Alkalmazott mesterséges intelligencia (AMI)

http://www.mit.bme.hu/oktatas/targyak/vimibb01

7. ea. (2019 ősz)

Gépi tanulás

Pataki Béla

BME I.E. 414, 463-26-79

pataki@mit.bme.hu,

http://www.mit.bme.hu/general/staff/pataki

A múlt órán olyan módszereket kezdtünk vizsgálni, amikor minták, **mintapéldák alapján akarjuk kialakítani a döntési rendszerünket**.

Nagyon gyakran a számítógép tanulásához rendelkezésünkre áll egy csomó mintapélda, és ez hordozza az információt. Nincsenek előre kialakított szabályaink a feladatra, csak a mintáink. Persze ilyenkor is valamilyen struktúrában igyekszünk felhasználni a minták hordozta tudást.

Ilyenkor a módszerünk – az, hogy hogyan használjuk fel a mintákat – bizonyos mértékben problémafüggetlen.

A hétköznapi életben nagyon gyakori a minta alapján való tanulás, míg a klasszikus (számítógépes) megoldásokban a humán fejlesztő szabályokat alkotott, és ezeket programozta be.

Egy algoritmust, pl. egy döntést (ágensfüggvényt) alapvetően két módon lehet megvalósítani?

(1) Analitikus tervezés:

Begyűjteni az analitikus (fizikai, kémiai stb. összefüggésekből felépített) modelleket az adott problémára. Példa: logikai modellek, szabályalapú rendszerek

Megtervezni analitikusan a konkrét mechanizmust, és azt algoritmusként implementálni.

(2) **Tanulás:**

Megtervezni a tanulás mechanizmusát, és azt algoritmusként implementálni.

Majd alkalmazásával megtanulni vele a minták alapján a döntés tényleges mechanizmusát, és azt algoritmusként implementálni.

A következőkben a (2)-vel foglalkozunk

Sokféle tanítható eszközt kitaláltak:

- neurális hálók
- Bayes-hálók
- kernelgépek
- döntési fák
- legközelebbi szomszéd osztályozók
- stb. stb.

A múlt órán a döntési fákkal foglalkoztunk – működésük az emberi gondolkodás számára jóval jobban áttekinthető, mint pl. a mesterséges neurális hálóké. (white box $\Leftrightarrow$ black box)

(Vannak regressziós döntési fák is, de mi csak az osztályozással foglalkozunk.)

Vizsgáljuk meg a gépi tanulás alapvető fajtáit: **kvíz következik**

felügyelt tanulás egy-egy esetnél mind a bemenetet, mind a kimenetet észlelni tudjuk (bemeneti minta + kívánt válasz)

megerősítéses tanulás az ágens az általa végrehajtott tevékenység csak bizonyos értékelését kapja meg, esetleg nem is minden lépésben (jutalom, büntetés, **megerősítés**)

felügyelet nélküli tanulás semmilyen információ sem áll rendelkezésünkre a helyes kimenetről (az észlelések közötti összefüggések tanulása)

féligellenőrzött tanulás a tanításra használt esetek egy részénél mind a bemenetet, mind a kimenetet észlelni tudjuk (bemeneti minta + kívánt válasz), a másik – tipikusan nagyobb – részénél csak a bemeneti leírás ismert

07.1. kvíz

A múlt órán vizsgált – döntési fa nevű – eszközt milyen jellegű tanulással tanítottuk?

A. Felügyelt tanulás
B. Megerősítéses tanulás
C. Felügyelet nélküli tanulás
D. Féligellenőrzött tanulás

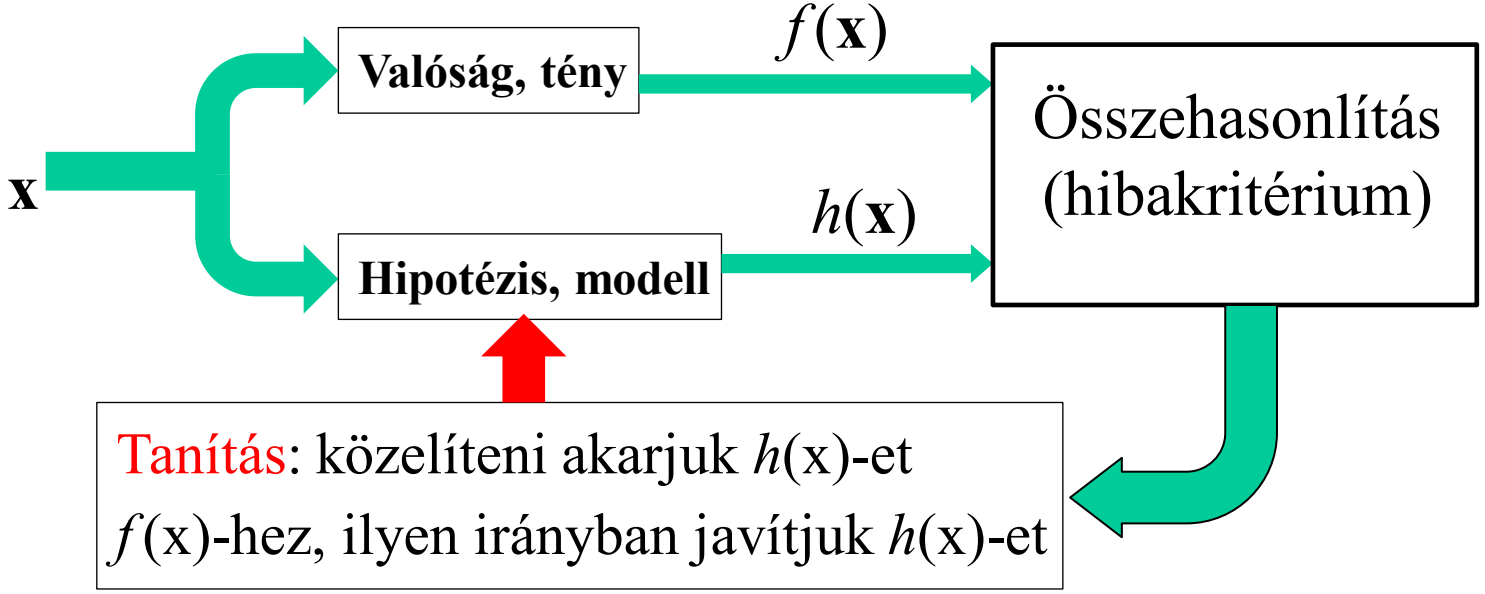
Felügyelt tanulás (pl. induktív következtetés):

tanulási példa: $(\mathbf{x}, f(\mathbf{x}))$ adatpár, ahol $f(\mathbf{x})$ ismeretlen

tanulás célja: $f(\mathbf{x})$ értelmes közelítése egy $h(\mathbf{x})$ hipotézissel

$h(\mathbf{x}) = f(\mathbf{x})$, $\mathbf{x}$ – ismert példákon gyakran teljes pontosság

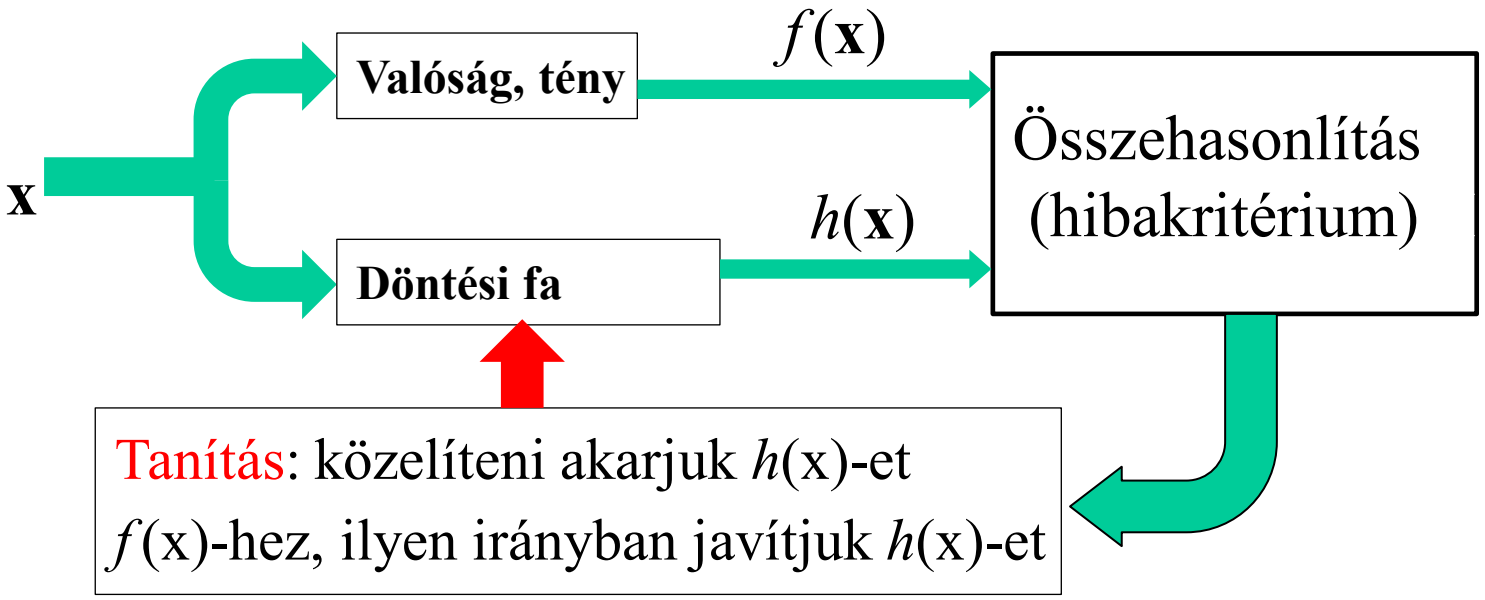
$h(\mathbf{x}') \cong f(\mathbf{x}')$, **$\mathbf{x}'$ – a tanulás közben még nem látott esetek (általánosító képesség)**



Feladat: *az ismeretlen **f**-re vonatkozó példák egy halmaza alapján (**tanítóhalmaz**), adjon meg egy olyan **h** függvényt (**hipotézist**), amely tulajdonságaiban jól közelíti az **f**-et (amit **teszthalmazon** verifikálunk).*

Az eddigi tanítási példánk (döntési fák) ebbe a – felügyelt tanulás - csoportba tartozik.

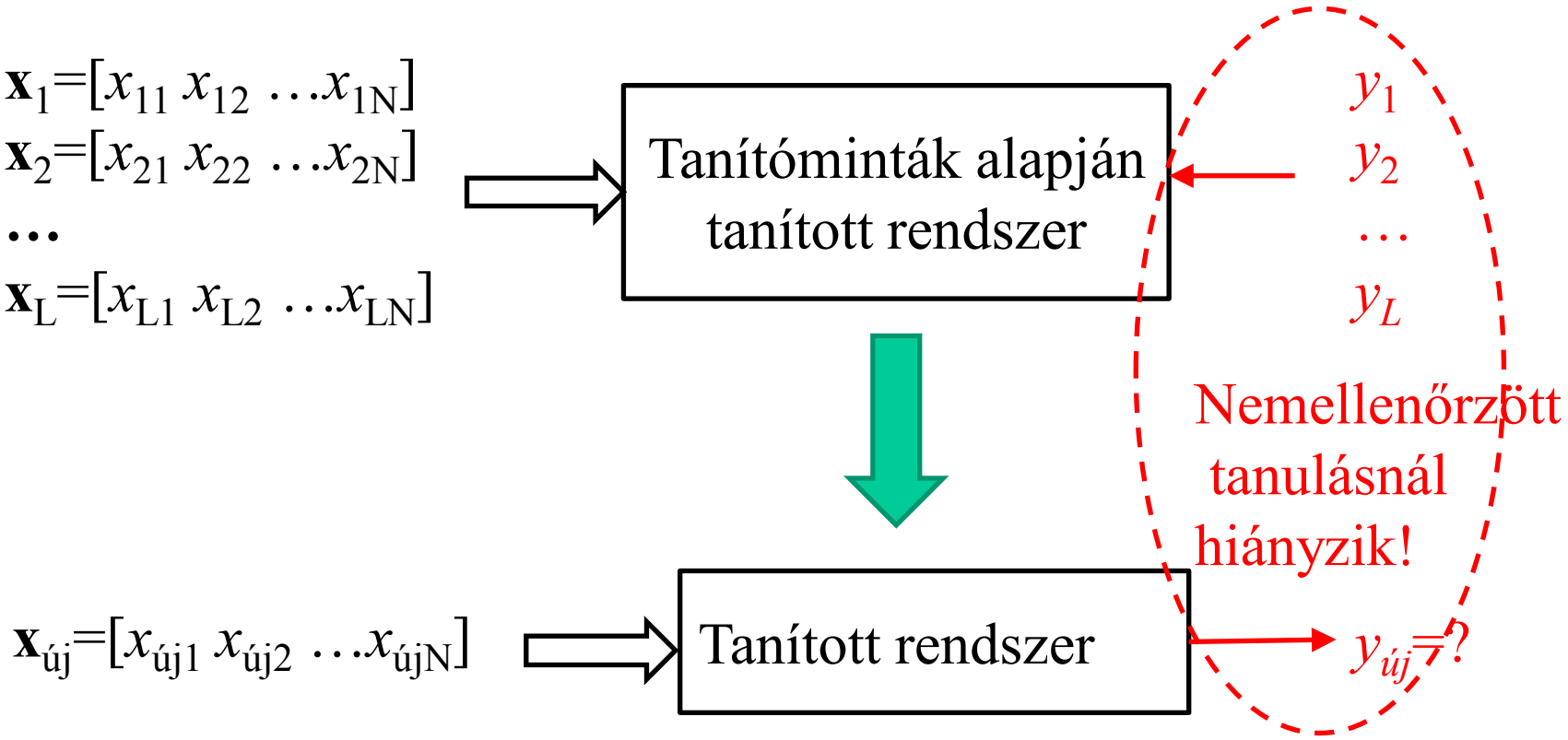
Itt a tanítás az újabb és újabb csomópontok kifejtést jelenti, ezzel közelítjük a rendszerünk válaszát a megkívánt (a mintákban adott) válaszhoz



Klaszterezés (nemellenőrzött tanulás)

Az eseteket szokásos módon az **x** paramétervektor írja le (fényesség, kontraszt, textúraparaméterek stb.).

Összehasonlítás céljából: a besorolást ***ellenőrzött tanulás*** (osztályozás) esetén az *y* kimeneti címke adja meg.



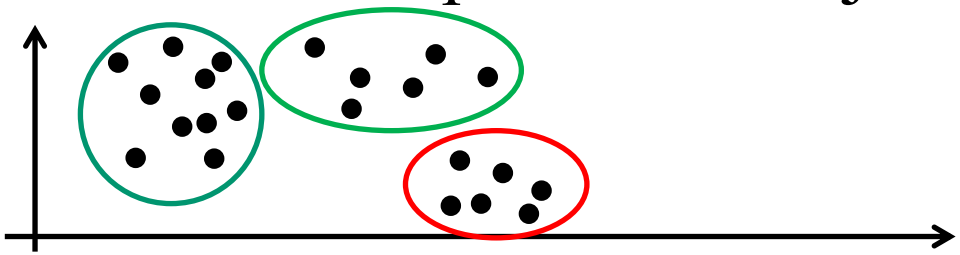
Célunk rendszerint annak megadása, hogy az újonnan érkező $\mathbf{x}_{új}$ mintához milyen valószínűséggel tartozik egy y címke, ehhez a $P(y|\mathbf{x})$ eloszlást kéne ismernünk.

Bayes-tétel: $$P(\mathbf{x}, y) = P(y|\mathbf{x})P(\mathbf{x}) = P(\mathbf{x}|y)P(y)$$

*Ellenőrzött tanulás*nál a tanulóminta-halmazból becsüljük $P(\mathbf{x}|y)$-t és $P(\mathbf{x})$-t $\Rightarrow$ innen becsüljük $P(y|\mathbf{x})$-t.

*Nemellenőrzött tanulás*nál nem ismerjük az y címkéket: csak $P(\mathbf{x})$-t tudjuk becsülni.

Klaszterezés: a $P(\mathbf{x})$ olyan becslése, melyben a mintatérben elkülönülő csoportokra bontjuk a mintákat



Miért lehet jó klaszterezni a mintákat?

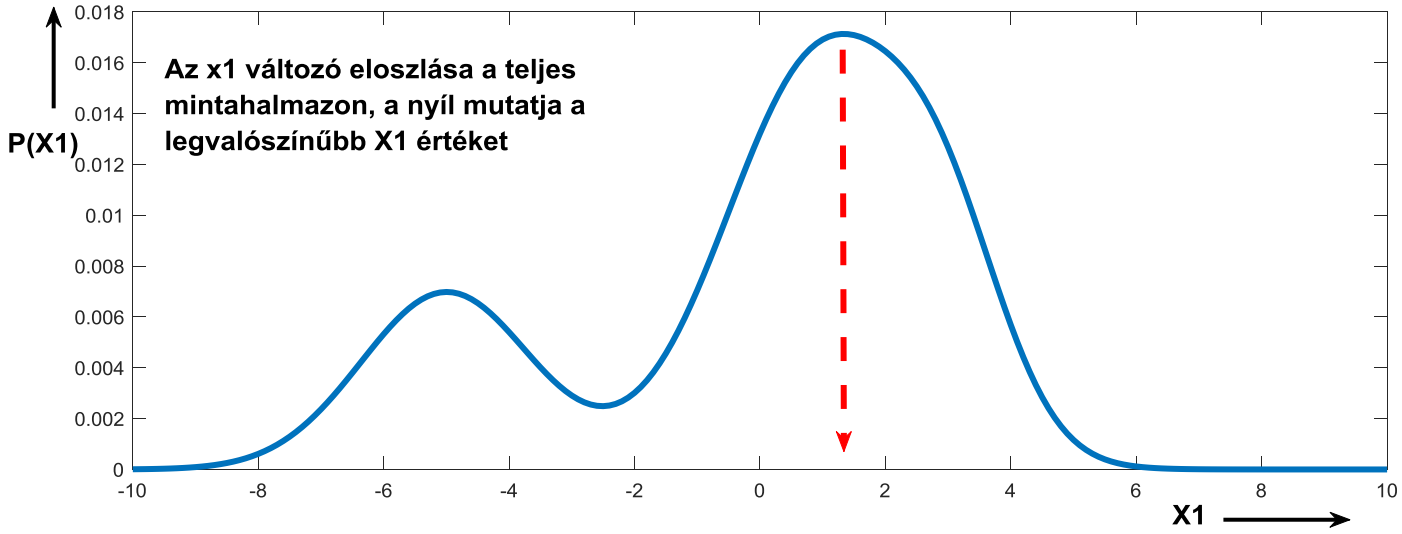
segíthet a kilógó (outlier) adatok felderítésében – ha mindegyik kialakuló klasztertől távoli adat érkezik, valószínűleg hibás (kilógó, outlier)

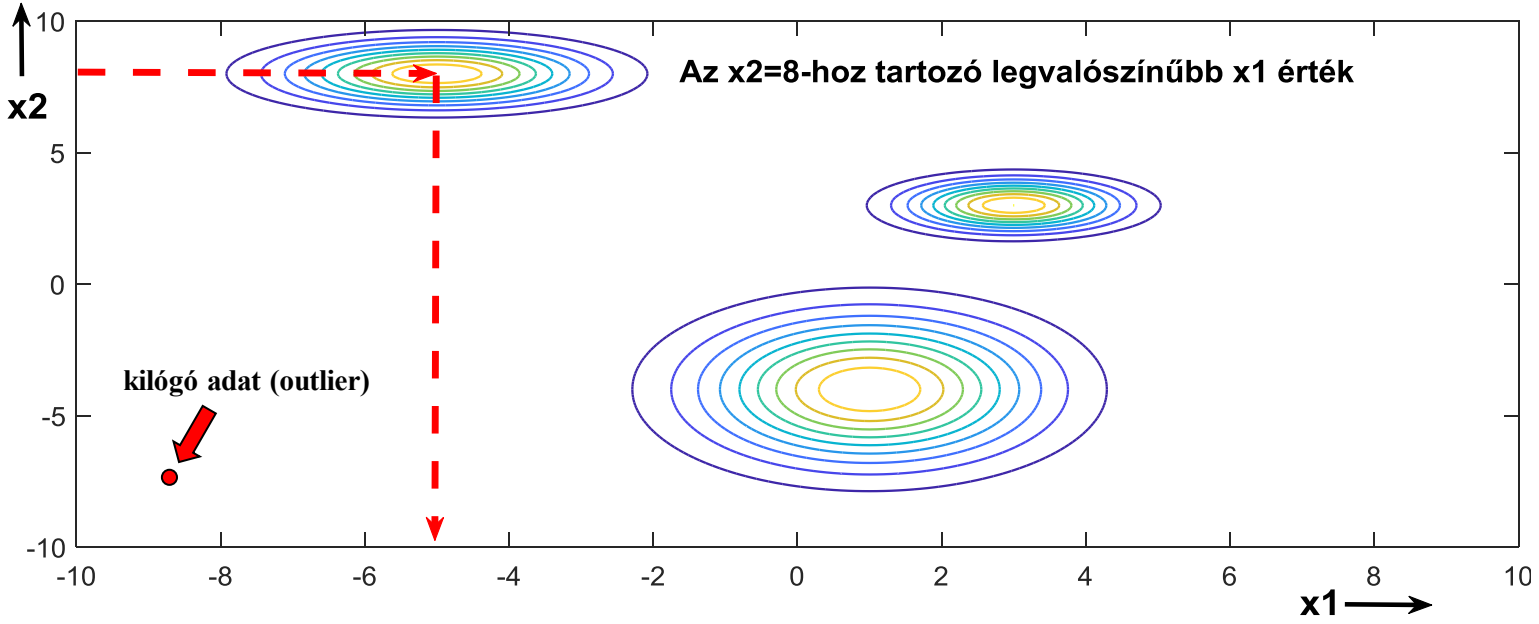
megmutatja, hogy a jelenségnek, folyamatnak vannak-e tipikus állapotai – pl. egy gyártási folyamatban, egy elektromos vagy egyéb fogyasztási viselkedésben lehetnek tipikus helyzetek

felhasználható hiányzó paraméterek pótlására – ha el tudjuk dönteni, vagy valószínűsíteni tudjuk, hogy melyik klaszterbe tartozik a néhány paraméterében hiányos adat, akkor a klaszter tipikus paraméterei jobb javítást, pótlást tesznek lehetővé

Demópélda

n-dik minta: x_{n2}=8, de hiányzik az x_{n1} paraméter. Nézzük meg, melyik x_{n1} legvalószínűbb értéke az x_1 eloszlás alapján:

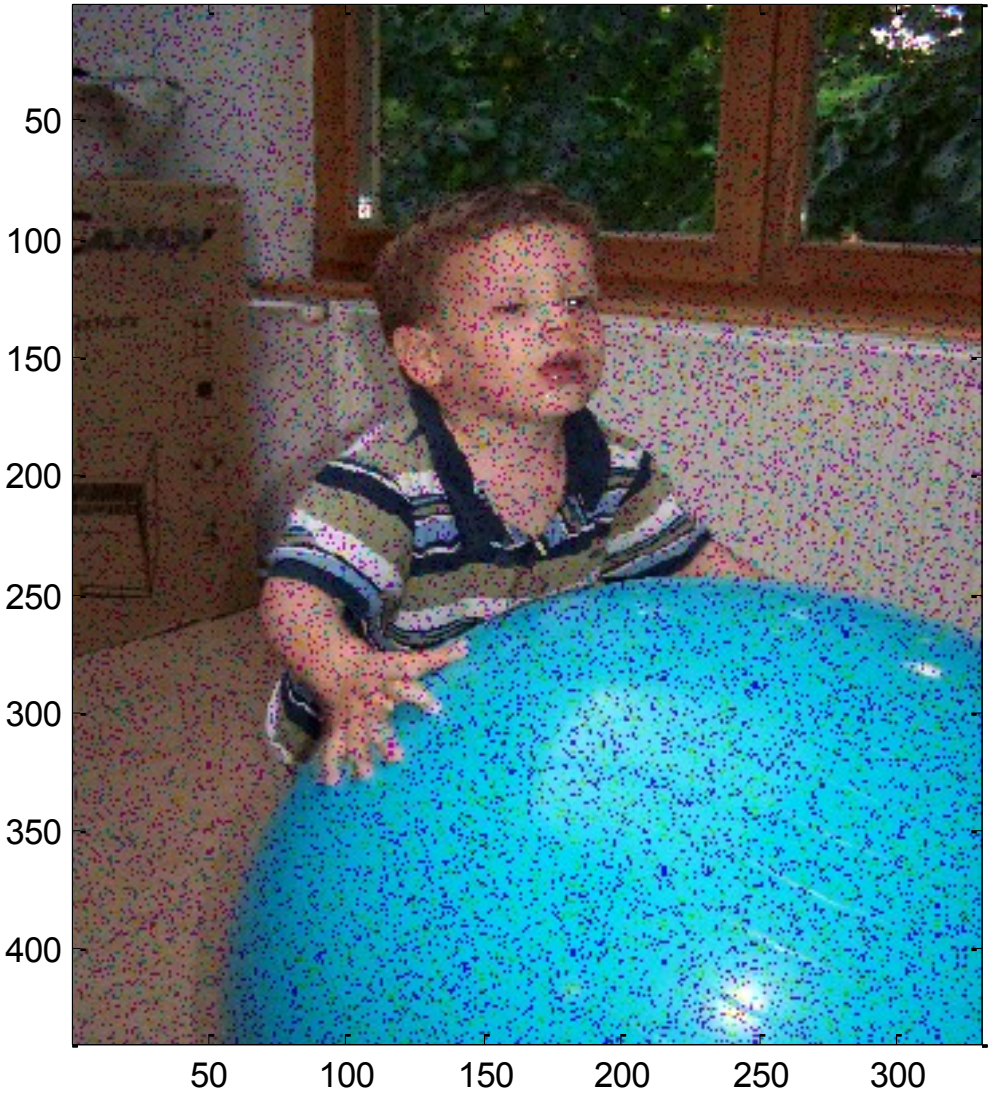




Második demópélda

Eredeti kép

20%-ban hibás pixelek (1-1 színkomponens elveszett)

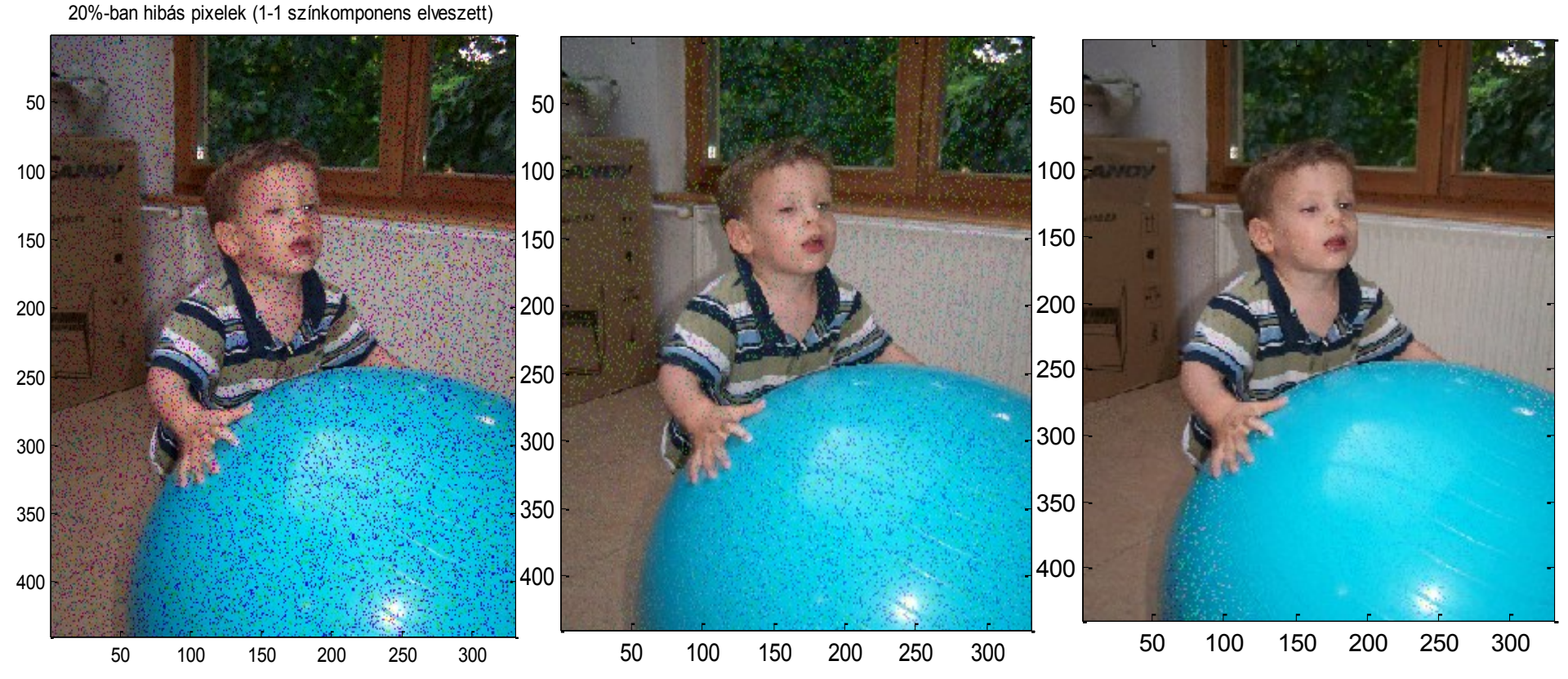


Balról-jobbra:

a hibás kép (20% pixel egyik komponense hiányzik),

a paraméterek globális átlagával javítottuk a hiányzó komponenseket,

klaszterenkénti átlagparaméterekkel javítottunk (a maradék két komponens alapján eldöntjük, hogy valószínűleg melyik klaszterbe tartozik)



Klaszterezési eljárások

Diszkriminatív:

az egyes esetek (minták) közti távolságot használjuk fel: a közeli minták tartozzanak egy csoportba, a távoliak külön csoportba.

Kritikus a jó távolságmérték megtalálása!

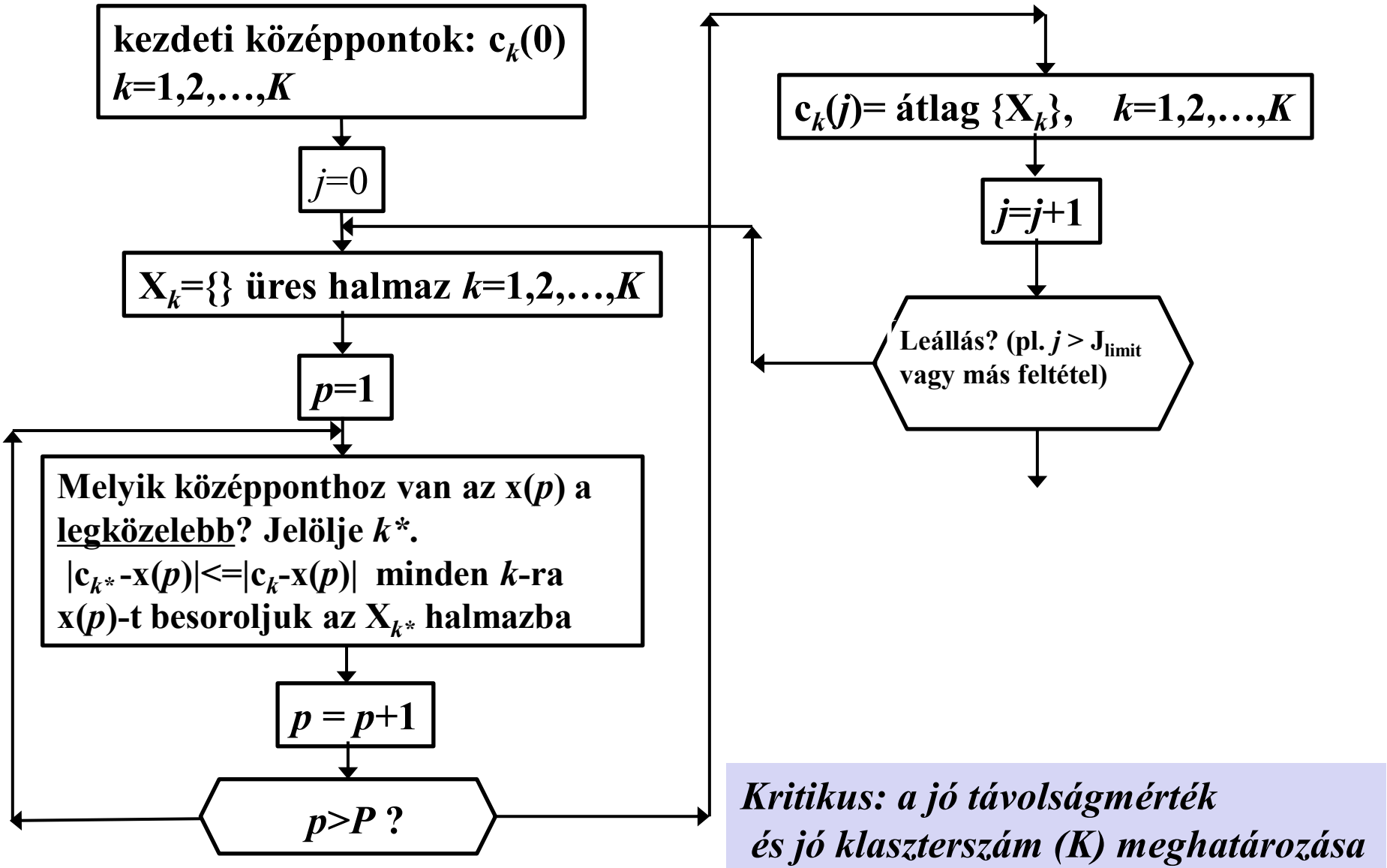
Generatív (ezzel részletesebben nem foglalkozunk):

Egy, a mintahalmaz létrehozását (generálását) magyarázó modellt használunk. A modell magyarázza az egyes csoportok létrejöttét – a modellparamétereket tanuljuk.

Kritikus a jó modell megtalálása!

Diszkriminatív klaszterezés

K–átlagképző, K-középpontképző eljárás (*K-means*)



Kritikus: a jó távolságmérték és jó klaszterszám (K) meghatározása

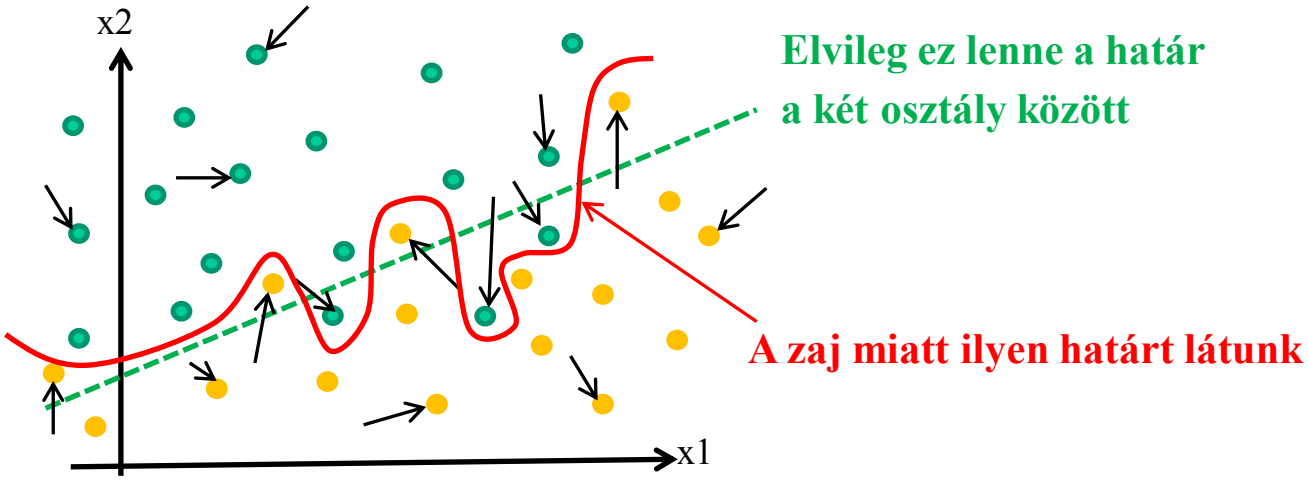
kvíz következik!

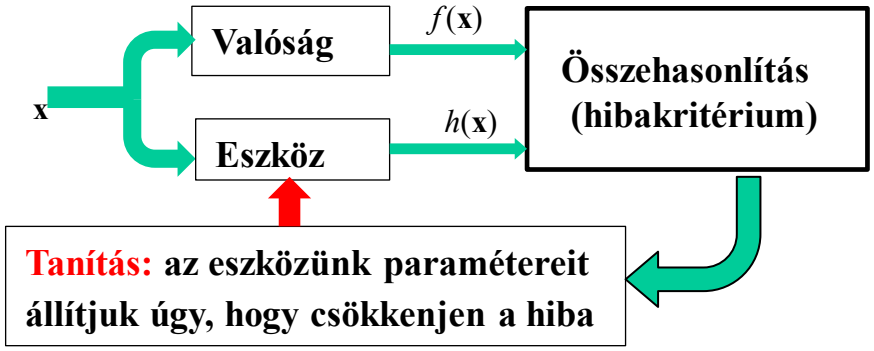
Az egyik általános probléma a gépi (de nem csak a gépi) tanulásnál: **a**

túltanulás

A túltanulás akkor lép fel, amikor már nem a lényegi információt, hanem a mintákban mindig jelenlévő zajt tanuljuk.

Kétosztályos problémát tanulunk minták alapján. A mintáinkat 2 paraméter jellemzi (x1 és x2). A zaj miatt nem a valós x1 és x2 értékeket mérjük, a pontokat a valós helyükről elmozdulva látjuk.





08.1. kvíz

Tanítható eszközünk tanítását néhány szabad paraméter változtatásával végezzük annak érdekében, hogy a 10.000 tanítóminta-pontunkon minél kisebb hibát vétsünk. Várhatóan melyik eszköz lesz a leghajlamosabb a túltanulásra?

A. Amelyiknek 2 állítható szabad paramétere van

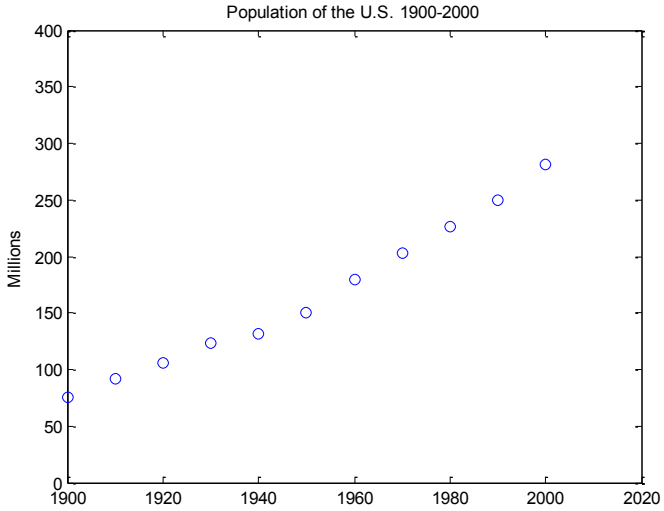
B. Amelyiknek 100 állítható szabad paramétere van

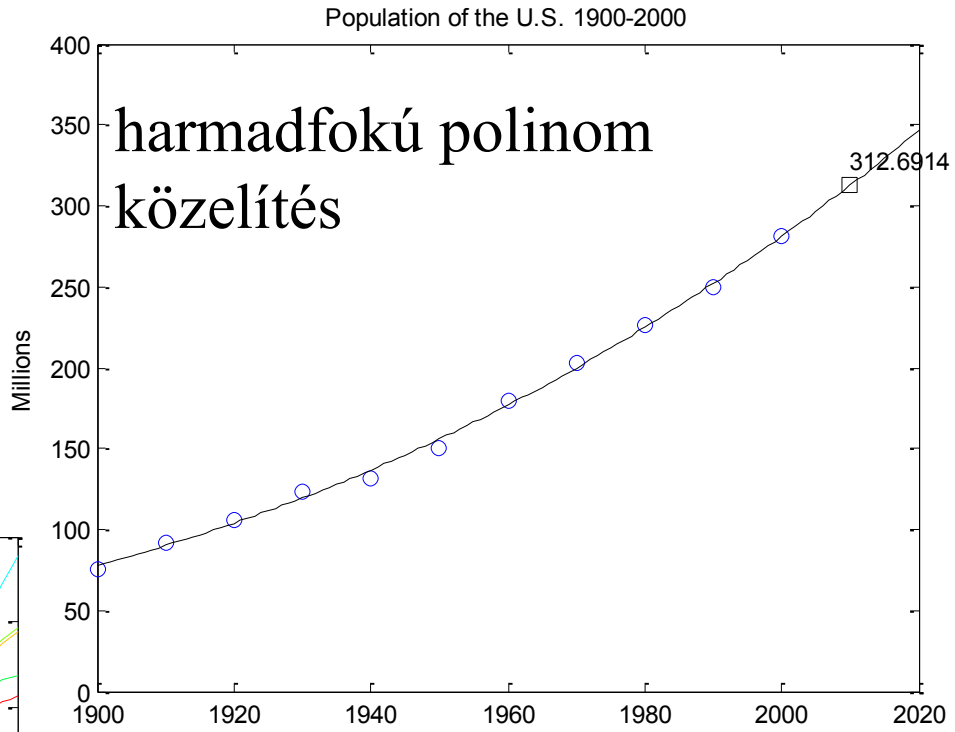
C. Amelyiknek 1000 állítható szabad paramétere van

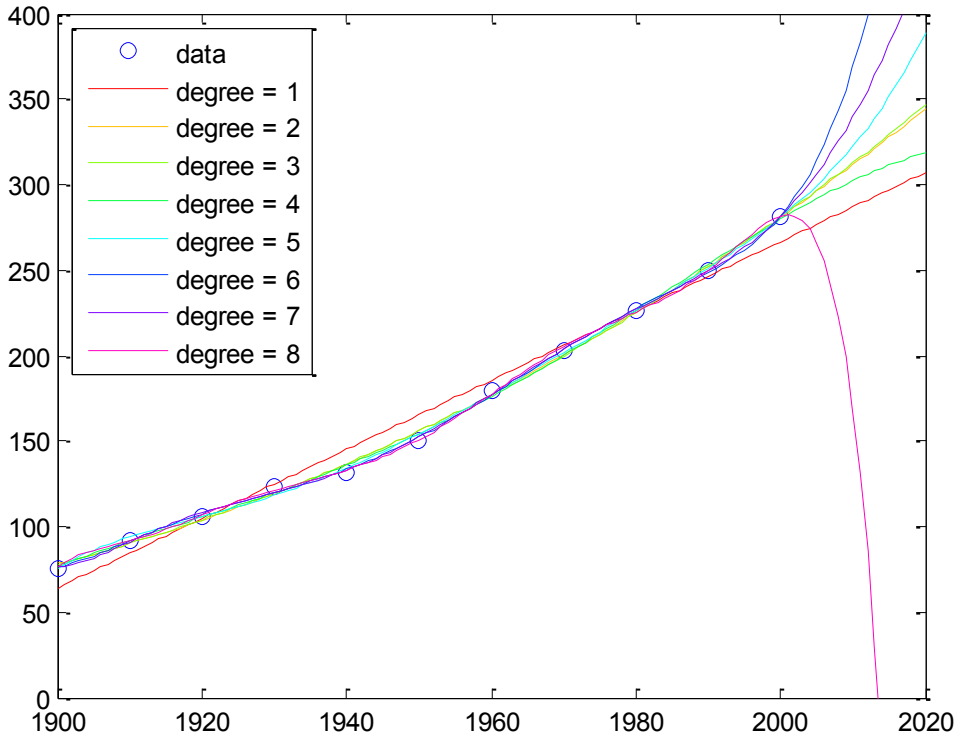
D. Amelyiknek 2000 állítható szabad paramétere van

A mintákat torzító zaj hatása – túltanulást okozhat!

(a faék egyszerűségű gondolkodás dicsérete! ☺)







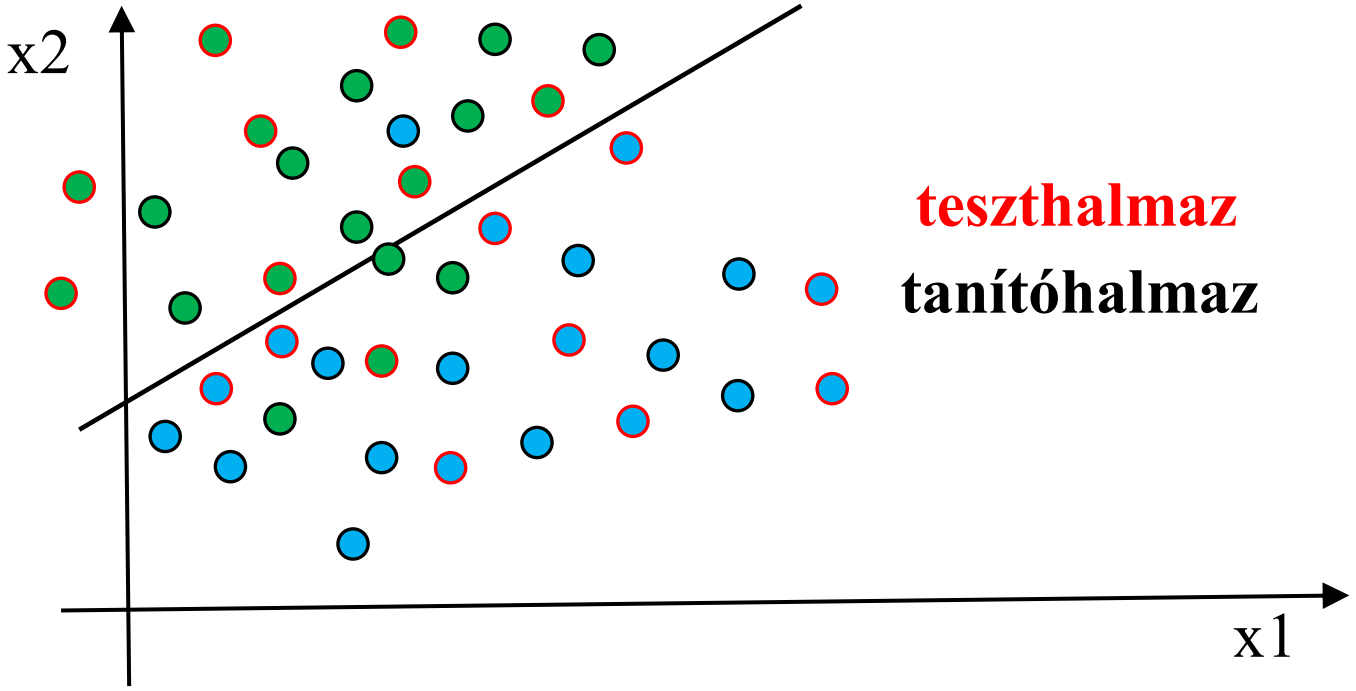
Matlab demó: az 1900-2000 közti népszámlálási adatokból becsüljük 2010-es USA népességet. (Regressziós feladat, de a jelenség a döntéseknél hasonló hatást okoz.)

A nagyon „okos” = túl komplex eszköz a zajt is megtanulja, rosszabbul becsül, mint az egyszerűbb!

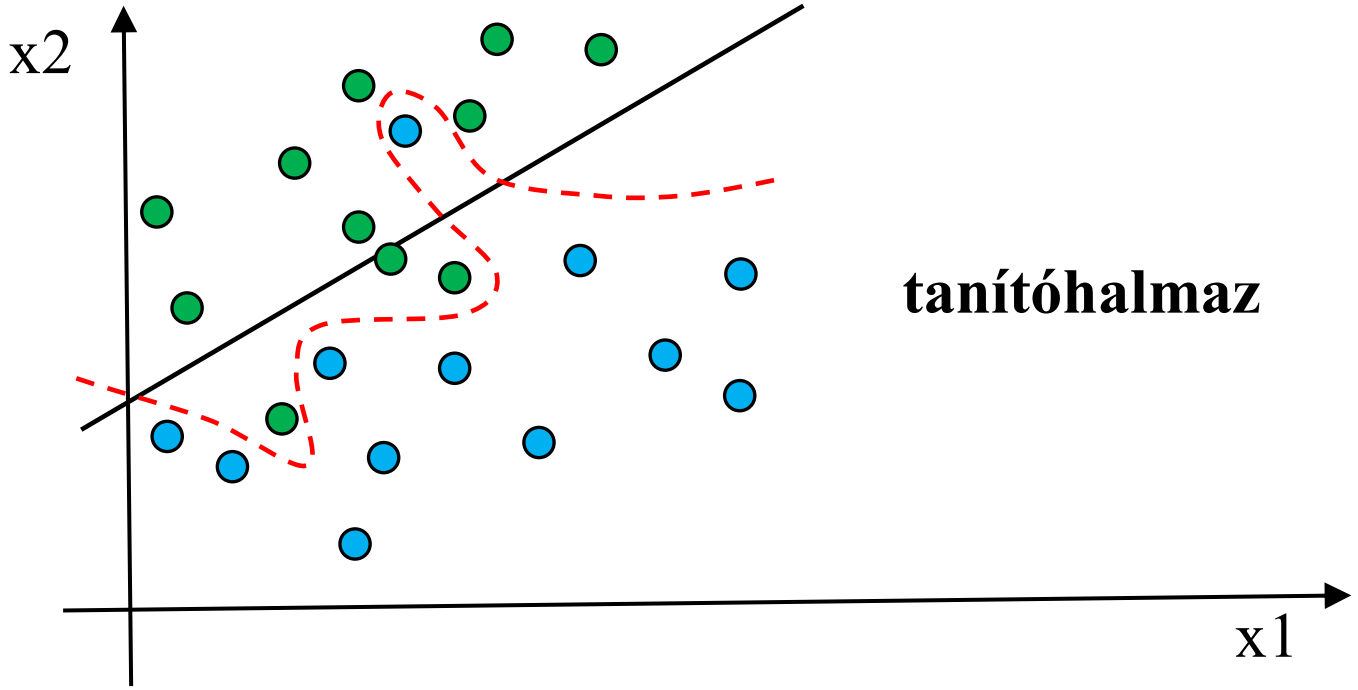
Mikor álljunk le a tanulással? (A döntési fa növesztésével?)

(A túltanulás elleni alapvető módszerek bemutatását az eddigiekben tárgyalt eszközön – döntési fákon végezzük.)

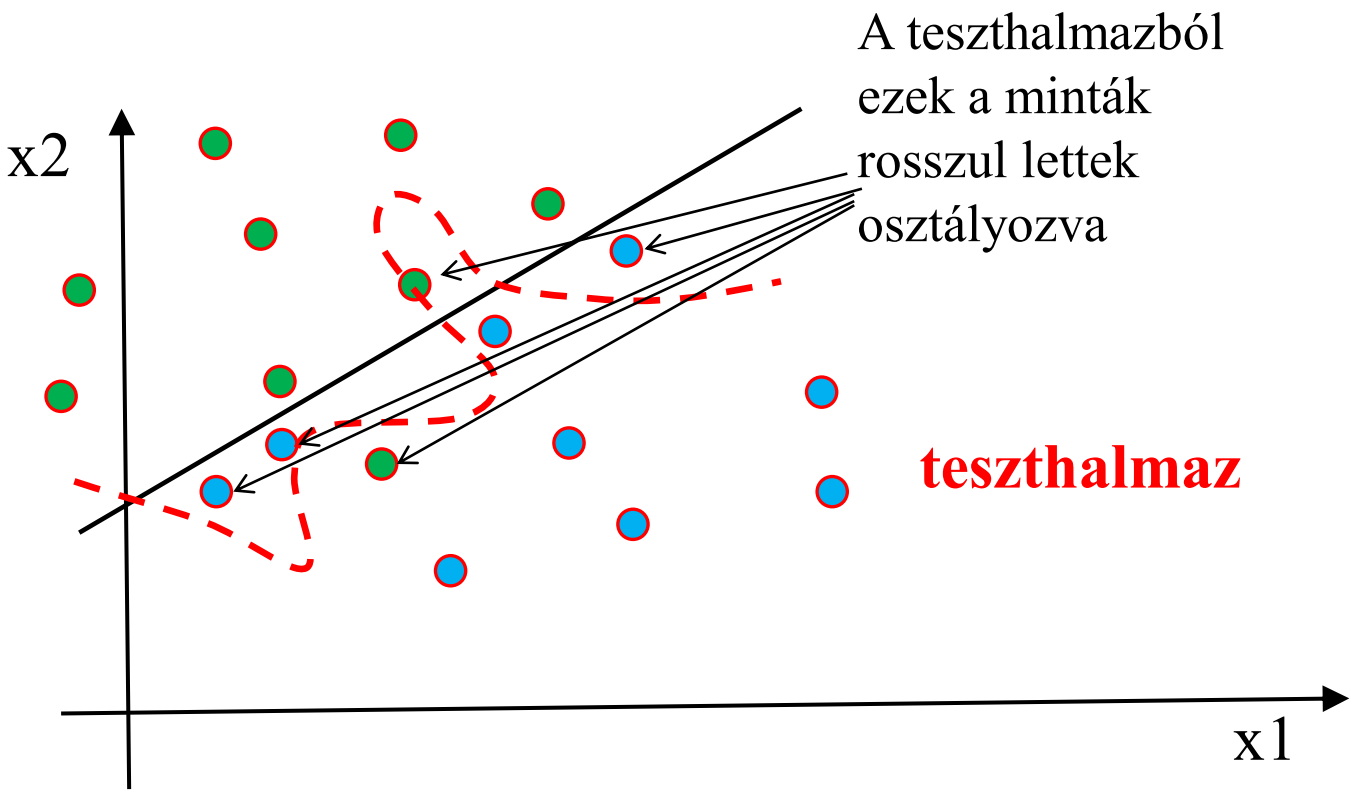
1. stratégia: Korai leállás



1. stratégia: Korai leállás



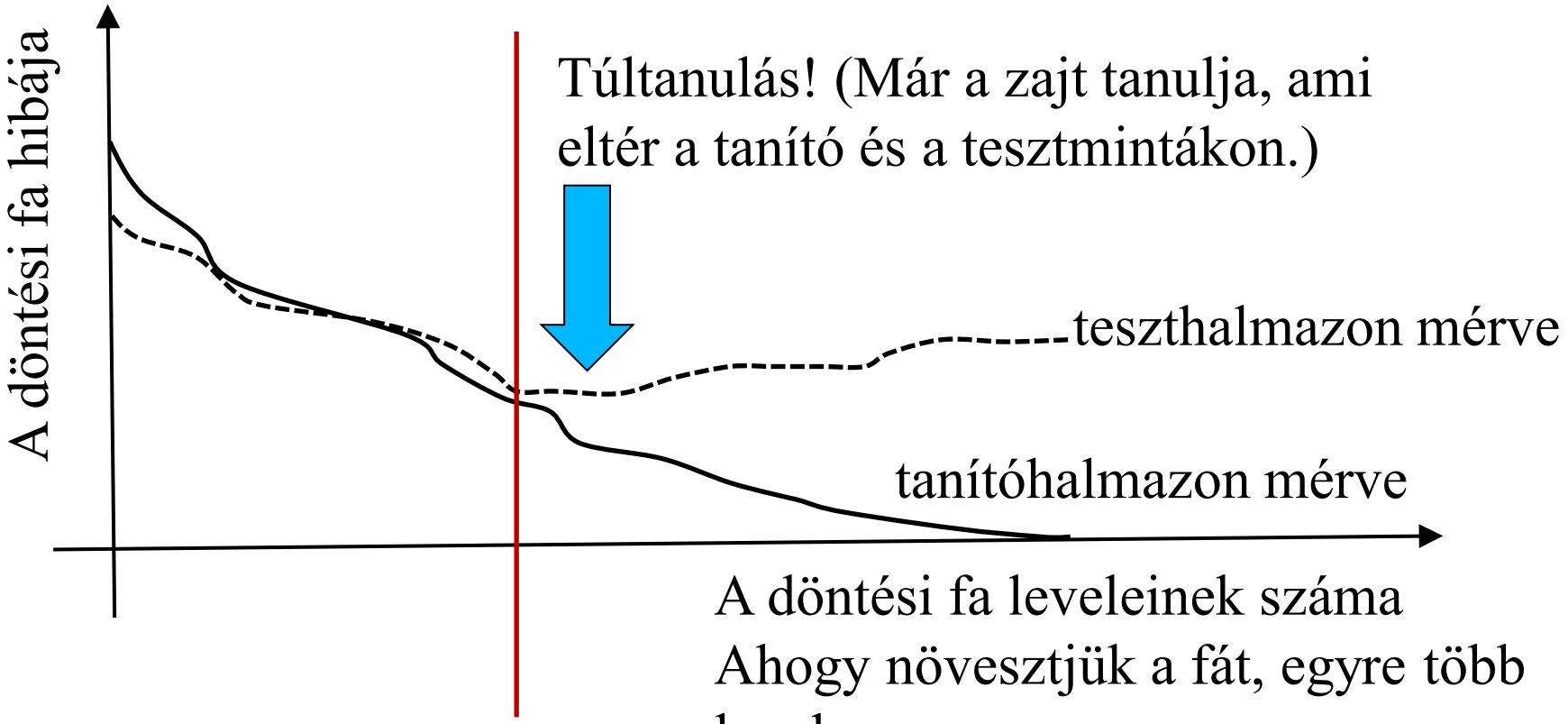
1. stratégia: Korai leállás



Mikor álljunk le a tanulással? (A döntési fa növesztésével?)

(A túltanulás elleni alapvető módszerek bemutatását az eddigiekben tárgyalt eszközön – döntési fákon végezzük.)

1. Stratégia: Korai leállás

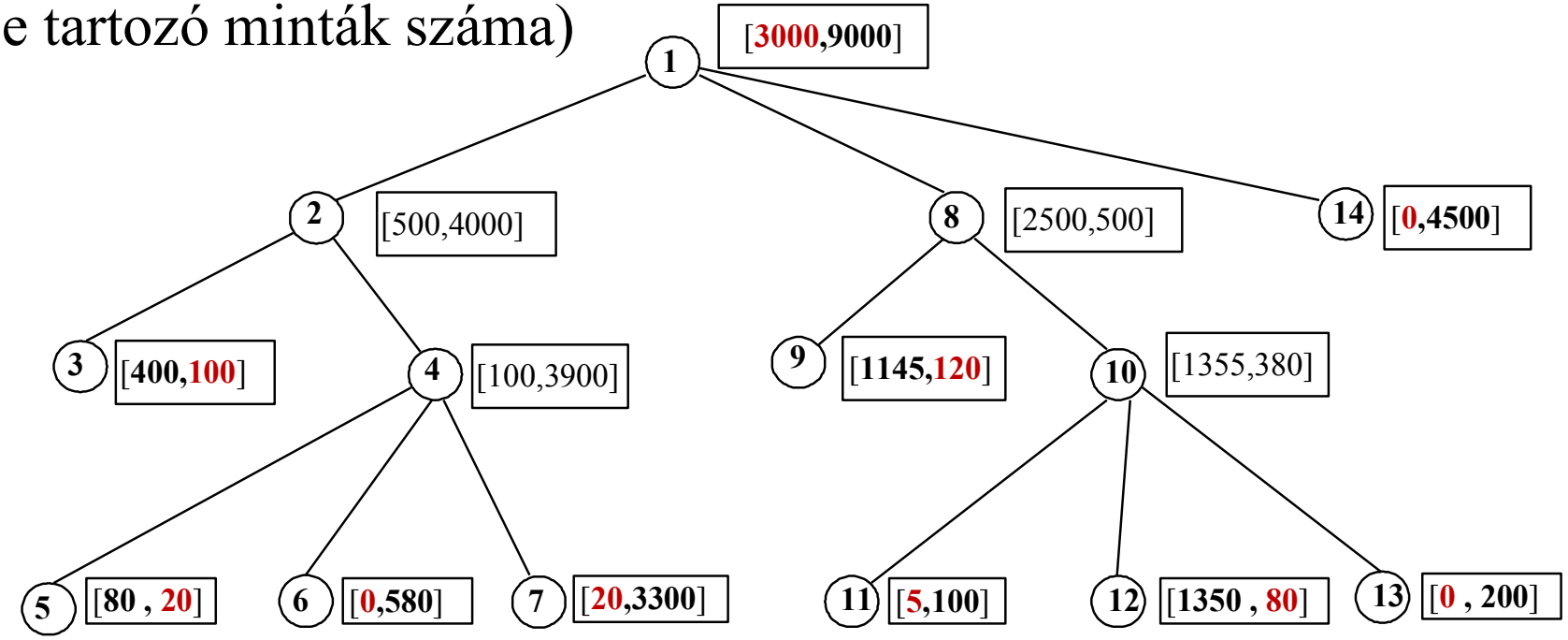


(A levelekkel mérhetjük, hogy mekkorára nőtt a fa, mennyire komplex.)

2. Stratégia

Egy kétosztályos (bináris) osztályozási példa kapcsán

(Baloldali szám a C1 osztályba tartozó minták száma, jobboldali a C2-be tartozó minták száma)

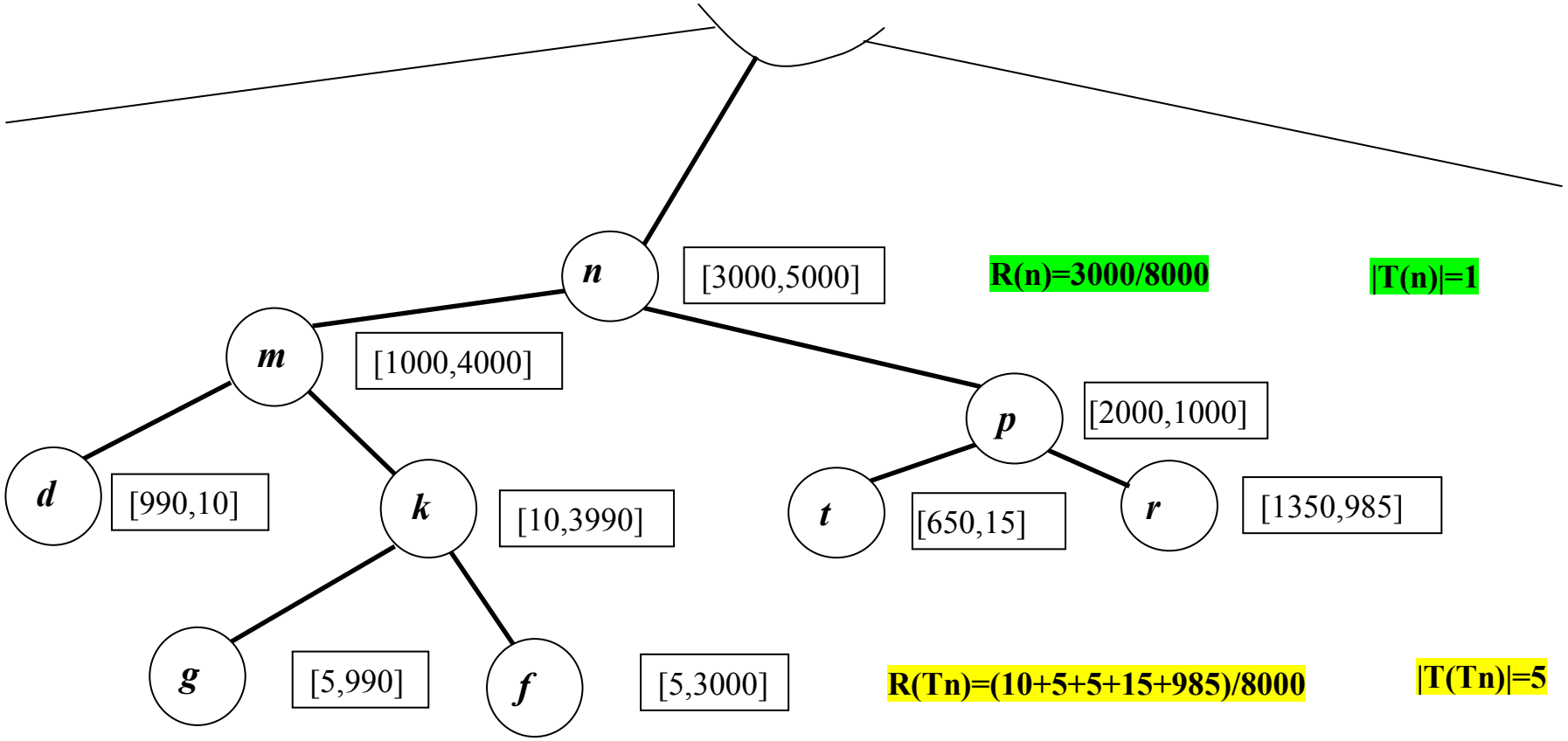


Komplexitás – legyen a levelek száma (lehetne más is), a példánkban:
a gyökérnél 1, a kifejtett fánál 9

Hibaarány:
a gyökérnél 3000/12000 (hiszen a jobb válaszunk C2 lenne),
a kifejtett fánál: (100+20+0+20+120+5+80+0+0)/12000

Számozzuk a csomópontokat, jelölés: **kvíz következik**

- az *n*-dik csomópontot önmagában (mintha levél lenne) jelölje *n*
- az *n*-edik csomópontból kiinduló részfa T*n*



A **hiba** is fáj nekünk, mert téves döntést hozunk miatta. A **komplexitás** is fáj nekünk, mert bonyolultabb (lassabb, drágább) lesz az eszközünk, ráadásul könnyebben esik a túltanulásba.
Viszont a hibát általában úgy tudjuk lejjebb vinni, ha a komplexitást növeljük. (A komplexebb, összetettebb tudású szakember vagy MI eszköz – kevesebbet téved.)

Kvíz 08.2.

Mit tudunk kezdeni a kétféle szemponttal?

A. **Nincs általános megoldás, mindig újra el kell döntetnünk, hogy melyik fáj jobban**
B. **Mindig a komplexitáscsökkentésnak adjunk nagyobb prioritást**
C. **Mindig a hibaarány-csökkentésnek adjunk nagyobb prioritást**
D. **Meg tudjuk határozni, hogy melyik a leggyengébb teszt a fában**

2. Stratégia: *Döntési fa nyesése (pruning):*

(1) Beállított max. mélység (mint a mélységkorlátozott keresésnél), ha elérjük, akkor ezen a szinten mindenképpen leállunk, nem fejtjük ki tovább az adott ágat.

(2) Vizsgáljuk meg az elvégzett teszteket, ismerjük fel a nem releváns (semmitmondó) felhasznált attribútumot, és az adott ágat ott fejezzük be (metsszük vissza a fát). Ennek egyik – jól áttekinthető) módszere a

Hibaarány – komplexitás kompromisszumon alapuló metszés

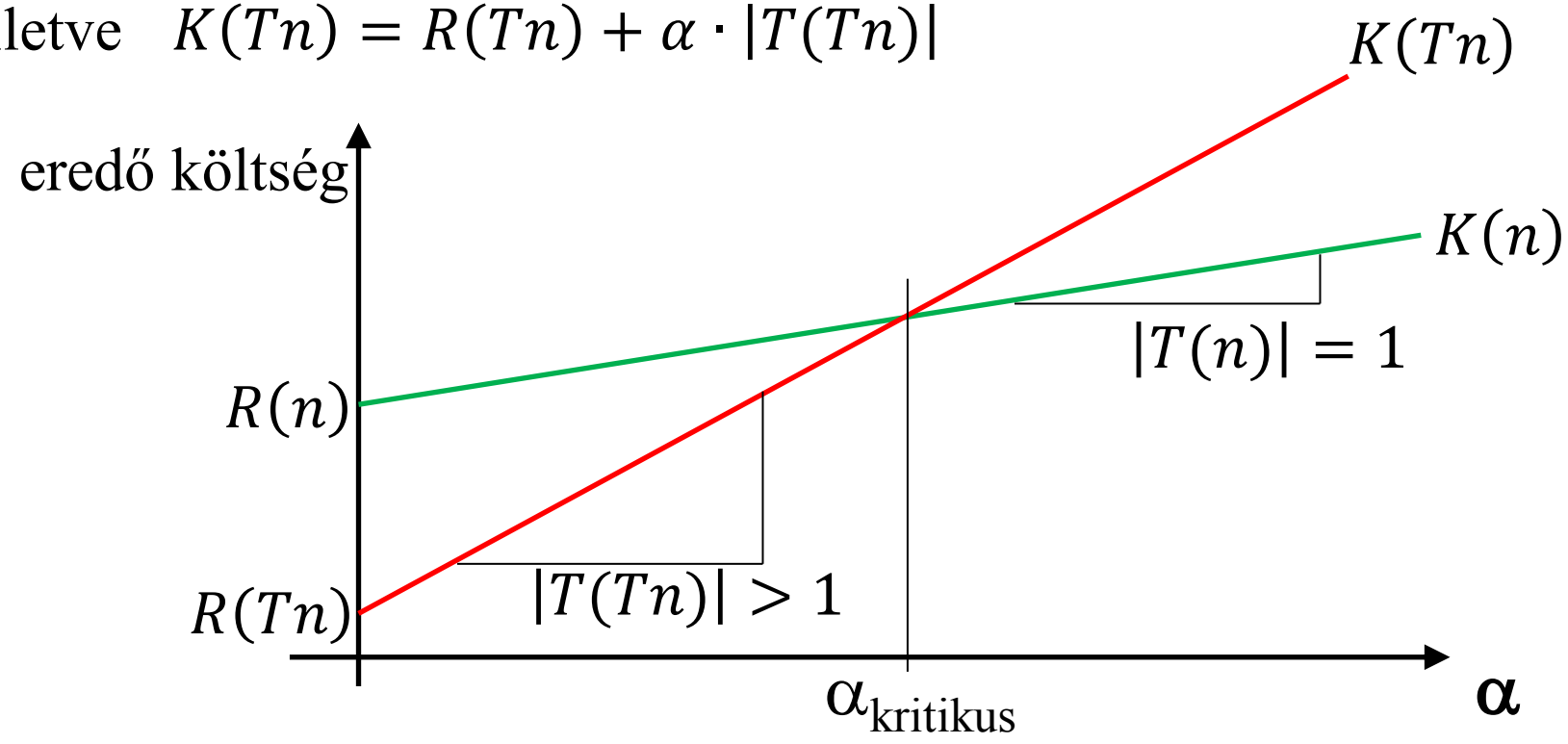
Legyen az összetett, eredő költség (pl. az n-dik csomópontra):

$$K^*(n) = EgységnyiHibaKöltsége \cdot R(n) + EgységnyiKomplexitásKöltsége \cdot |T(n)|$$

Legyen a két költség aránya: $\alpha = \dfrac{EgységnyiKomplexitásKöltsége}{EgységnyiHibaKöltsége}$

Így $K(n) = \dfrac{K^*(n)}{EgységnyiHibaKöltsége} = R(n) + \alpha \cdot |T(n)|$

illetve $K(Tn) = R(Tn) + \alpha \cdot |T(Tn)|$



$\alpha_{kritikus}$ amikor mindegy, hogy levágjuk-e a részfát, az összetett költség azonos az *n*-edik csomópontra, mint levélre, és az *n*-dik csomópontból kiinduló *Tn* részfára

$$R(n) + \alpha_{kritikus} \cdot |T(n)| = R(Tn) + \alpha_{kritikus} \cdot |T(Tn)|$$

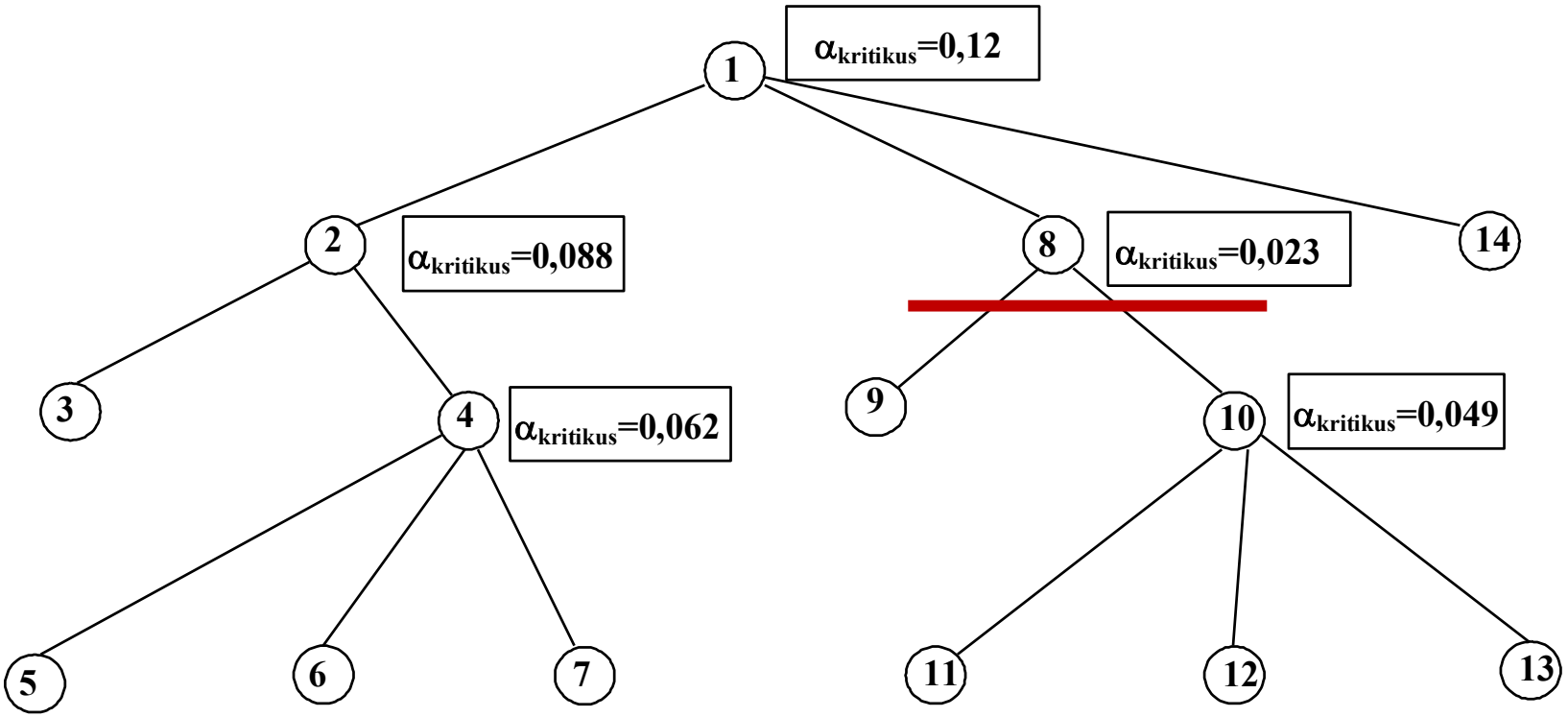
$$\alpha_{kritikus} = \frac{R(n) - R(Tn)}{|T(Tn)| - |T(n)|} = \frac{R(n) - R(Tn)}{|T(Tn)| - 1}$$

1. Ha $\alpha=0$, akkor a komplexitásnak nincs ára, soha nem érdemes visszametszeni
2. $0 < \alpha < \alpha_{kritikus}$, akkor még mindig jobb nem visszametszeni
3. Ha $\alpha = \alpha_{kritikus}$, akkor mindegy, hogy visszavágjuk-e
4. Ha $\alpha_{kritikus} < \alpha$, akkor érdemes visszavágni, olcsóbb abbahagyni az *n*-dik levélnél, nem kifejteni az innen induló részfát

Gondoljuk végig, hogy mi történik, ha α–t folyamatosan növeljük 0-tól indulva!

➡ amelyik csomópont $\alpha_{kritikus}$ értékét először érjük el, ott érdemes ***leginkább*** metszeni a fát!

(Persze ha a hiba nem nő túl nagyra, ha a hiba már elfogadhatatlan, de a komplexitás is túl nagy még, akkor újra kell gondolnunk megoldást.)



A számok légből kapottak (csak a példa kedvéért)