Alkalmazott mesterséges intelligencia (AMI)

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

8. ea. (2023 ősz)

Gépi tanulás: klaszterezés

Előadó: Pataki Béla

a fóliák
Dobrowiecki Tadeusz és
Hullám Gábor anyagainak
felhasználásával készültek

https://www.esrcheck.com/2023/06/05/artificial-intelligence-ai-experts-sign-statement-on-ai-risk/

Pataki Béla

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

pataki@mit.bme.hu,

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

Olyan módszereket kezdtünk vizsgálni, amikor minták, **mintapéldák alapján akarjuk kialakítani a döntési (vagy mérési, illetve szabályozási) 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.

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 – 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.

Az utóbbi néhány előadáson és a mostanin a (2)-vel foglalkozunk

Sokféle tanítható eszközt kitaláltak:

- **neurális hálók**
- Bayes-hálók - tanításukkal nem foglalkoztunk
- kernelgépek
- **döntési fák**
- legközelebbi szomszéd osztályozók
- stb. stb.

A gépi tanulás alapvető fajtái:

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

megerősítéses tanulás az ágens az általa végrehajtott, lépéssorozatokból álló tevékenység csak bizonyos jutalmazását kapja meg, rendszerint nem is minden lépésben (jutalom, büntetés, **megerősítés**)

felügyelet nélküli (nemellenőrzött) tanulás semmilyen információ sem áll rendelkezésünkre a helyes kimenetről

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

Felügyelt tanulás (pl. induktív következtetés):

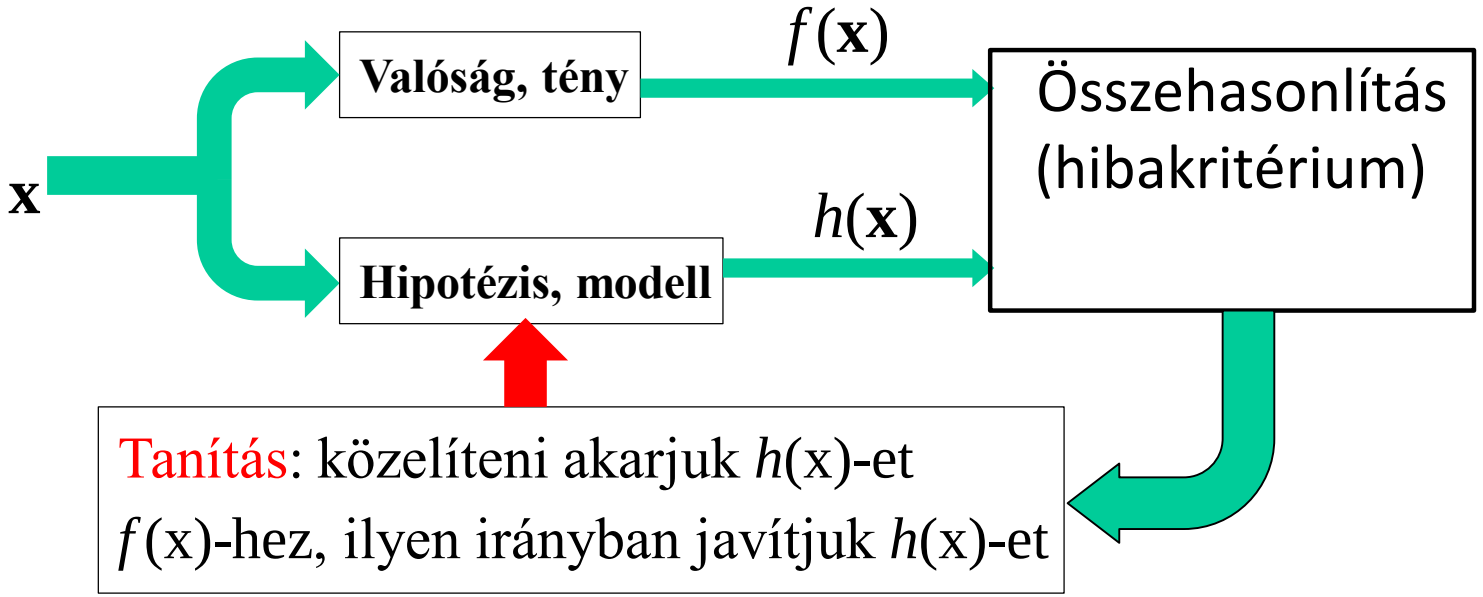
tanulási példa: $\{(\mathbf{x}_k, f(\mathbf{x}_k))\}$ adatpárok halmaza, ahol $f(\mathbf{x})$ ismeretlen

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

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

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

(**általánosító képesség**)

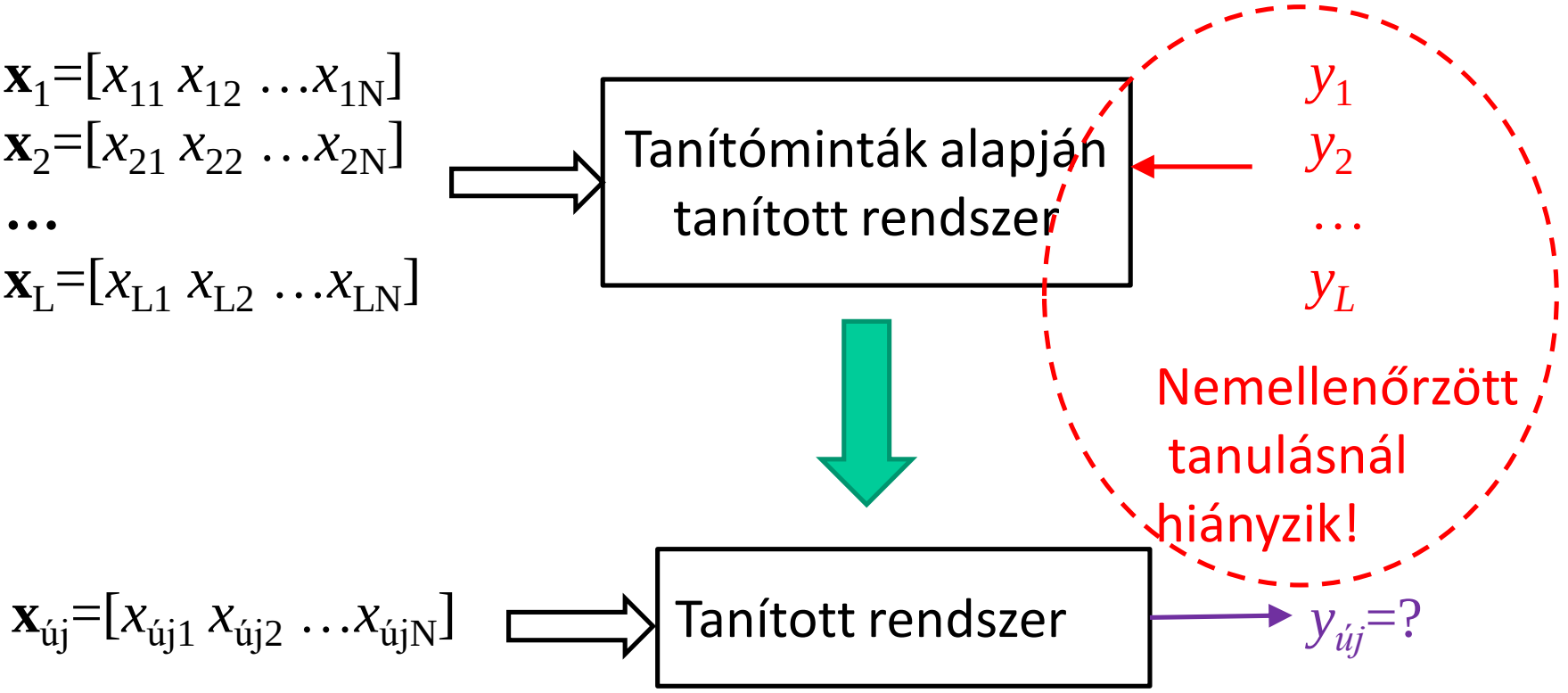


Feladat: *az ismeretlen **f**-re vonatkozó példák egy halmaza alapján (**tanítóhalmaz**), adjon meg egy olyan **h** függvényt (**hipotézist**), amely tulajdonságaiban jól közelíti az **f**-et (amit **teszthalmazon** verifikálunk).*

Klaszterezés (nemfelügyelt/nemellenőrzött tanulás)

Az eseteket szokásos módon az **x** paramétervektor írja le (fényesség, kontraszt, textúraparaméterek, megtakarítás, vagyon, dolgozói létszám, lúdtalpas-e stb.).

Összehasonlítás céljából: a besorolást *ellenőrzött tanulás* (osztályozás) esetén az *y* kimeneti címke adná meg. Itt – *nemellenőrzött tanulás* esetén – nem áll rendelkezésünkre!



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

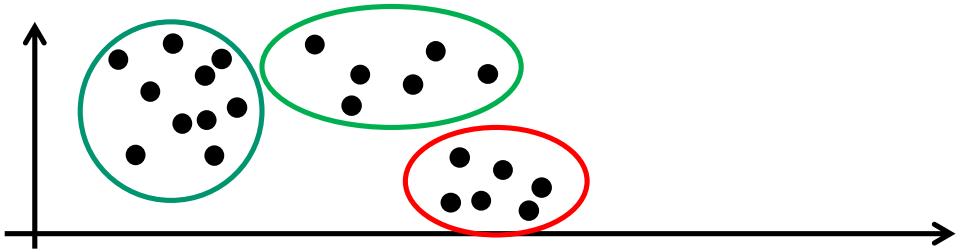
Bayes-tétel: $$P(\mathbf{x}, y_k) = P(y_k|\mathbf{x})P(\mathbf{x}) = P(\mathbf{x}|y_k)P(y_k)$$

$$P(y_k|\mathbf{x}) = \frac{P(\mathbf{x}|y_k)P(y_k)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|y_k)P(y_k)}{\sum_j P(\mathbf{x}|y_j)P(y_j)}$$

*Felügyelt tanulás*nál a tanulóminta-halmazból becsülhetjük $P(\mathbf{x}|y_k)$-t és $P(y_k)$-t $\Rightarrow$ innen becsüljük $P(y_k|\mathbf{x})$-t. ($P(\mathbf{x})$ csak normálásra kell, becsüljük vagy kiadódik)

Nemfelügyelt tanulásnál nem ismerjük az y_k címkéket: csak $P(\mathbf{x})$-t tudjuk becsülni.

Klaszterezés: $P(\mathbf{x})$ olyan becslése, amelyben a mintatérben elkülönülő csoportokra (klaszterekre) bontjuk a mintákat $\rightarrow y_k$???



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:

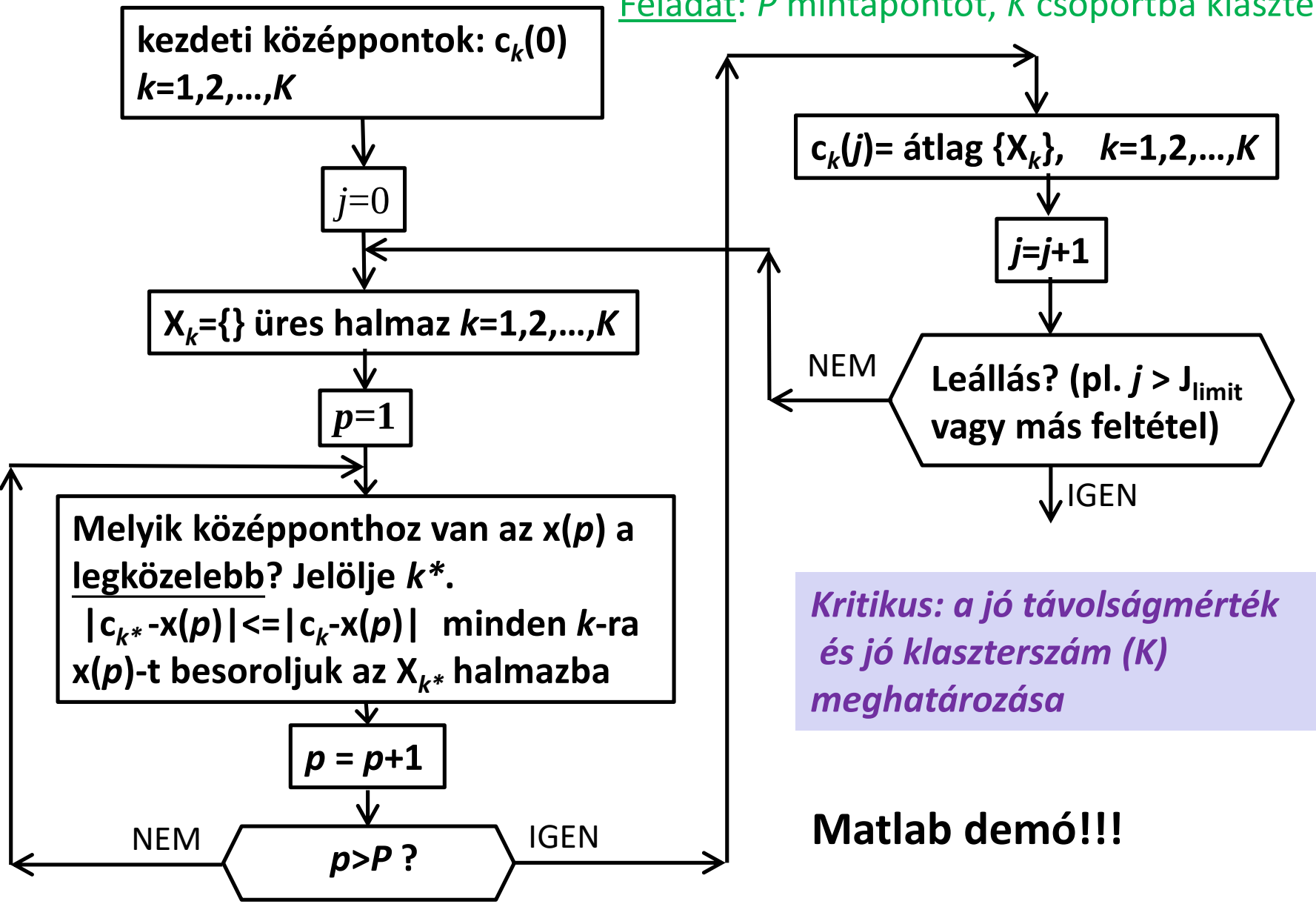
Egy, a mintahalmaz létrehozását (generálását) *magyarázó modell*t használunk. A modell magyarázza az egyes csoportok létrejöttét – a modellparamétereket tanuljuk.

Kritikus a jó modell megtalálása!

Diszkriminatív klaszterezés

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

Feladat: *P* mintapontot, *K* csoportba klaszterezünk



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

Matlab demó!!!

A **generatív klaszterezés** alapötlete

Egy, **az észlelt mintákat magyarázó modellt** keresünk (alkotunk), amely klaszterekben (csoportokban) generálja a mintaeloszlást, és a modell paramétereit becsüljük a minták alapján

Generáló modell:

$$\Theta = \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \\ \dots \\ \Theta_K \end{bmatrix} = ?$$

Minták:

p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8,
.... p_{N-1}, p_N,

Leggyakrabban alkalmazott eset: Gauss eloszlások keveréke

$$P(\mathbf{x}) = \sum_{k=1}^{K} P(k) \cdot P(\mathbf{x}|k)$$

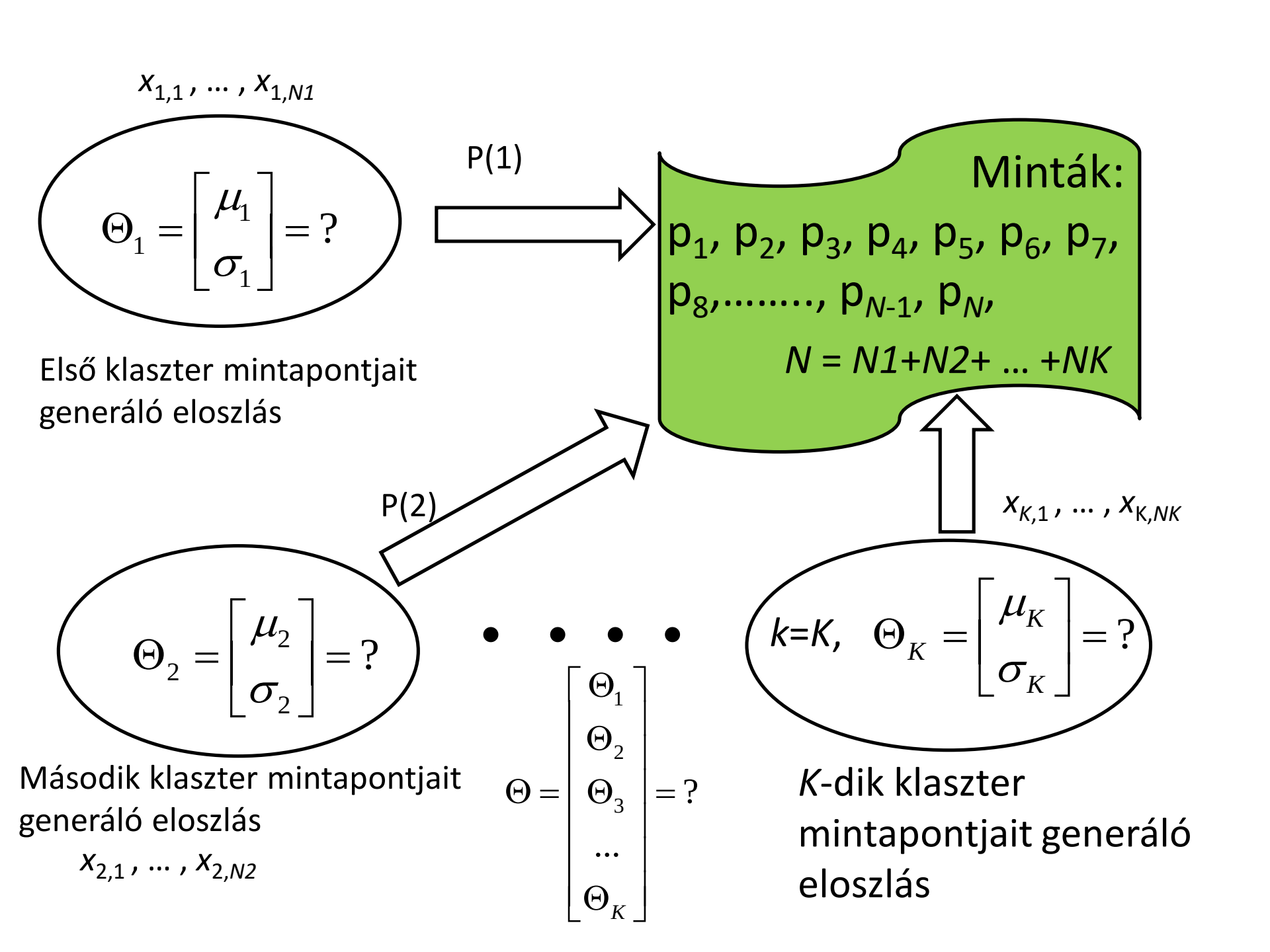
Természetesen a jobboldalon egyik tényező sem ismert.
Becsüljük meg a klaszterek *a priori* valószínűségét $P(k)$-t, és az egyes klasztereket generáló $P(\mathbf{x}|k)$-kat az adatokból.

Az egyszerűség kedvéért az egyes klasztereket generáló eloszlás legyen Gauss (normális), ezt most skalár (x) paraméterrel jellemzett esetre írjuk fel:

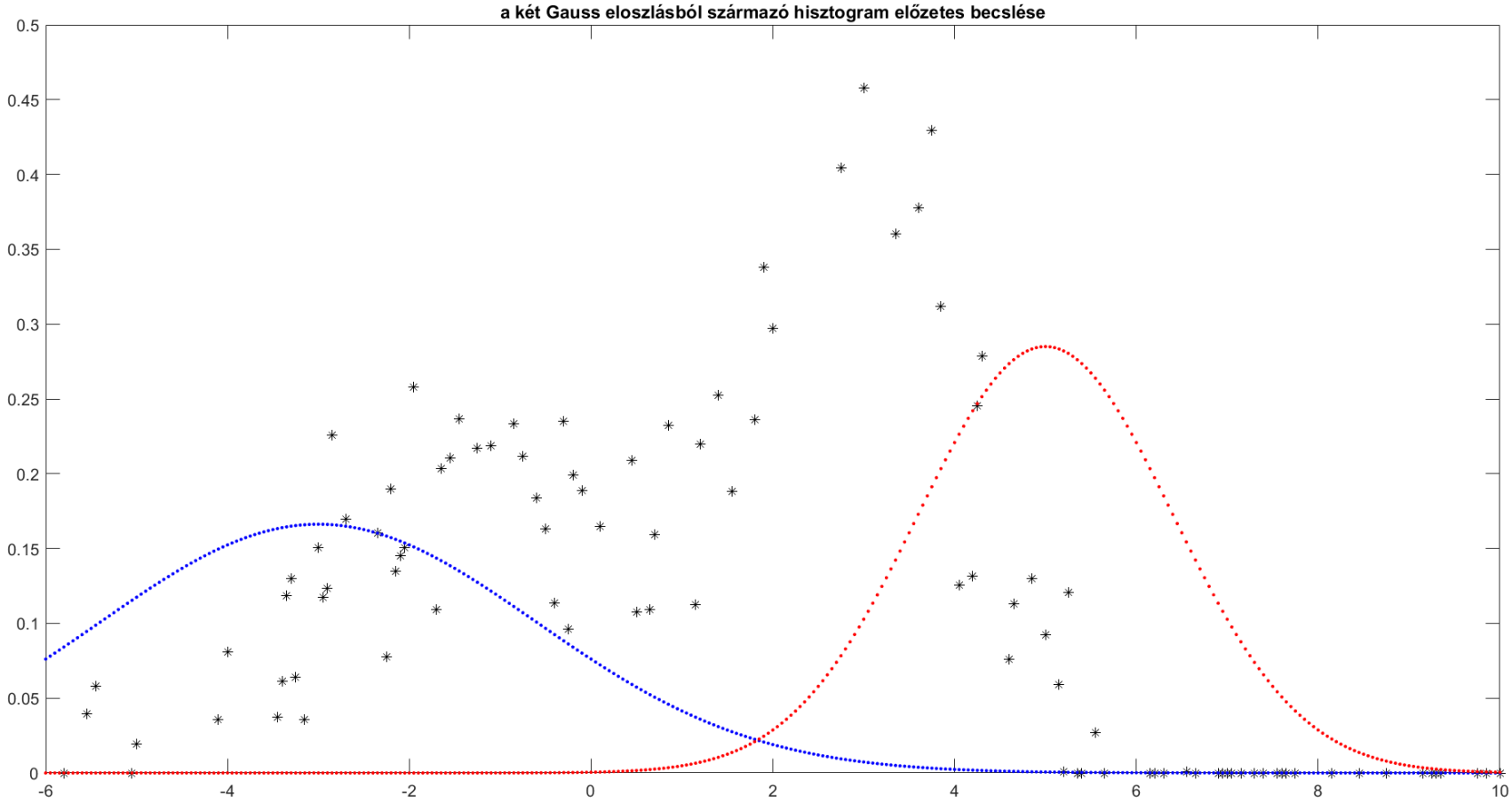
$$P(x|k) = \frac{1}{\sigma_k \sqrt{2\pi}} \cdot e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

Ehhez a részmodellben a μ_k és σ_k paramétereket kell megbecsülnünk, így:

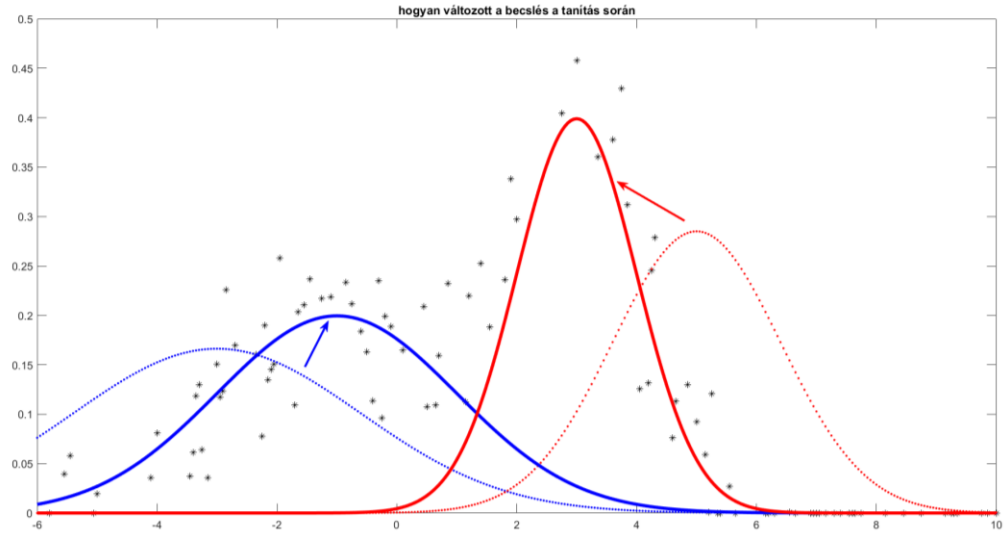
$$\Theta_K = \begin{bmatrix} \mu_K \\ \sigma_K \end{bmatrix}$$


$x_{1,1}, \ldots, x_{1,N1}$
$\Theta_1 = \begin{bmatrix} \mu_1 \\ \sigma_1 \end{bmatrix} = ?$
P(1)
Minták:
$p_1, p_2, p_3, p_4, p_5, p_6, p_7,$
$p_8, \ldots\ldots, p_{N-1}, p_N,$
$N = N1+N2+ \ldots +NK$
Első klaszter mintapontjait generáló eloszlás
P(2)
$x_{K,1}, \ldots, x_{K,NK}$
$\Theta_2 = \begin{bmatrix} \mu_2 \\ \sigma_2 \end{bmatrix} = ?$
$k=K, \quad \Theta_K = \begin{bmatrix} \mu_K \\ \sigma_K \end{bmatrix} = ?$
$\Theta = \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \\ \ldots \\ \Theta_K \end{bmatrix} = ?$
Második klaszter mintapontjait generáló eloszlás
$x_{2,1}, \ldots, x_{2,N2}$
K-dik klaszter mintapontjait generáló eloszlás

Demópélda #1: egy mintaponthalmazra becsüljük a mintasűrűséget (fekete csillagok), és 2 normális eloszlásból álló generatív modellt illesztünk. A kezdeti modellben rossz helyen vannak a normális eloszlások, a szélességük és a magasságuk se megfelelő. A generatív klaszterező eljárás olyan irányba mozgatja és méretezi őket, hogy a lehető legjobban illeszkedjenek a mintahalmazra:

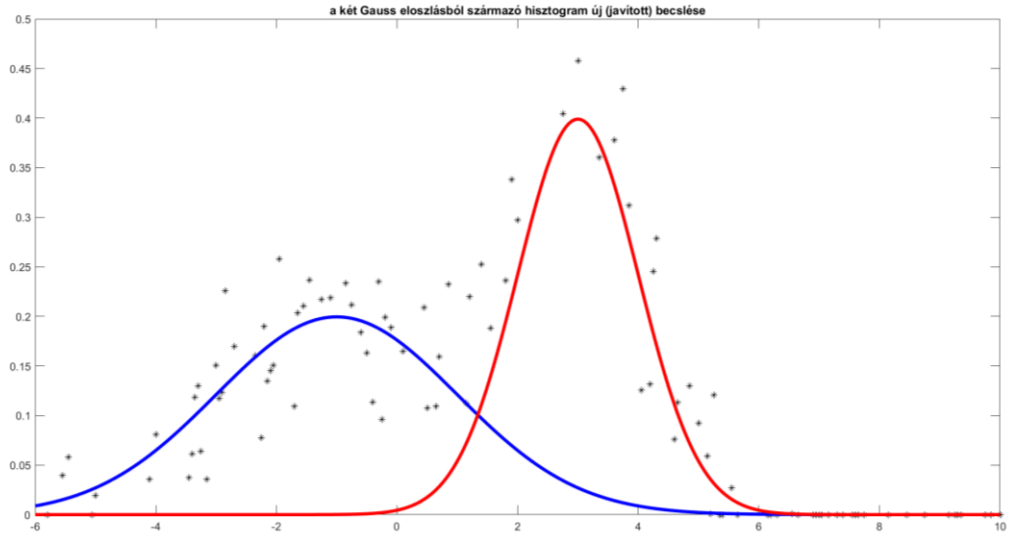


Demópélda #1: A kezdeti modellben rossz helyen vannak a normális eloszlások, a szélességük és a magasságuk se megfelelő. A generatív klaszterező eljárás olyan irányba mozgatja és méretezi őket, hogy a lehető legjobban illeszkedjenek a mintahalmazra:



Változtatja a modellünkben szereplő Gauss függvények magasságát és szélességét

Az új becslés jobban illeszkedik a pontokra!



Demópélda #2:

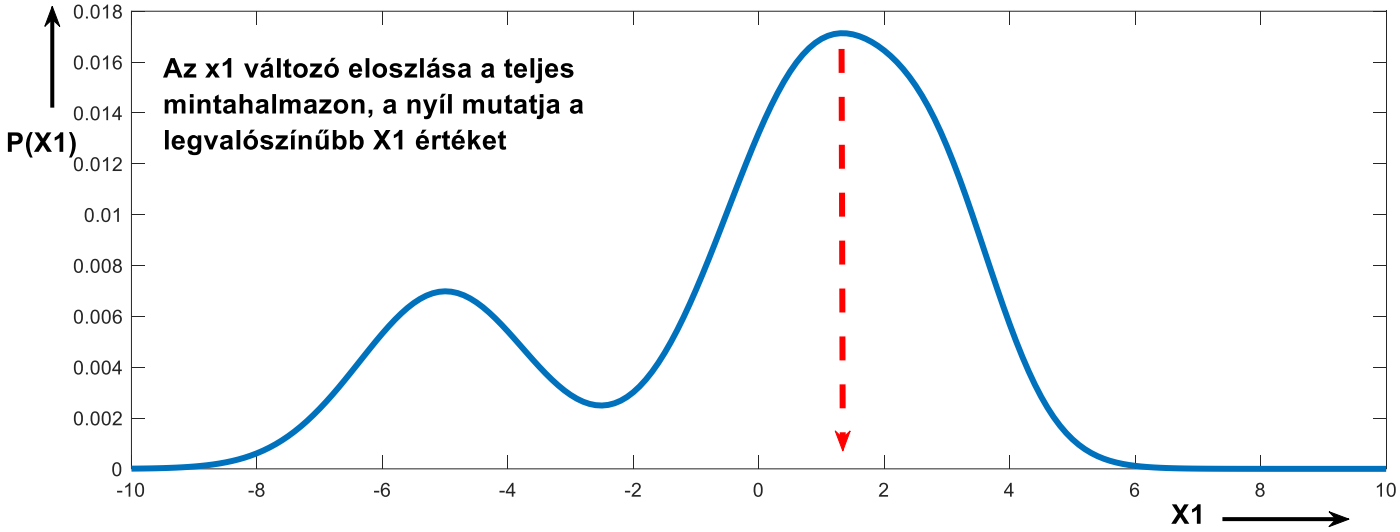
Matlab demó

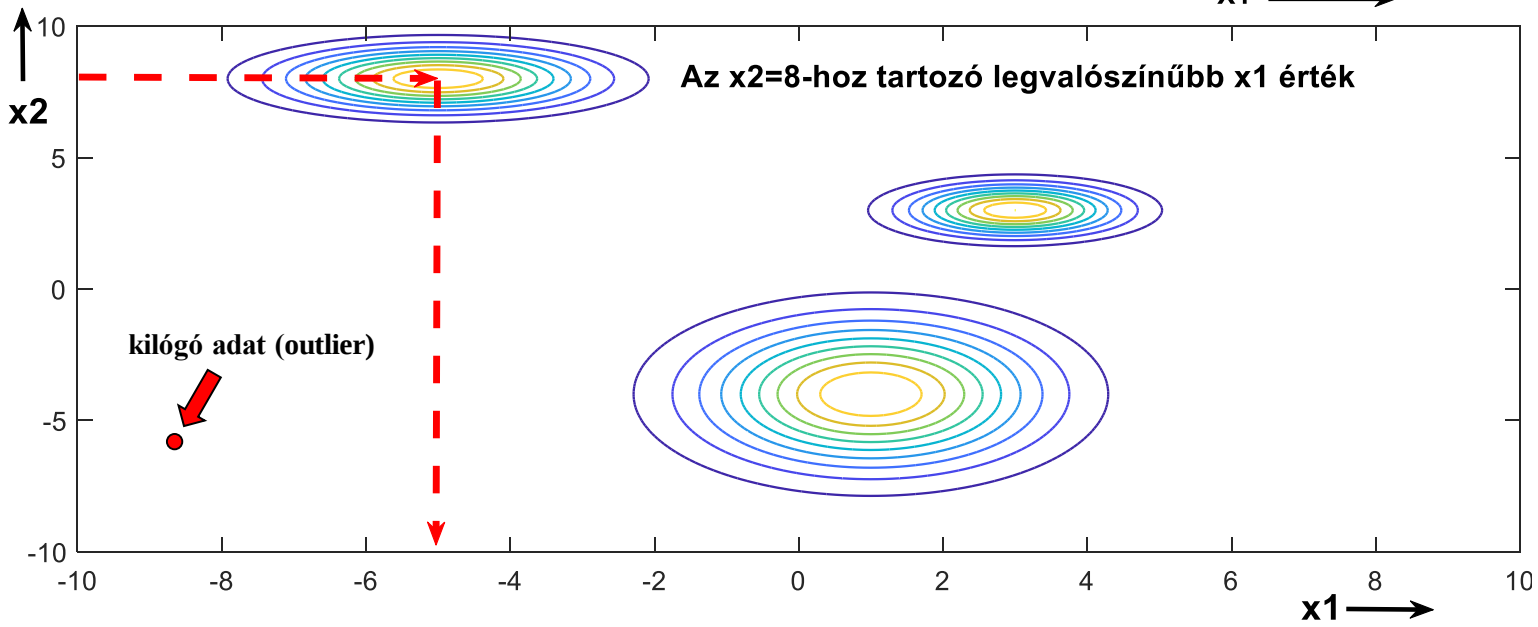
Miért lehet jó klaszterezni a mintákat?

- Találhatunk érdekes csoportokat az adatainkban – pl. képszegmentálásnál (előző demópélda)
- Segíthet a kilógó (outlier) adatok felderítésében – ha mindegyik kialakuló klasztertől távoli adat érkezik, *valószínűleg* hibás (kilógó, outlier)
- Megmutathatja, 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 #3 (Hiányzó adat pótlása)

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

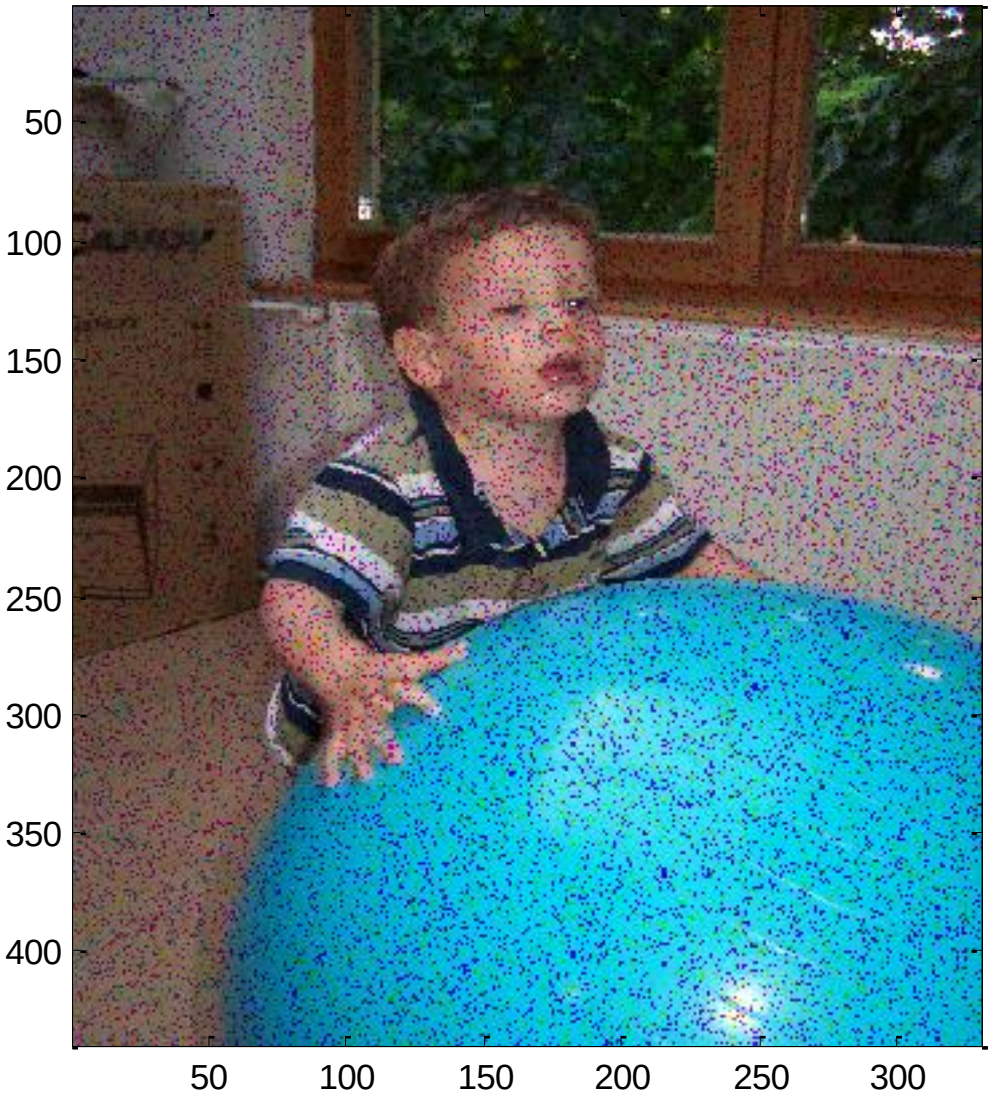




Demópélda #4

Eredeti kép

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



A képpontok 20%-ánál a 3 színkomponenesből 1-1 elveszett, 0 van helyette.

Balról-jobbra:

1. a hibás kép (20% pixel egyik komponense hiányzik),
2. a paraméterek ***globális átlag***ával javítottuk a hiányzó komponenseket,
3. ***klaszterenkénti*** átlagparaméterekkel javítottunk (a maradék két komponens alapján eldöntjük, hogy valószínűleg melyik klaszterbe tartozik, és annak a klaszternek a színátlagát vesszük)