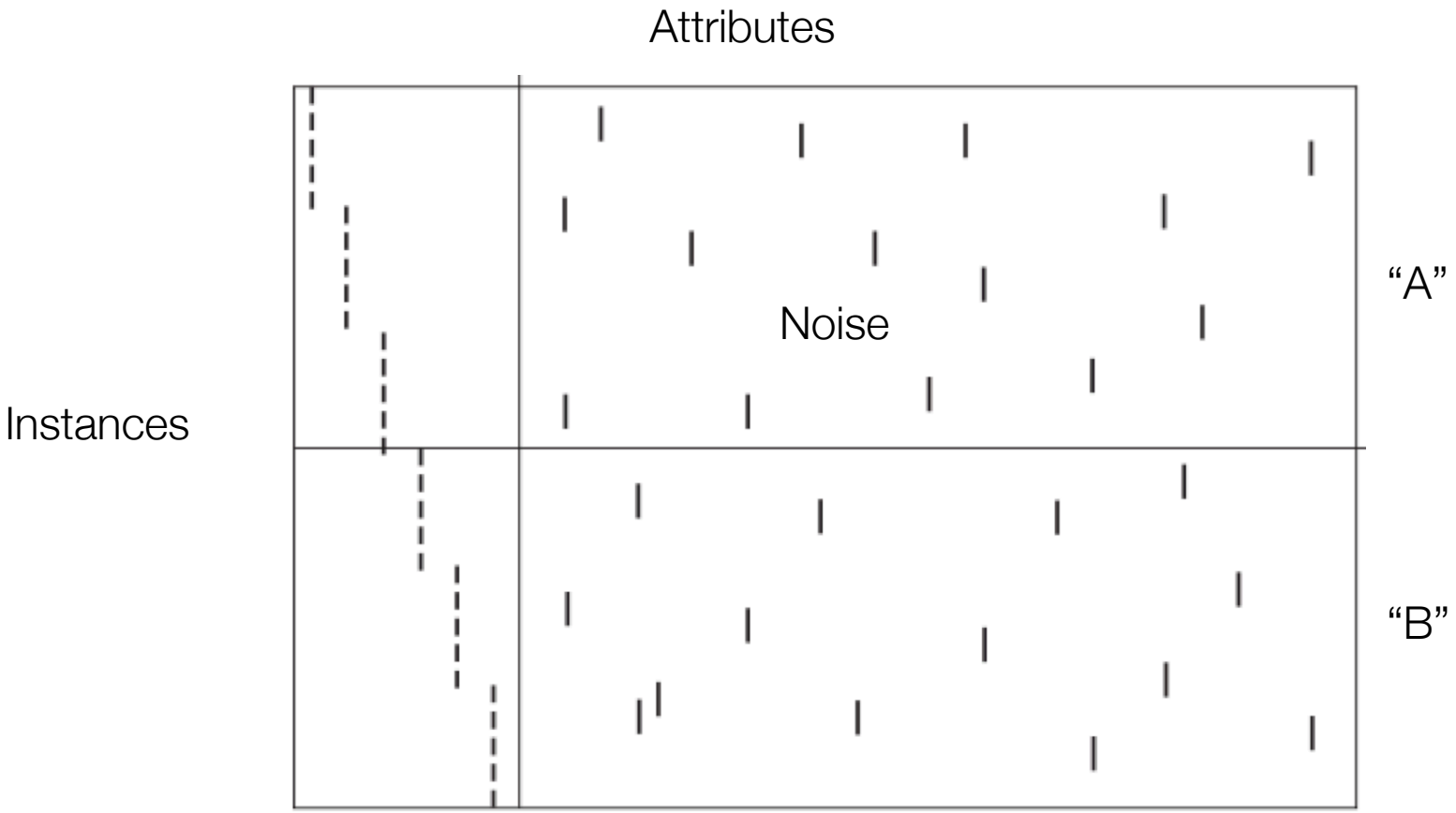
Olyan esetekben is p lesz a posterior valószínűség, ha egyáltalán nincs olyan tanulópont melynek y_i az osztályváltozója.

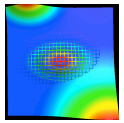
M-estimate nagyban segíti a zajos, hiányos vagy egyszerűen speciális esetek korrekt kiértékelését anélkül, hogy azok szerepeltek volna az tanulóhalmazban.



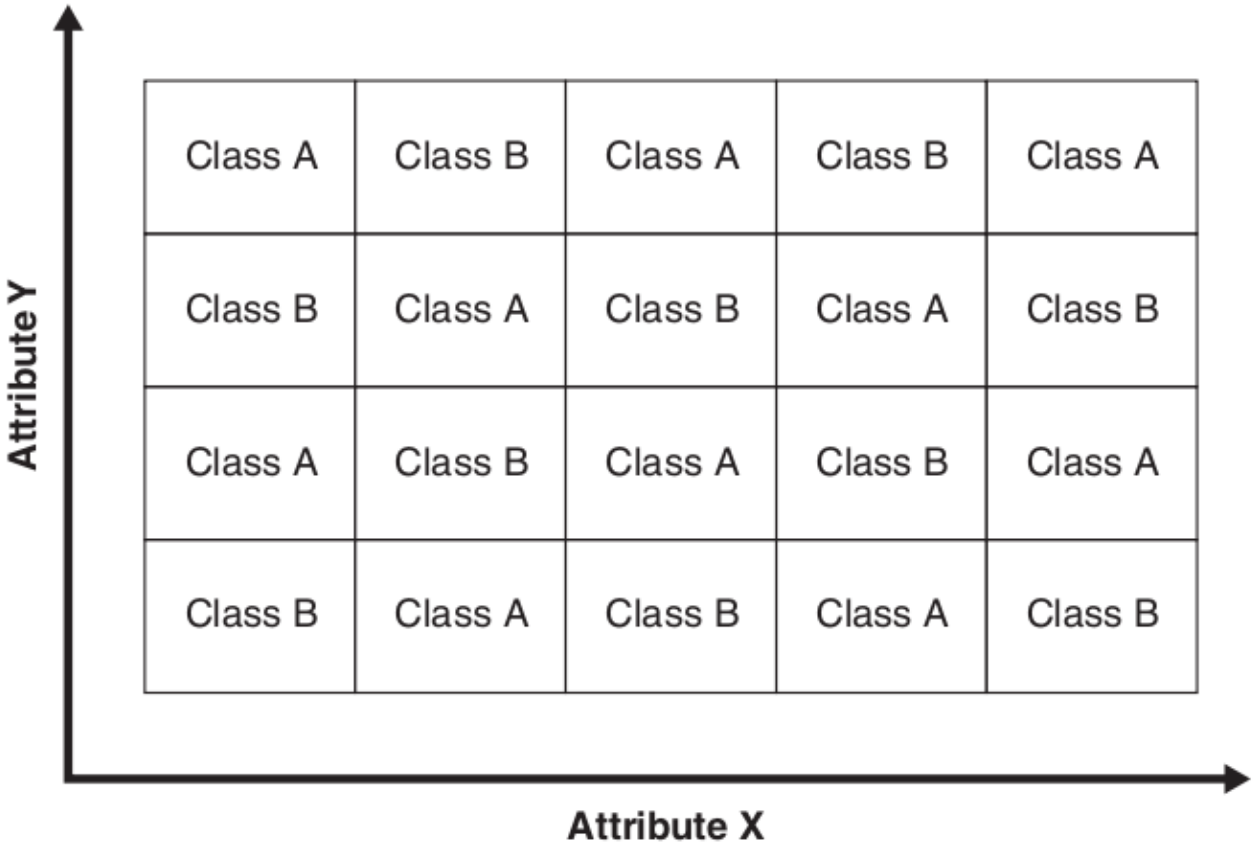
Zajos attribútumok

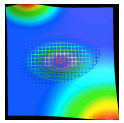
Hogyan teljesít a kNN, DT vagy BN?



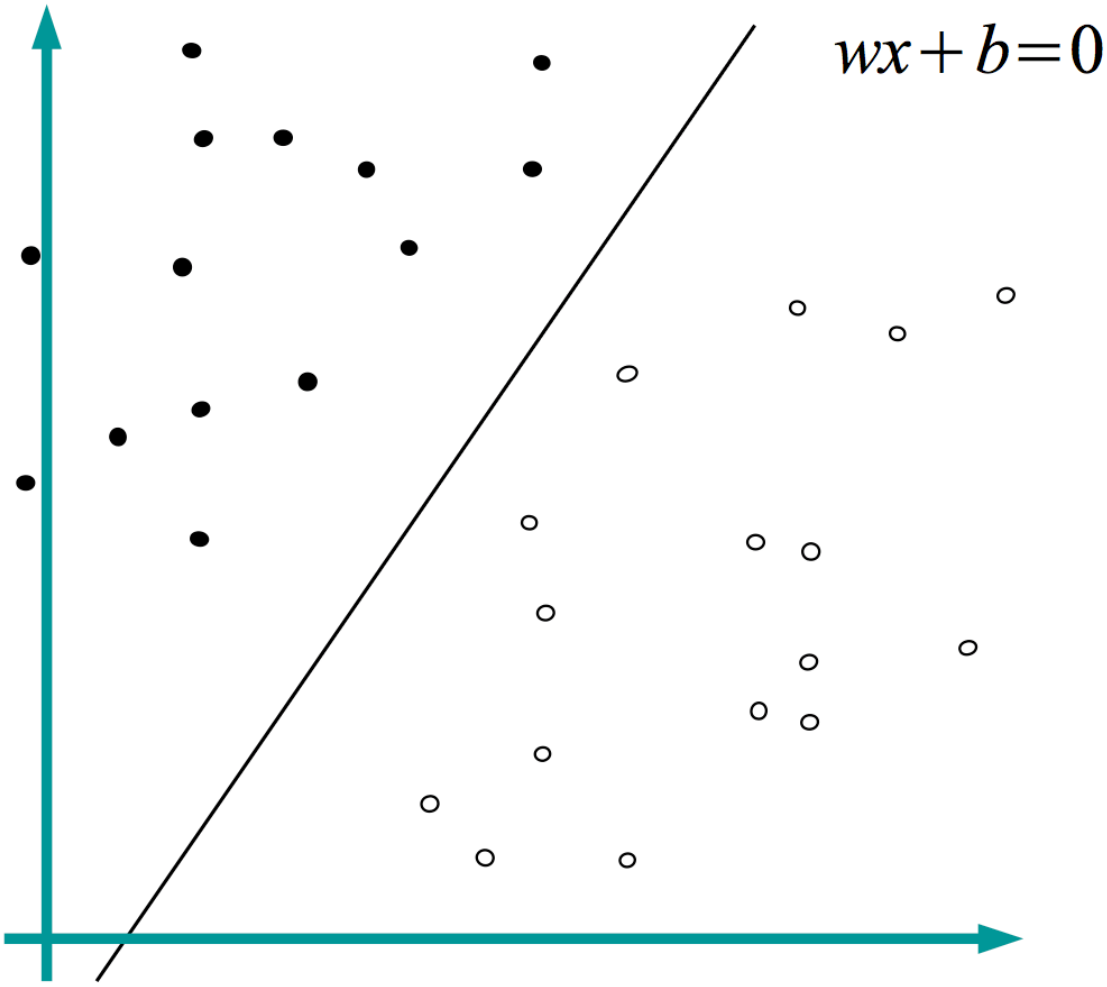


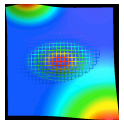
S itt? ☺️
(K-NN, Naïve-Bayes, DT)





Lineáris szeparálás



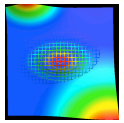


Lineáris szeparálás

Legyen adott egy véges n elemű mintahalmaz $\mathbf{X}=\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ egy d-dimenziós térben illetve minden elemhez rendeljünk egy osztályváltozót ($\mathbf{y}_i$). Keressünk egy d-dimenziós **w** normál vektort melyre a következő egyenlőtlenségek igazak

$$\mathbf{w}\cdot\mathbf{x}_i > b \quad \text{minden } \mathbf{x}_i \text{ elemre melynek a cimkéje } +1$$
$$\mathbf{w}\cdot\mathbf{x}_i < b \quad \text{minden } \mathbf{x}_i \text{ elemre melynek a cimkéje } -1$$

Azon vektor-küszöb párosok (**w**, b), melyekre igazak a fenti egyenlőtlenségek **X,y** lineáris szeparátorai.



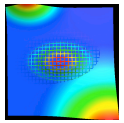
Lineáris szeparálás

Egészítsük ki a meglévő vektorainkat egy extra dimenzióval. A probléma átírható a következő formára

$$(\mathbf{w'} \cdot \mathbf{x'}_i)\, l_i > 0$$

ahol $1 \le i \le n$ és $\mathbf{x'}_i = (\mathbf{x}_i, 1)$ illetve $\mathbf{w'} = (\mathbf{w}, b)$.

Hogyan találjunk egy lineáris szeparáló hipersíkot?



Perceptron tanulás

Iteratív algoritmus, hogy találjunk egy lineáris szeparálót

Kezdőállapot:

Legyen $\mathbf{w} = y_1\mathbf{x_1}$ és $|\mathbf{x_i}|=1$ minden $\mathbf{x_i}$-re

Amíg létezik $\mathbf{a_i}$ melyre nem igaz az egyenlőtlenség azaz $(w \cdot a_i)l_i \leq 0$, módosítsuk a modellünket

$$\mathbf{w^{t+1}} = \mathbf{w^t} + y_i\mathbf{x_i}$$

Az algoritmus megáll ha létezik lineáris szeparáló!

Líneáris regresszió

Tegyük fel újra, hogy a tanulóhalmaz attribútumait kiegészítettük megint egy bias változóval. (X a tanulóhalmaz (N elemű), Y az osztályváltozó) Ebben az esetben az előbbi líneáris kombináció :

$$Y = X^T w$$

Szeretnénk megtalálni a regressziós görbét, melynél a négyzetesen hiba minimális (lehet más hibamértéket is alkalmazni):

$$error_{square}(f) = E[(Y - f(X))^2]$$

Mivel a tanulóhalmaz véges, a hiba :

$$error(f) = \sum_1^N (y_i - x_i^T w_i)^2$$

Líneáris regresszió

Vagy:

$$error(f)=\sum_1^N (y_i-x_i^T w_i)^2=(y-Xw)^T(y-Xw)$$

Mindig létezik minimuma és egyértelmű, így képezzük w szerinti deriváltját, s megkeressük hol 0-a:

$$-2X^T y+2X^T Xw=0 \qquad X^T(y-Xw)=0$$

Melyből következik , hogy ha nem 0-a a determináns →

$$w=(X^T X)^{-1}X^T y$$

Logisztikus regresszió

Eddig nem vettük figyelembe

Épp ezért el kell döntenünk hogy mikor döntünk 1-es osztályra vagy 0-as osztályra (feltételezve, hogy eredetileg az osztályváltozóink értékkészlete {0,1}). Amennyiben a jóslás nagyobb 0.5 akkor (ebben az esetben közelebb van 1-hez vagy nagyobb egynél) legyen a jóslat értéke 1, ha pedig kisebb 0-a.

$$y = x^T w - 0.5$$

$$f(y) = y_{osztály} = \frac{1 + sgn(y)}{2}$$

Sajnos már nem alkalmazhatunk lineáris regressziót a w meghatározásához.

Logisztikus regresszió

Hogy meghatározhassuk w-t, keresnünk kell egy olyan f(y) függvényt melyre a következő állítások igazak:

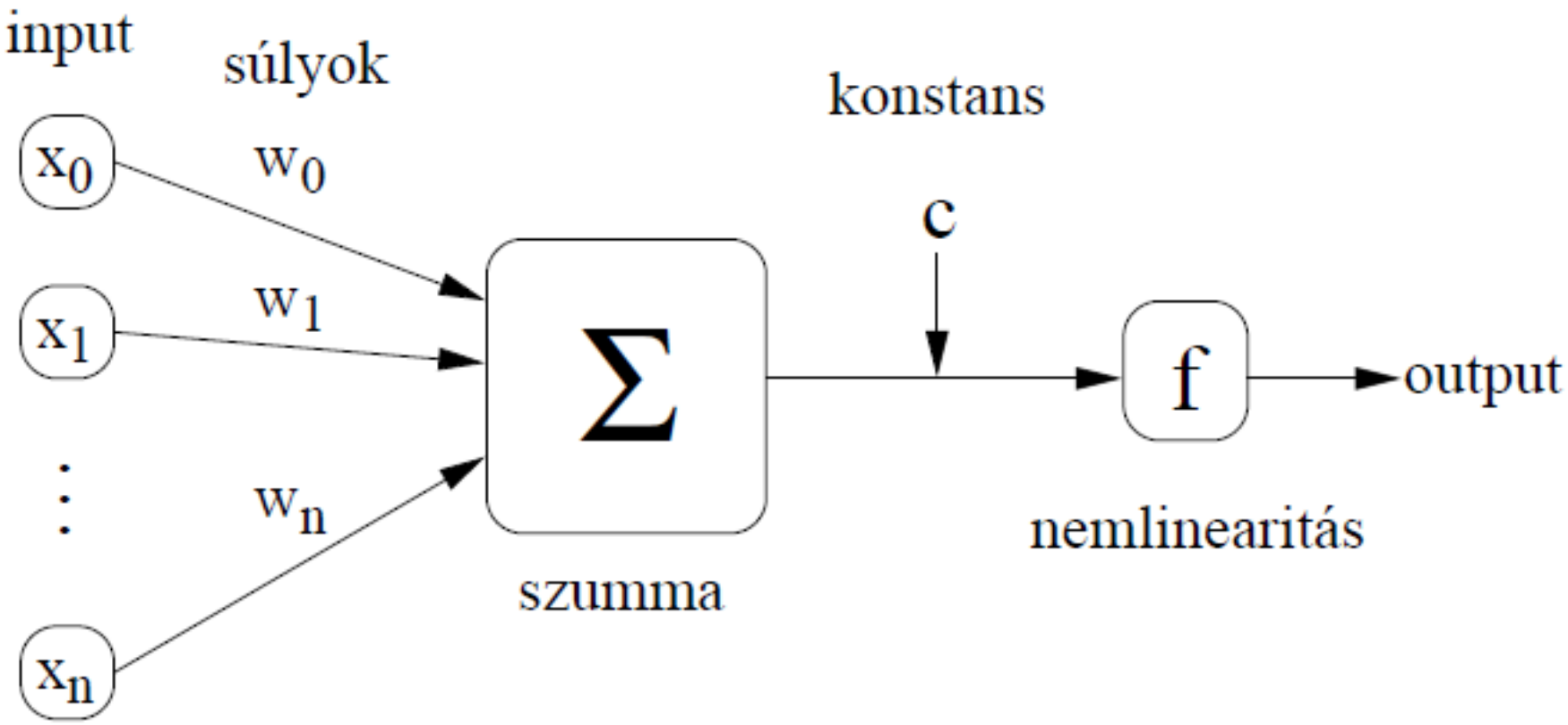
1. értéke 1-hez közelít, ha y értéke végtelenhez tart
2. értéke 0-hoz közelít, ha y értéke mínusz végtelenhez tart
3. f(0) = 0.5
4. szimmetrikus nullára nézve, tehát f(y) + f(-y) = 1 azaz 2f(0)
5. f(y) differenciálható minden pontban
6. ne legyenek lokális szélsőértékei (pl. monoton növekvő)

A szigmoid függvények megfelelnek a fenti követelményeknek

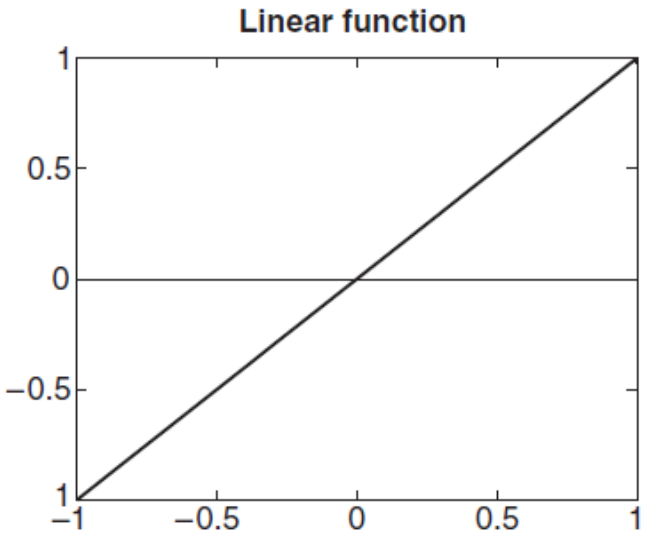
$$f(y) = 1/(1 + a^{(-y)})$$

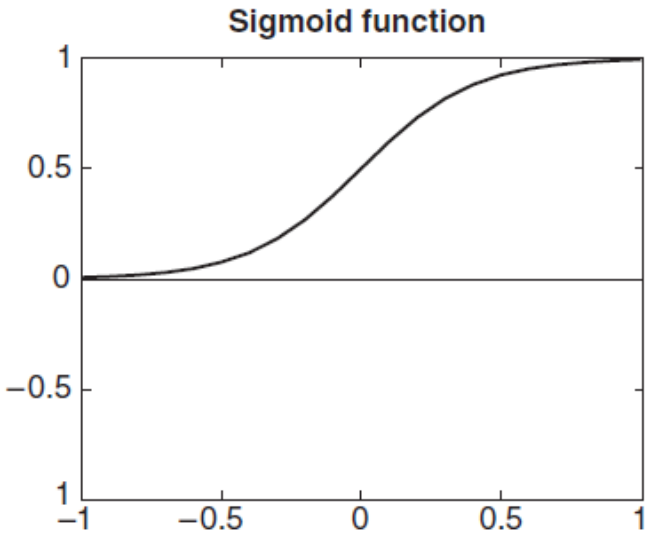
amennyiben a > 1.

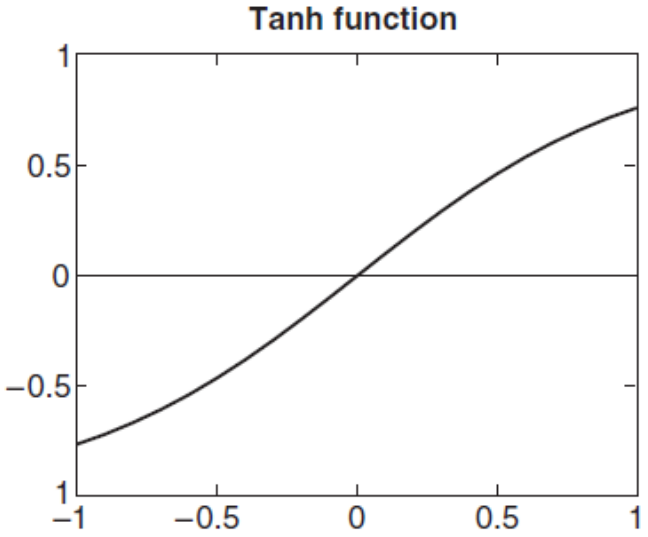
Logisztikus regresszió

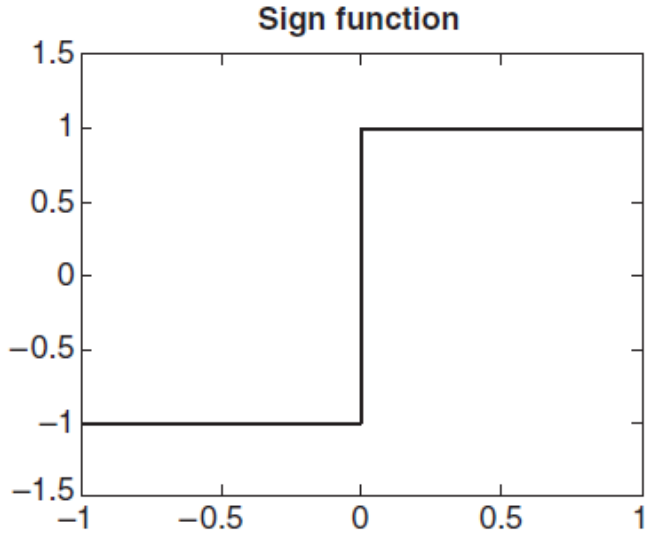


Aktiválási függvények









Logisztikus regresszió

Optimalizálás lehet pl. gradiens módszerrel (részletesen később):

$$w_{opt}=\text{argmax}_w \sum \ln(P(y_i|x_i,w))$$

Viszont felhasználhatjuk, hogy bináris osztályozás esetén $y_i=0$ vagy $y_i=1$:

$$L(w)=\sum y_i \ln(P(y_i=1|x_i,w)) + (1-y_i) \ln(P(y_i=0|x_i,w))$$

Kiindulunk $w^{(0)}$-ból, majd minden iterációban $w^{(l)}$-hez hozzáadjuk $dL(w)/dw$ (parciális deriváltak) λ szorosát (mely egy előre meghatározott konstans, a tanulási növekmény): (w_j-re)

$$\sum x_{ij}(y_i - P(y_i|x_{ij},w_j))$$

Logisztikus regresszió

A fenti feltételezésünk átfogalmazva:

$$\ln \frac{p(x)}{1-p(x)} \approx x^T \omega + \omega_0$$

Melyből: $$p(x \mid \omega) = sigm(x \mid \omega) = \frac{1}{1+\mathrm{e}^{-(x^T\omega+\omega_0)}}$$

Amennyiben a log-likelihood-ra optimalizálunk (tanulóhalmaz $X=\{x_1,..,x_t\}$) és a mintáinkat függetlennek tekintjük:

$$\frac{\partial \mathcal{L}(\omega \mid X)}{\partial \omega_i} = \sum_{x_t \in X^{(+)}} \frac{\partial \ln p(x_t \mid \omega)}{\partial \omega_i} + \sum_{x_t \in X^{(-)}} \frac{\partial \ln(1-p(x_t \mid \omega))}{\partial \omega_i}$$

$$= \sum_{x_t \in X^{(+)}} (1-p(x_t \mid \omega))x_{ti} - \sum_{x_t \in X^{(-)}} p(x_t \mid \omega)x_{ti}$$

$$= \sum_{x_t \in \{X^{(-)},X^{(+)}\}} (y_t - p(x_t \mid \omega))x_{ti}$$