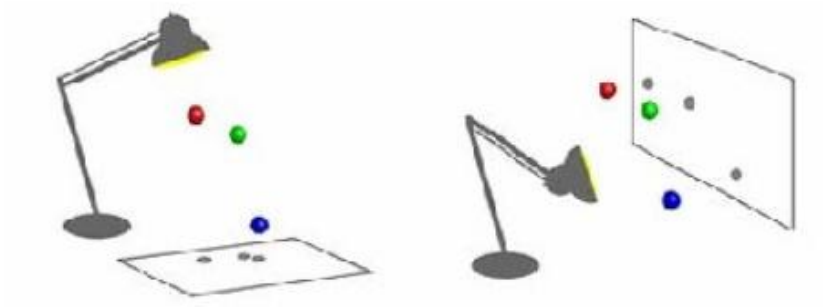
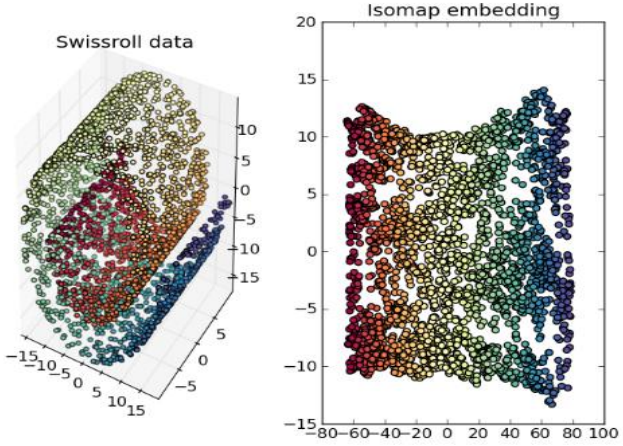
Dimenziócsökkentés



Low Variance Projection

Maxima Variance Projection



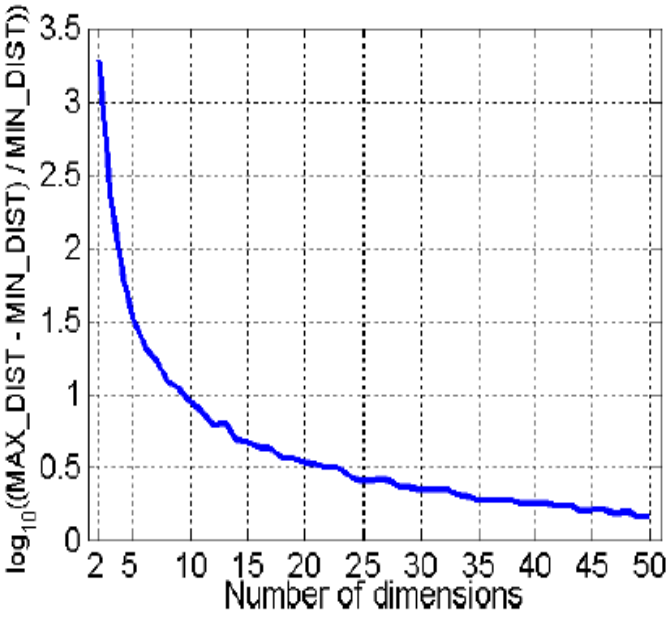
Fogarassyné Vathy Ágnes

vathy.agnes@mik.uni-pannon.hu

A dimenzionalitás átka

Dimenzió probléma:

- Amikor a dimenzió nő az objektumok (pontok) egyre ritkábbak lesznek a térben, ahol elhelyezkednek.
- Az objektumok (pontok) közötti sűrűség – melyek alapvetőek csoportosításnál és kiugró adatok meghatározásánál – csökken.
- Ahhoz, hogy az adatok egyforma sűrűségűek legyenek, *a dimenzióval exponenciálisan növekvő mennyiségű adattal kell rendelkeznünk*.



- **Generáljunk 500 véletlen pontot**
- **Számítsuk ki az összes pontpár közötti távolság maximuma és minimuma különbségét**

Dimenziócsökkentés

Cél:

- Elkerülni a dimenzió problémát.
- Csökkenteni az adatbányászati algoritmusokhoz szükséges időt és memóriát.
- Segíteni az adatok könnyebb megjelenítését.
- Segíteni a hiba csökkentését és a lényegtelen jellemzők meghatározását, majd elhagyását.
- Egymással összefüggő adatok kiküszöbölése.

Dimenziócsökkentés

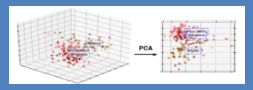
Dimenziócsökkentési módszerek

Jellemzők szelektálása (Feature selection)

- Előrelépéses kiválasztás (Sequential Forward Selection - SFS)
- Visszalépéses eliminálás (Sequential Backward Selection SBS)
- Random Forest
- Low Variance Filter
- High Correlation Filter
- Missing Value Ratio
- …

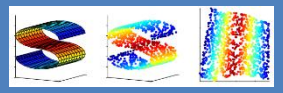
Objektumtér transzformálása (Feature extraction)

Algebrai (Komponens alapú)

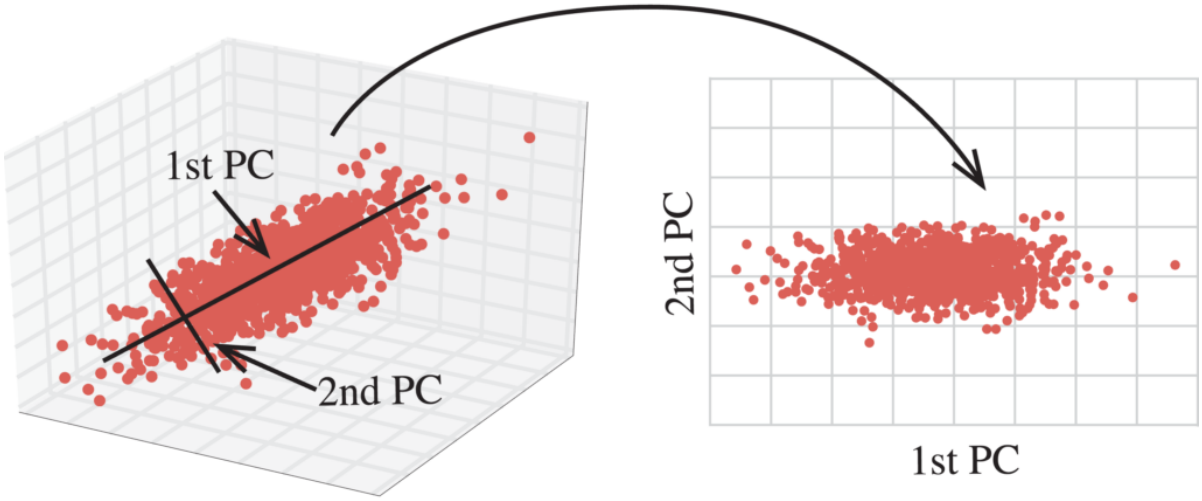


- Faktoranalízis
- Főkomponens analízis (PCA)
- Independent Component Analysis (ICA)
- …

Manifold learning (projection based)



- Kernel PCA
- MDS
- Isomap
- LLE
- tSNE
- UMAP
- …



DIMENZIÓCSÖKKENTÉS OBJEKTUMTÉR TRANSZFORMÁCIÓVAL

Algebrai transzformáció

Johnson-Lindenstrauss lemma

A *magas dimenzionalitású pontok alacsony dimenzionalitású térbe* történő *kis torzítású beágyazására* vonatkozik.

Johnson-Lindenstrauss lemma: Tetszőleges *n* számosságú magas dimenzionalitású pont beágyazható az alacsony dimenzionalitású térbe oly módon, hogy *az adatpontok közti távolság nagyrészt megőrizhető*.

- az adatpontok geometriája megmarad

Lineáris objektumtér transzformáció

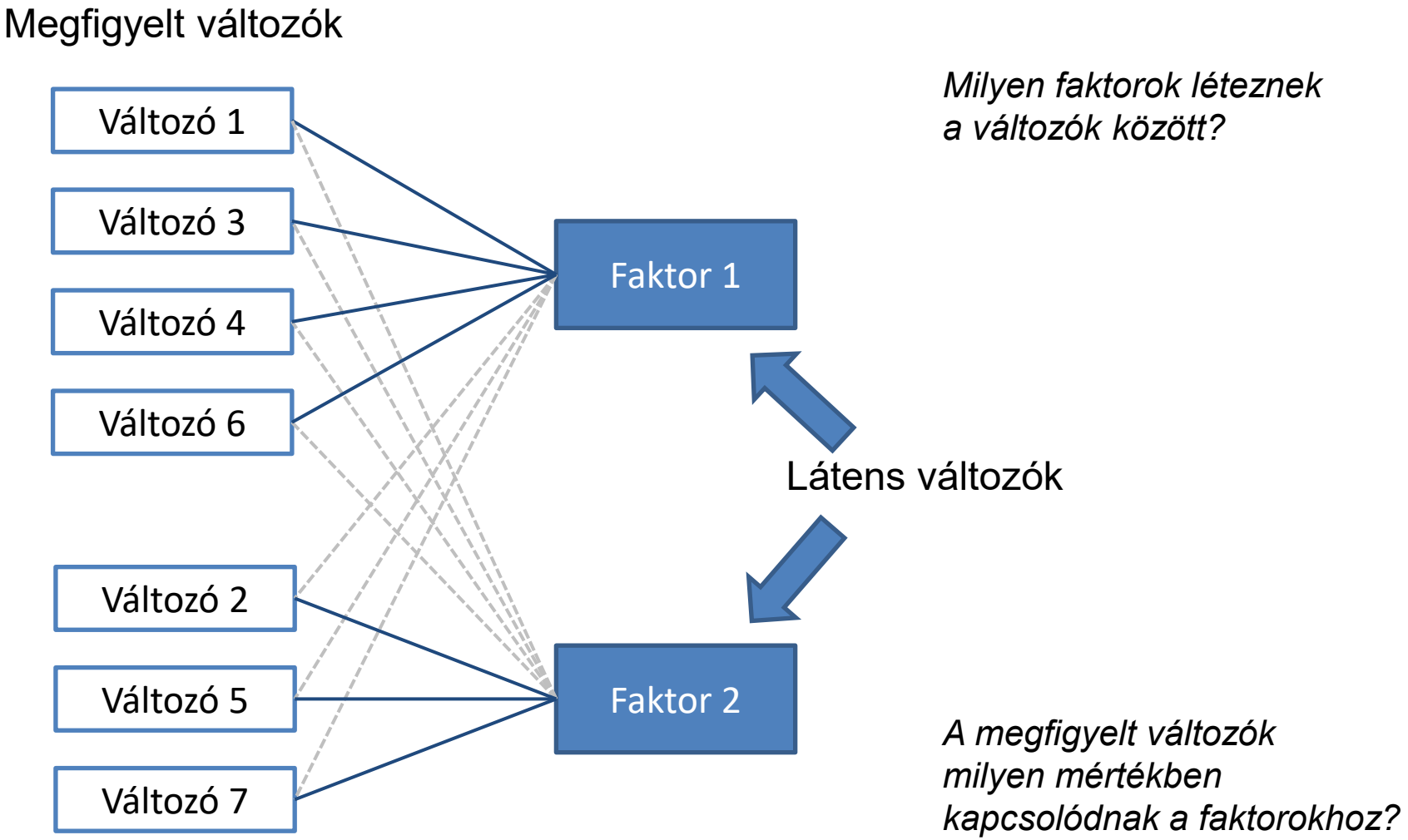
- Faktoranalízis (FA)
- Főkomponens analízis (PCA)

Faktoranalízis (FA)

- Faktoranalízis: a p dimenzió leképezése $k \ll p$ dimenzióba úgy, hogy *az egymással nagymértékben korreláló* változókat 1 csoportnak tekintjük és 1-1 csoport lesz 1-1 faktor (új, látens változó)
  - a faktoroknak nincs fizikai jelentésük, közvetlen nem figyelhetők meg, nem mérhetők, létezésüket csak feltételezzük az eredeti változók kapcsolatai alapján
  - a faktorok az eredeti változók lineáris kombinációjaként számíthatók
  - egy-egy változót egy-egy faktor eltérő súllyal befolyásol

  - speciális esete: Főkomponens elemzés (PCA)

Faktoranalízis (FA)



Faktoranalízis (FA)

FA modell:

$$x_i = l_{i1}F_1 + l_{i2}F_2 + \cdots + l_{ik}F_k + V_iU_i$$

Kifejtve:

$$x_1 = l_{11}F_1 + l_{12}F_2 + \cdots + l_{1k}F_k + V_1U_1$$
$$x_2 = l_{21}F_1 + l_{22}F_2 + \cdots + l_{2k}F_k + V_2U_2$$
$$\cdots$$
$$x_p = l_{p1}F_1 + l_{p2}F_2 + \cdots + l_{pk}F_k + V_pU_p$$

Mátrix alakban:

$$\boldsymbol{x} = \boldsymbol{LF} + \boldsymbol{\epsilon}$$

- $\boldsymbol{F}$: faktor mátrix
- $\boldsymbol{L}$: faktorsúly (loading) mátrix
- $\boldsymbol{\epsilon}$: error term

X_i: i-dik megfigyelt változó

x_i: i-dik **standardizált változó**, amelyet úgy kapunk, hogy a változóból kivonjuk az átlagát

$$x_i = X_i - \bar{X}_i$$

p: a változók száma (eredeti, megfigyelt változók)

F_j: a j-dik **közös faktor**

k: a közös faktorok száma

l_{ij}: a j-dik faktor súlya az i-dik változóban (loading)

U_i: az **egyedi faktor** az i-dik változóban

V_i: az egyedi faktor súlya az i-dik változóban

Arra törekszik, hogy egy változóhalmaz közös varianciáját (korrelációját) a lehető legkevesebb faktorral magyarázza.

Faktoranalízis (FA)

- Kommunalitás (h^2):
  - A bevezetett faktoroknak az eredeti változó szórásának százalékban megvalósított értékelését mutatja.
    - Minél nagyobb a kommunalitás (maximum 1 lehet), annál jobb a választott faktormodell.

$$h_i^2 = \sum_{j=1}^{k} l_{ij}^2$$

- Specificitás (u^2):
  - a varianciának az a része, amit az egyedi faktor ad

$$1 = h^2 + u^2$$

Variable	Factor loadings F_1	Factor loadings F_2	Communality h_i^2	Specificity u_i^2
x_1	0.470	0.734	0.759	0.241
x_2	0.510	0.704	0.756	0.244
x_3	0.481	-0.258	0.298	0.702
x_4	0.888	-0.402	0.949	0.051
x_5	0.956	-0.233	0.968	0.032
Variance explained	2.413	1.317	$\Sigma h_i^2 = 3.730$	$\Sigma u_i^2 = 1.270$
Percentage	48.3	26.3	74.6	25.4

$$x_3 = 0.481F_1 - 0.258F_2 + e_3$$

Faktoranalízis (FA)

FA modell:

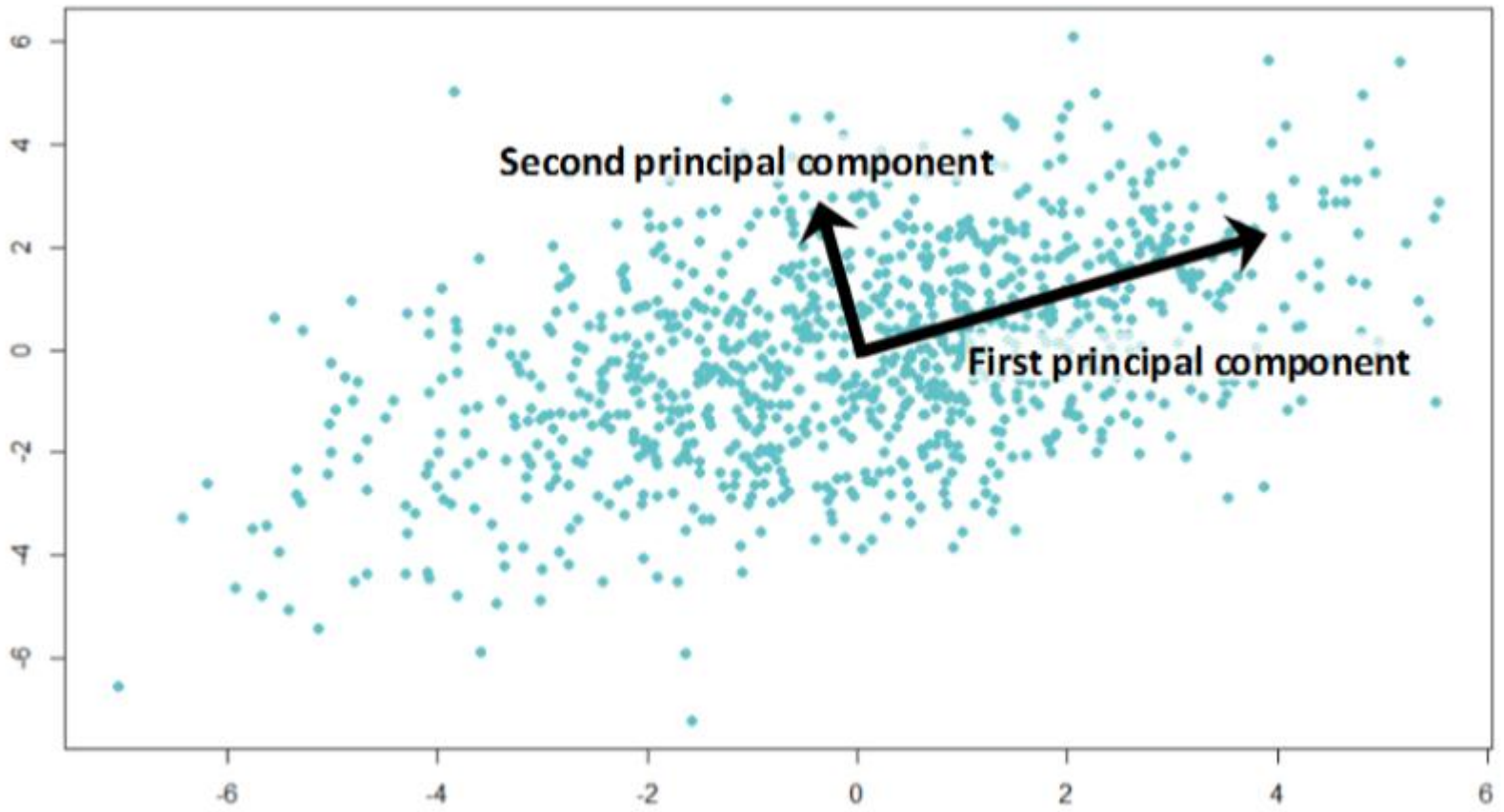
$$x_i = l_{i1}F_1 + l_{i2}F_2 + \cdots + l_{ik}F_k + V_iU_i$$

$F_1, F_2, \ldots, F_k, V_i$ –k függetlenek (nem korrelálnak)!

- a közös faktorok ($F_i, i = 1, \ldots, k$) kifejezhetők a megfigyelt változók lineáris kombinációjaként:

$$F_i = w_{i1}x_1 + w_{i2}x_2 + \cdots + w_{ip}x_p$$

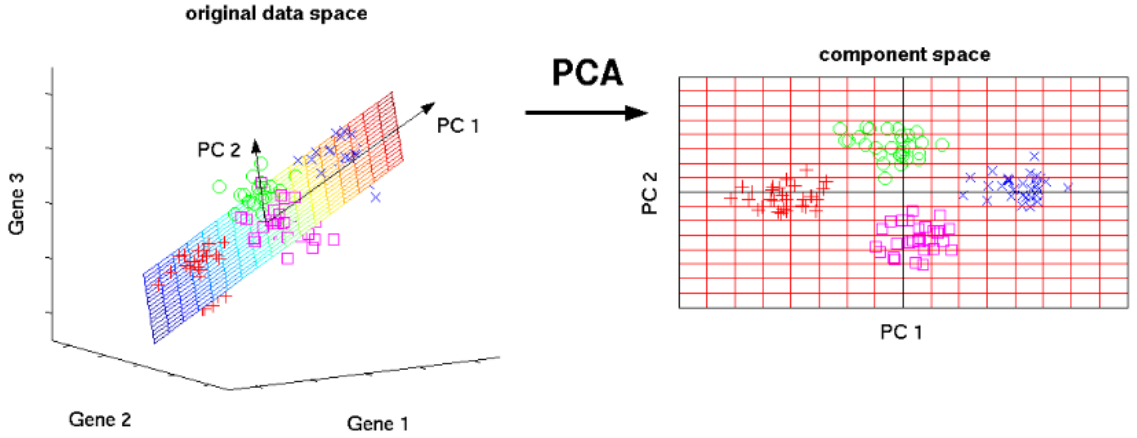
Főkomponens elemzés (PCA)



Főkomponens elemzés (PCA)

Főkomponens elemzés (Principal Component Analysis – PCA)

- Szemléletesen:
  - Az adatokat egy új, kisebb dimenziójú koordináta rendszerbe transzformálja úgy, hogy
  - egy bizonyos projekció szerinti legnagyobb variancia az első koordinátán helyezkedik el, a második legnagyobb variancia a második koordinátán, stb.



  - Tekintsük úgy, mintha egy p dimenziós objektumra egy $k \ll p$ dimenziós ellipszoidot szeretnénk illeszteni úgy, hogy minél többet megőrizzük a varianciából.

Főkomponens elemzés (PCA)

Főkomponens elemzés (Principal Component Analysis – PCA)

- létező változóhalmazból új változókat hoz létre – ezek az új változók a főkomponensek
- oly módon, hogy minél inkább megőrizze az adathalmazban rejlő varianciát

- Főkomponens
  - az eredeti változók lineáris kombinációja
  - az első főkomponens az adathalmaz maximális varianciáját magyarázza
    - a második főkomponens a maradék varianciát próbálja magyarázni és független az első főkomponenstől (merőleges rá)
    - a harmadik főkomponens független az első 2 főkomponenstől (merőleges rájuk) és azon varianciát magyarázza, amelyet nem foglal magában az első kettő főkomponens, és így tovább…
  - a főkomponensek merőlegesek, mivel a kovariancia mátrix sajátvektorai

Főkomponens elemzés lépései

1. Az adatokat 0 köré standardizálása (z-score), mivel a PCA érzékeny a változók skáláira

$$v' = \frac{v - avg_A}{s_A}$$

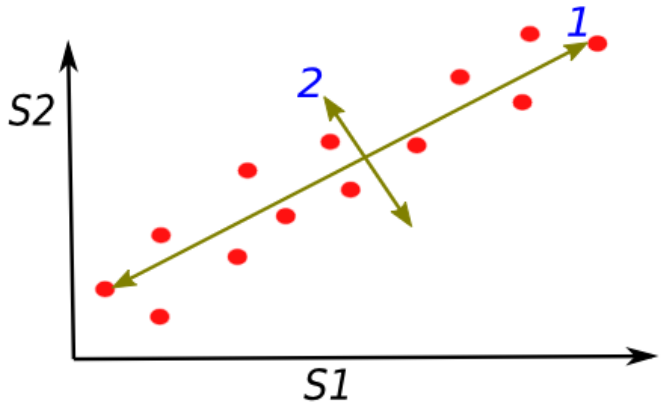
2. Kovariancia-mátrix számítása, amely megmutatja, hogy az egyes változók hogyan mozognak együtt.
   - Ha pozitív, akkor: a két változó együtt növekszik vagy csökken (korrelál)
   - Ha negatív, akkor: az egyik növekszik, amikor a másik csökken (negatív korreláció).
3. A kovariancia-mátrix sajátértékeinek és sajátvektorainak meghatározása és sajátérték szerinti sorba rendezése
   - Sajátvektorok a tengelyek azon irányai, ahol a legnagyobb variancia (ezek leszenek a főkomponensek)
   - A sajátértékek a főkomponensekhez tartozó együtthatók, amelyek megadják az azokban lévő variancia mennyiségét. A legnagyobb sajátérték határozza meg az első főkomponenst, és így tovább…
4. Főkomponens-mátrix kialakítása az első *p* sajátvektor alapján
5. Az eredeti adatok projekciója a kiválasztott főkomponensek által reprezentált tengelyekre: $Z=X \cdot P$, ahol P a főkomponens-mátrix

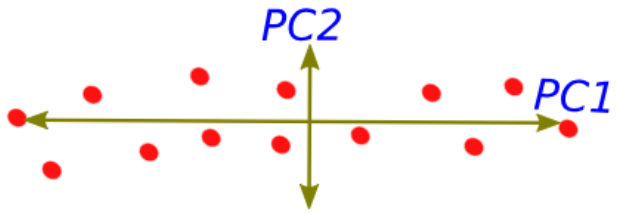
Főkomponens elemzés (PCA)

- Az adathalmazban lévő variancia hány százalékát ragadta meg a k dimenzió?
  - A kiválasztott k sajátvektorhoz tartozó sajátértékeket összegét kell osztani az összes sajátértékek összegével.

PCA egyszerűsítve

- A PCA az adatokat
  - **lineáris transzformáció** alkalmazása révén
  - **új, egymástól független** tulajdonságokba transzformálja,
  - úgy, hogy minél inkább **megőrizze az adathalmaz változatosságát.**





PCA – forgatások

A kapott főkomponensek sok esetben nem értelmezhetők, mert:

- kevert változókat tartalmaznak (minden eredeti változó hozzájárulhat hozzájuk)
- nem mindig feleltethetők meg dimenzióknak – az eredmény nehezen kapcsolható vissza az eredeti dimenziókhoz

Forgatások:

- segítenek az egyes főkomponenseket értelmezhetőbbé tenni azáltal hogy minimalizálják az átfedéseket a komponensek között.
- nem változtatja meg azt, hogy a főkomponensek hány százalékban magyarázzák az eredeti varianciát, de megváltoztatja az egyes komponensek szerkezetét és értelmezhetőségét

PCA – forgatások

2 fő forgatási módszer:

- Ortogonális forgatás (Varimax)
  - a főkomponensek ortogonálisak maradnak, de
  - az egyes komponensek csak néhány változóhoz rendelődnek hozzá
- Ferdeszögű forgatás (Oblimin)
  - megengedi, hogy a főkomponensek ne legyenek merőlegesek (vagyis összefüggjenek)
  - jobban illeszkedik az adatokhoz, mert a főkomponensek természetes módon lehetnem összefüggésben

FA vs.PCA

FA és PCA közti különbség:

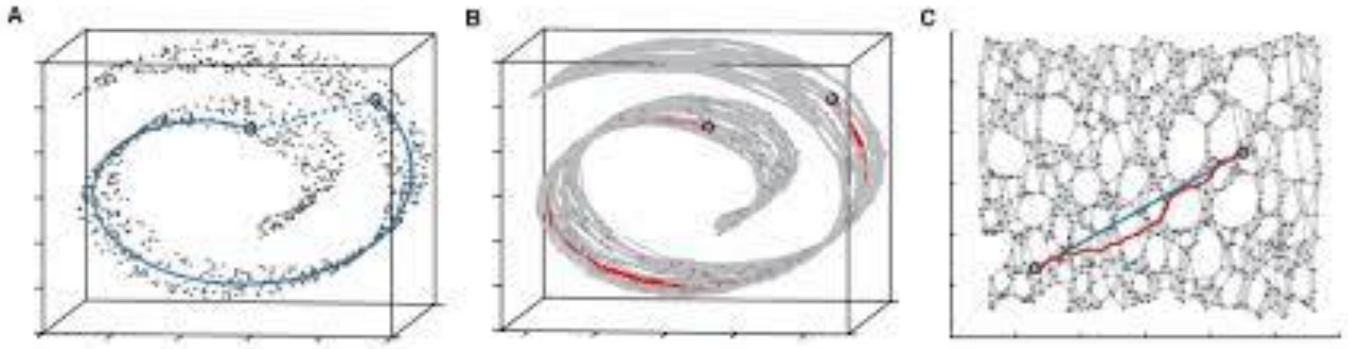
- a célokban és
- az alkalmazott eljárásban rejlik

FA:

- faktoranalízis alkalmazása előfeltételezi, hogy az adathalmazban rejlő változók közötti korreláció magyarázható egy szűkebb számú faktorral, és a cél a faktorok azonosítása
- az így létrehozott faktorszerkezet segíthet megérteni, hogyan kapcsolódnak egymáshoz a változók

PCA:

- azokat a főkomponenseket keresi, amelyek legjobban magyarázzák az adathalmaz varianciáját



DIMENZIÓCSÖKKENTÉS OBJEKTUMTÉR TRANSZFORMÁCIÓVAL

Manifold learning

Többdimenziós skálázás (MDS)

Többdimenziós skálázás (Multidimensional scaling – MDS):

- olyan leképezést keres, ahol az alacsony dimenzionalitású térben az adatpontok távolsága az eredeti dimenzióban értelmezett távolsággal van kapcsolatban
- 2 fajtája:
  - metrikus MDS:
    - távolságmetrikát használ
    - a cél az, hogy az alacsony dimenzionalitású térben számolt távolságmátrix minél hasonlóbb legyen a magas dimenzionalitású térben számolt távolságmátrixhoz
    - amennyiben a távolság az Euklideszi távolság, akkor ekvivalens a PCA-val
    - stress függvénye:

$$\sum_{i<j} d_{ij}(X) - \hat{d}_{ij}(X)$$

ahol $d_{ij}(X)$ az i. és j. adatpont eredeti térben mért távolsága, a $\hat{d}_{ij}(X)$ az i. és j. adatpont alacsony dimenzionalitású térben mért távolsága

  - nem metrikus MDS:
    - csak a távolságok rangjának a megőrzésére koncentrál

Többdimenziós skálázás (MDS)

- Előnyök:
  - nem feltételez linearitást, ezért alkalmazható nemlineáris összefüggés esetén is
  - betekintést enged abba, hogy az objektumok mennyire különböznek egymástól
- Hátrányok
  - a dimenziók jelentésének meghatározása nehéz

Isomap

Isomap:

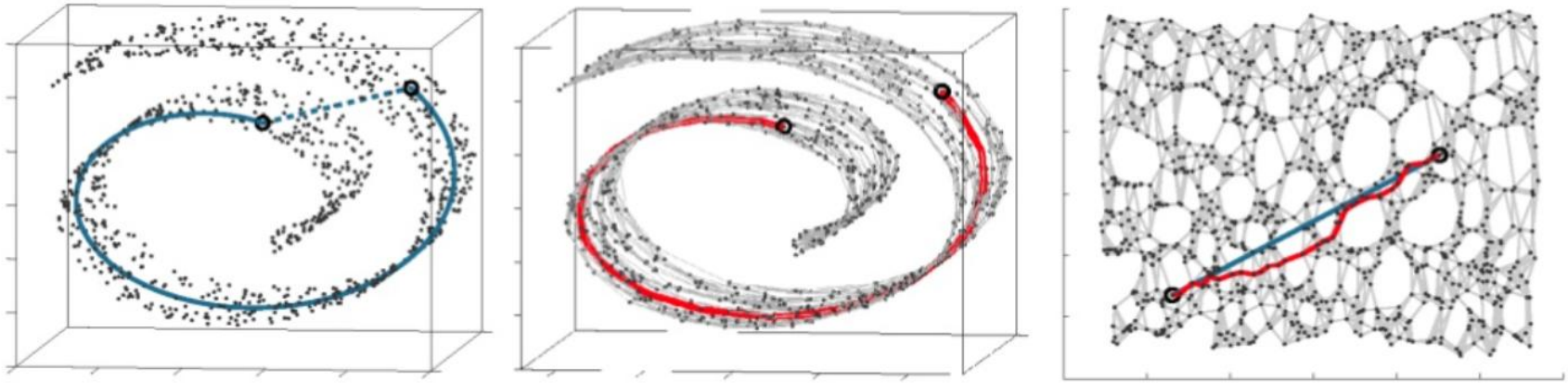
- az MDS nemlineáris verziója
- a távolságot geodéziai távolságként értelmezi
  - geodéziai távolság: a legrövidebb út hossza a gráfban

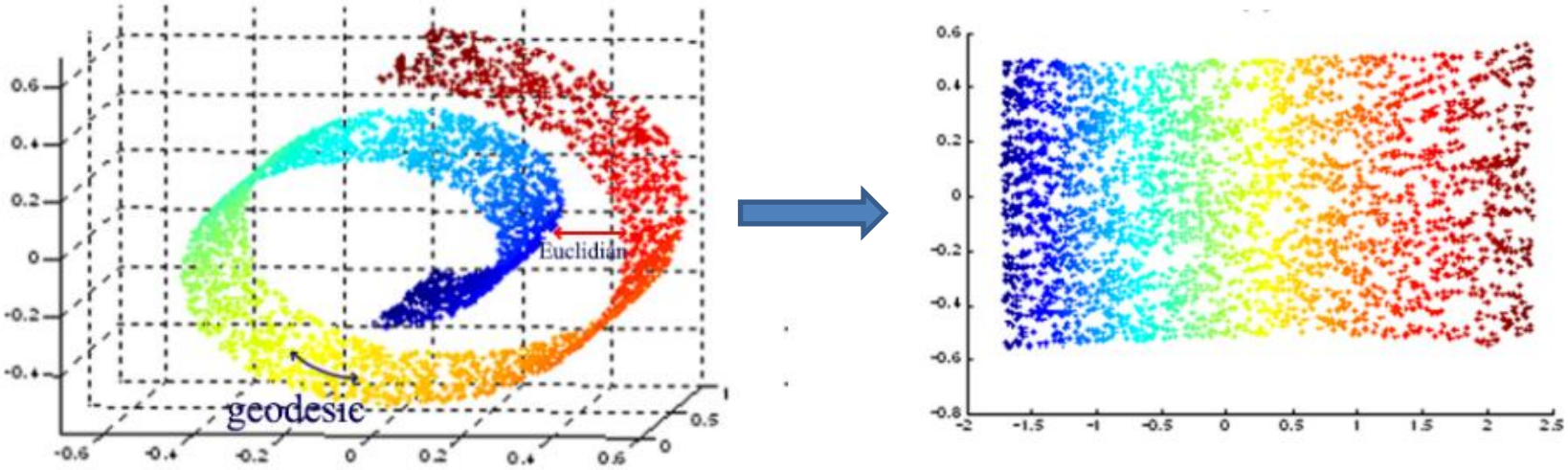
- Algoritmus:
  - keressük meg minden pont legközelebbi szomszédjait (pl. ϵ-sugarú környezeten belüli pontok, vagy k legközelebbi pont) és kössük össze velük => gráf
  - számítsuk ki minden lehetséges adatpontpárra a geodéziai távolságot.
  - végezzünk MDS-t

Isomap

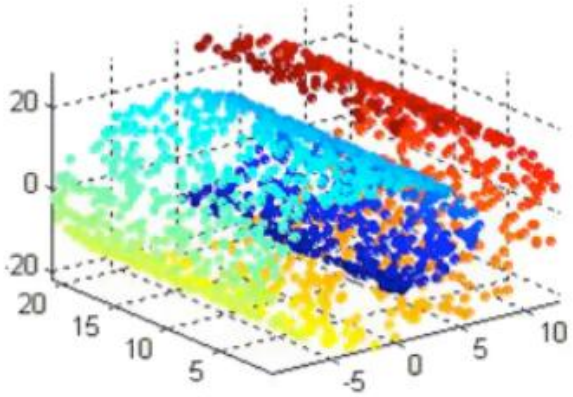
- Előnyök:
  - nemlineáris
  - nem iteratív
  - globális
- Hátrányok:
  - a gráftávolság felülbecsüli a valós távolságot
  - számítása (legrövidebb út) költséges

Isomap

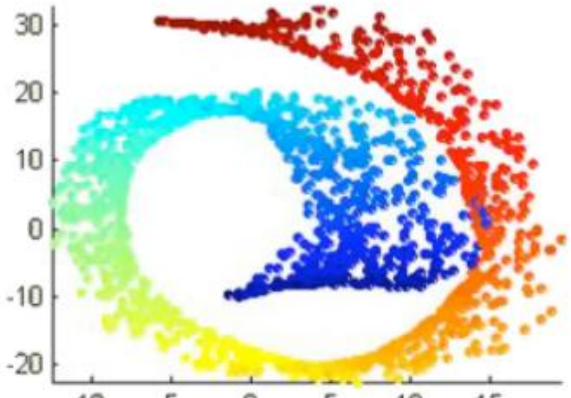




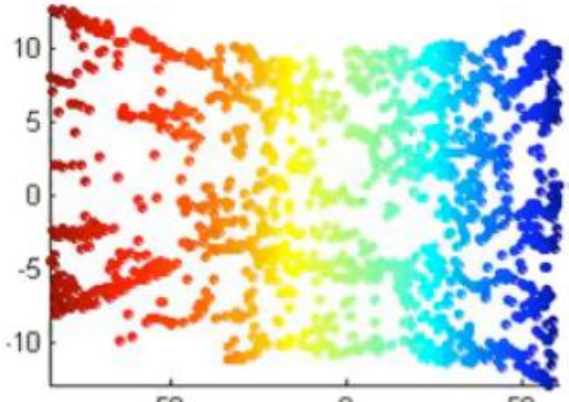
Isomap vs. metrikus MDS



Swiss Roll

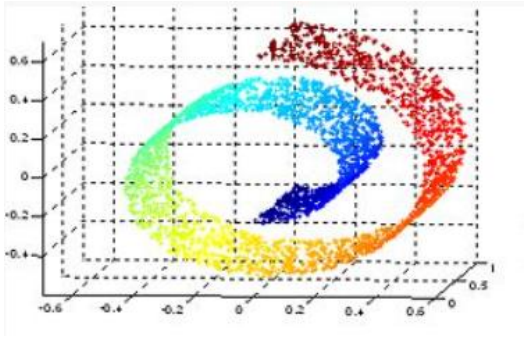


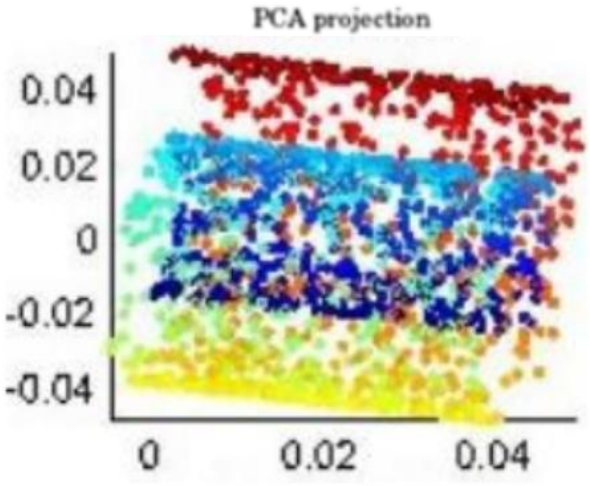
Metric MDS

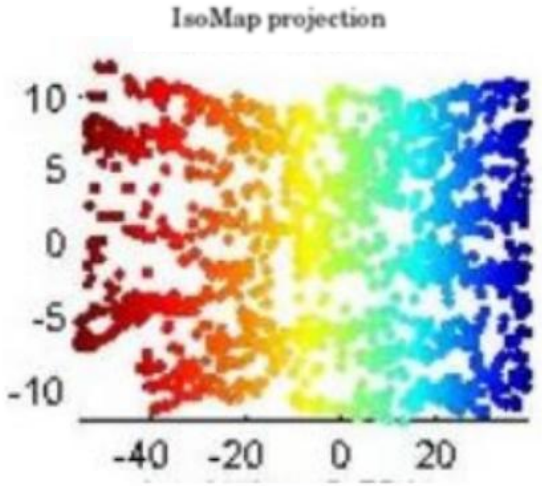


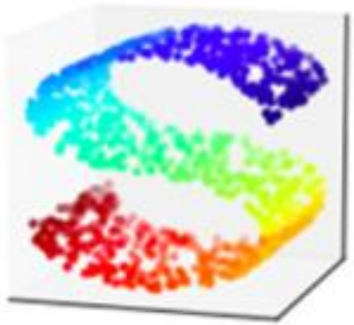
ISOMAP

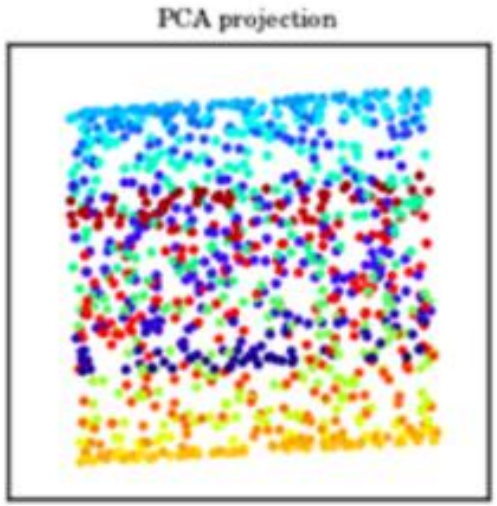
Isomap vs. PCA

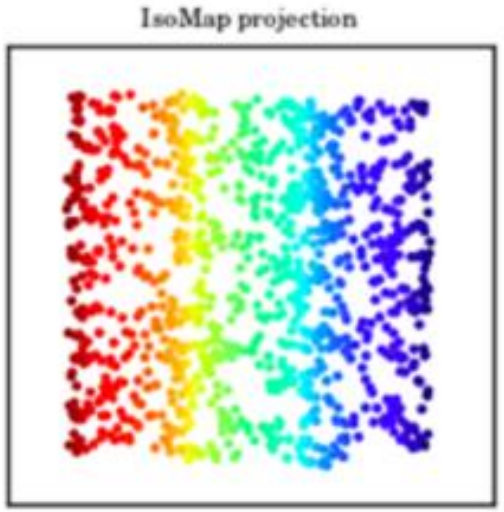








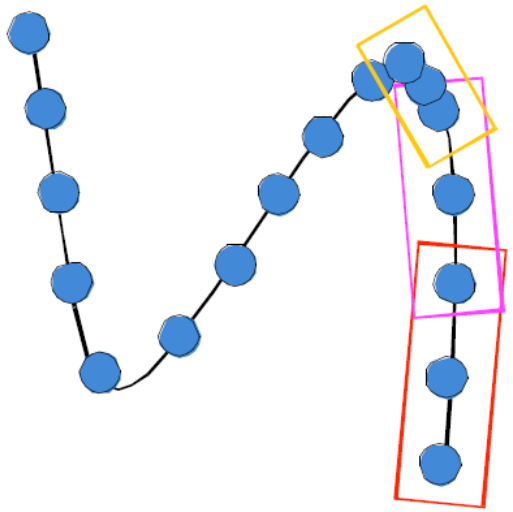




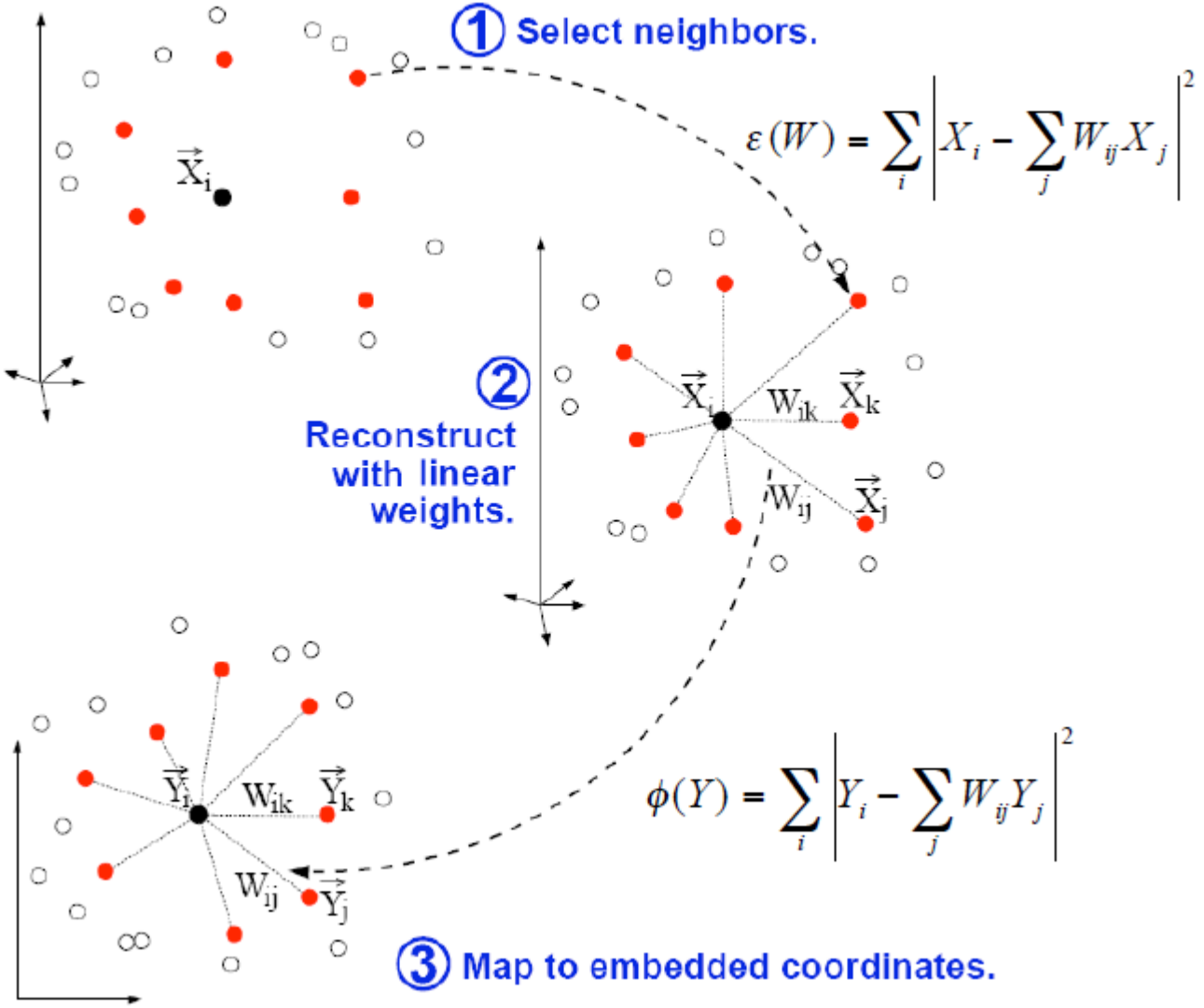
LLE

- LLE (Locally Linear Embedding)
  - az ISOMAP számítási költségét csökkenti
  - csak a lokális szomszédokat veszi figyelembe
  - az adatpontokat a lokális szomszédok lineáris kombinációjaként állítja elő
  - mégis megőrzi a globális struktúrát, mert a lokális utak átfednek

Az LLE alapja az a felismerés, hogy az adathalmazban lévő pontok legtöbbje helyi lineáris összefüggésben van egymással, még akkor is, ha az adathalmaznak összességében nincs lineáris struktúrája.

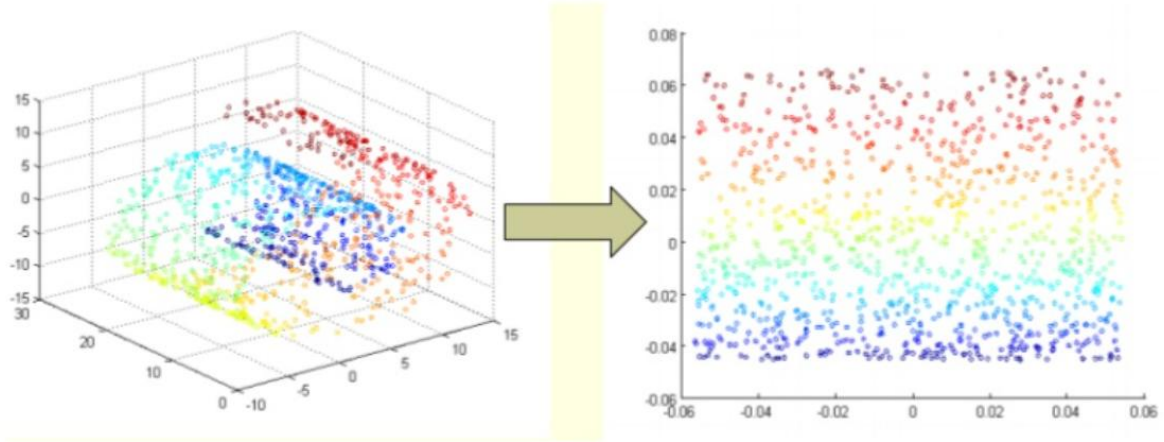


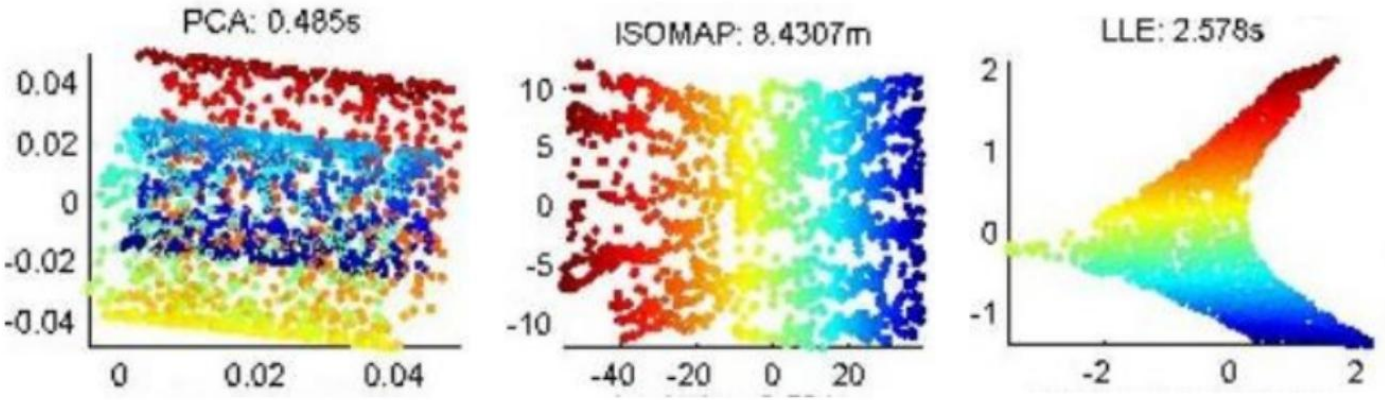
LLE



Source: Saul, 2002

PCA vs. ISOMAP vs. LLE

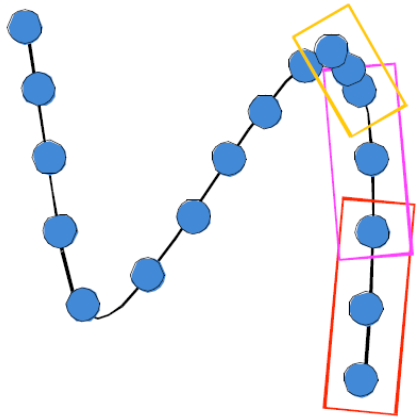




LLE

- LLE előnye:
  - képes megőrizni az adathalmazban rejlő struktúrát anélkül, hogy ismernénk az adathalmazra vonatkozó előzetes információkat
  - kisebb számítási költség

- LLE hátránya:
  - érzékeny lehet az adathalmazban található zajra
  - az adathalmazban lévő túl sok dimenzió esetén az eljárás nehézkes
  - ritka adatok esetén gyenge teljesítmény



SNE

SNE (Stochastic Neighbor Embedding) – sztochasztikus szomszéd beágyazás

- képes megőrizni az adatpontok lokális és globális struktúráját
- az adatpontok hasonlóságának valószínűségét számítja ki mind a magas, mind az alacsony dimenzionalitású térben
- a magas dimenzionalitású térben az euklideszi távolságokat feltételes valószínűséggé konvertálja:

$$p_{j|i} = \frac{\exp\left(-\frac{\left||x_i - x_j|\right|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\left||x_i - x_k|\right|^2}{2\sigma_i^2}\right)}$$

- az alacsony dimenzionalitású térben feltételes valószínűséget számol a következőképpen:

$$q_{j|i} = \frac{\exp(||y_i - y_j||^2)}{\sum_{k \neq i} \exp(||y_i - y_k||^2)}$$

- a valószínűségek számítása után az algoritmus minimalizálja a köztük lévő különbségeket

t-SNE

t-SNE (t-Distributed Stochastic Neighbor Embedding)

t-elosztott sztochasztikus szomszéd beágyazás

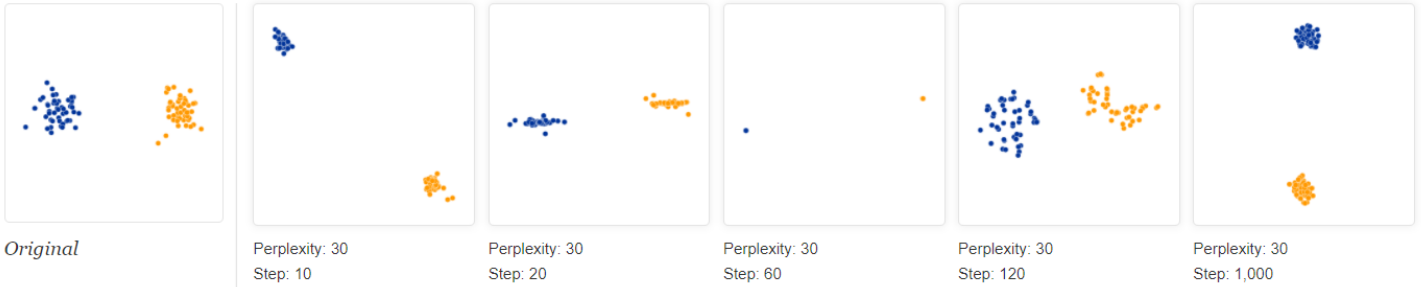
- az SNE változata, amely t-eloszláson alapul
- a magas dimenzionalitású térben ugyanúgy számol
- az alacsony dimenzionalitású térben a valószínűséget a következőképpen számítja:

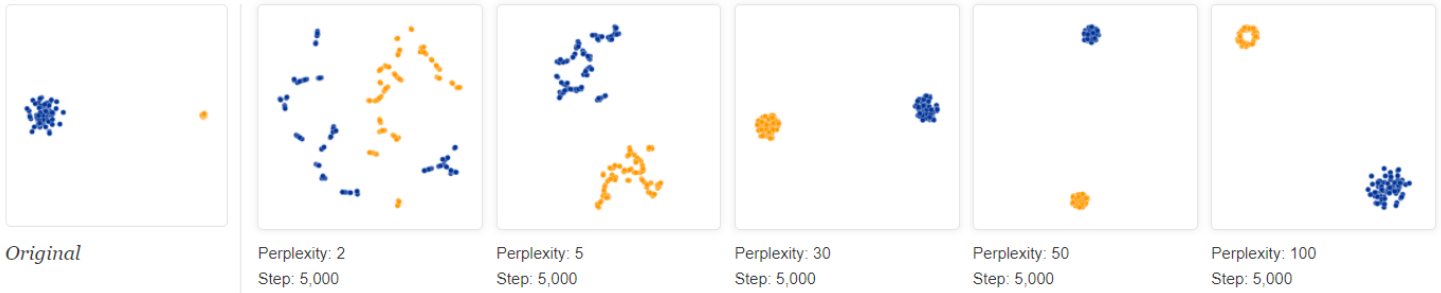
$$q_{j|i} = \frac{\left(1 + ||y_i - y_j||^2\right)^{-1}}{\sum_{k \neq i}(1 + ||y_i - y_k||^2)^{-1}}$$

- Előnyök:
  - a nem lineáris összefüggéseket hatékonyan kezeli
  - a lokális és globális struktúrát is megőrzi (a perplexity paraméter beállításától függően, de elsősorban a lokálisra koncentrál)
- Hátrányok:
  - nagy számítási költség
  - paraméterhangolást igényel (a PCA pl. nem)
  - nem determinisztikus
  - hajlamos random zajban is mintázatot feltárni, ezért többször kell futtatni különféle paraméterbeállításokkal és többször ellenőrizni az eredményt

t-SNE

A t-SNE módszer nagyon érzékeny a paraméterezésre:





https://distill.pub/2016/misread-tsne/

Érdemes elolvasni

- https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/
  - (a t-SNE képlete hibás)