Nem felügyelt tanulás

Pintér Balázs

2018-05-17

Tartalom

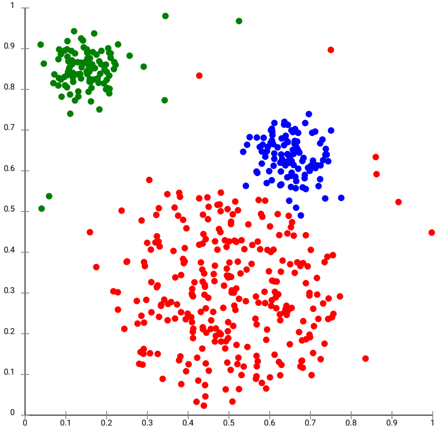
Tartalom

Nem felügyelt tanulás

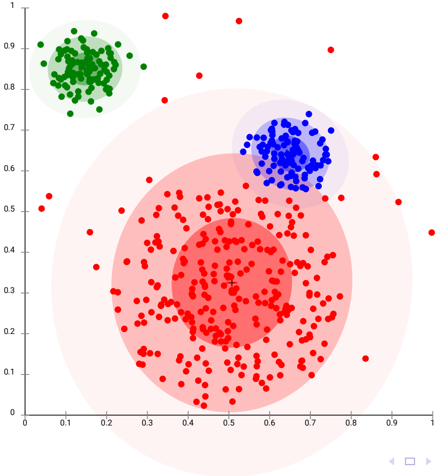
- Felügyelt tanulás: címkézett adatokból tanulunk valamilyen függvényt
- Más megközelítések
  1. **Nem felügyelt tanulás**
  2. Semi-supervised learning
  3. Megerősítéses tanulás
  4. Evolúciós algoritmusok
  5. Neuroevolúció
     `http://www.youtube.com/watch?v=qv6UVOQ0F44`

Tartalom

Példa – sűrűség alapú klaszterezés



Példa – eloszlás alapú klaszterezés



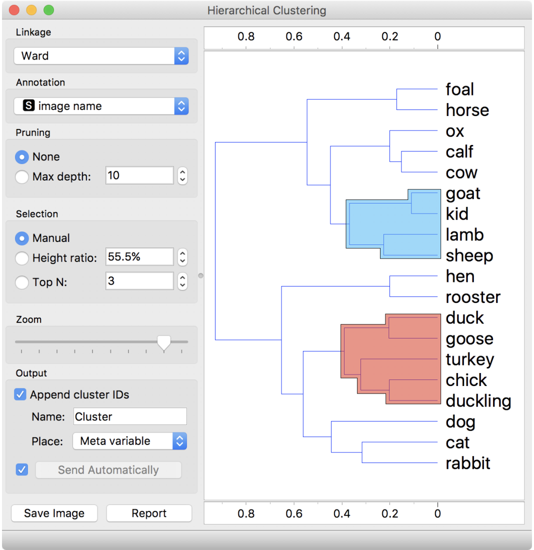
Feladat

- Úgy csoportosítunk dolgokat, hogy a hasonlóak egy csoportba kerüljenek
  - Klaszteren belül minél hasonlóbbak
  - Klaszterek között minél kevésbé hasonlóak
- A dolgok általában $\mathbb{R}^n$-beli (vagy gráfbeli) pontok, pl.:
  - Ügyféladatok piacszegmentáláshoz
  - Dokumentumok szózsákkal modellezve, témák meghatározásához,keresési találatok összegzésére
  - Szavak kontextusai, jelentések indukálásához
  - Szerverek adatai (melyikek aktívak általában együtt)
- Egy csoportot egy *klaszternek* hívunk

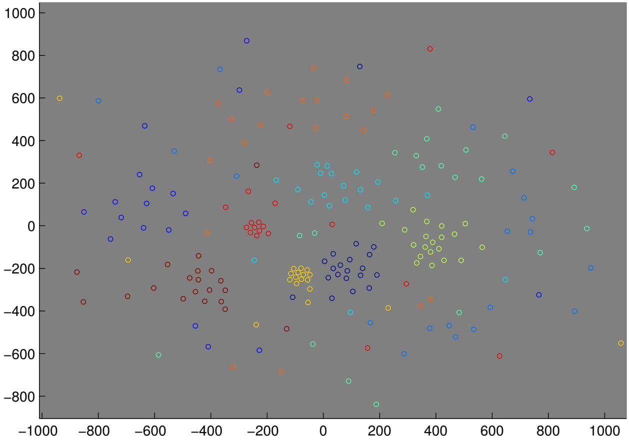
Fajtái

- Lehet hard vagy soft clustering
  - Hard clustering: egy adatpont csak egy klaszterben szerepelhet
  - Soft clustering: minden adatpontra megvan, hogy mennyire tartozik az egyes klaszterekbe
- Átmenetek
  - Átfedő klaszterezés: egy elem több klaszterbe is tartozhat, de vagy beletartozik, vagy nem
  - Hierarchikus klaszterezés: a klasztereket hierarchiába szervezzük, a gyerek klaszterbe tartozó elemek a szülőbe is beletartoznak

Hierarchikus klaszterezés

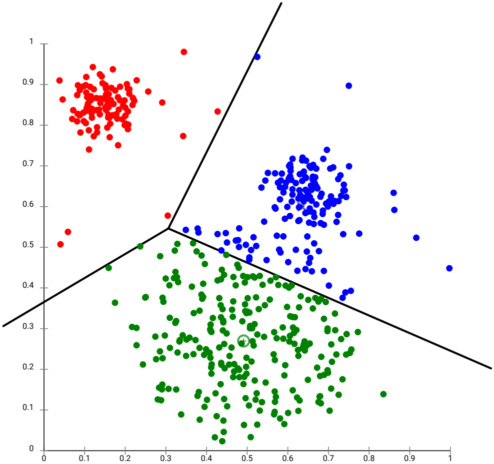


Példa természetes klasztereződésre – azonos értelmű szavak jelentései (t-SNE)



Tartalom

Példa



Feladat

- Adott: k, a klaszterek száma
- Minden klasztert a középpontjával reprezentálunk
  - Centroid, a klaszter pontjainak átlaga
- A feladat: keressük meg a k klaszter középpontot és az adatpontokat rendeljük ezekhez hozzá úgy, hogy a klaszteren belüli, középponttól számított távolságnégyzeteket minimalizáljuk
  - Ekvivalens a páronkénti távolságnégyzetek minimalizálásával
  - NP-nehéz, így approximáljuk
  - Csak lokális optimumot találunk
  - Többször futtathatjuk különböző véletlen inicializációkkal

k-means feladat

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}\in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{\mathbf{x},\mathbf{y}\in S_i} \|\mathbf{x} - \mathbf{y}\|^2$$

Algoritmus

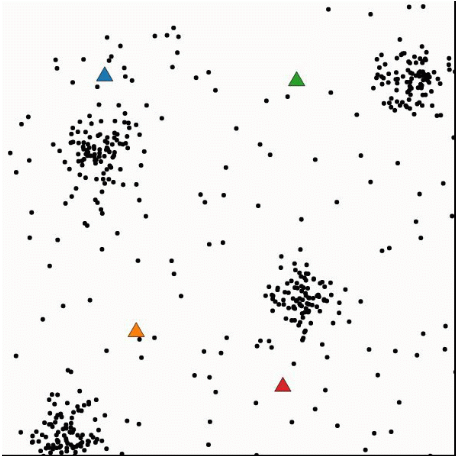
- Kezdetben adott k és az adatpontok $\mathbb{R}^n$-ben
- Inicializáljuk az $m_1^{(1)}, m_2^{(1)}, \ldots, m_k^{(1)}$ centroidokat
  - Véletlenszerűen kiválasztunk k adatpontot, vagy
  - Minden adatpontot véletlenszerűen egy klaszterbe sorolunk és kiszámoljuk a centroidokat
- Váltogatjuk a következő két lépést, amíg nem konvergálunk
  1. Minden adatpontot a legközelebbi centroidhoz rendelünk:

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \ \forall j, 1 \leq j \leq k \right\}$$

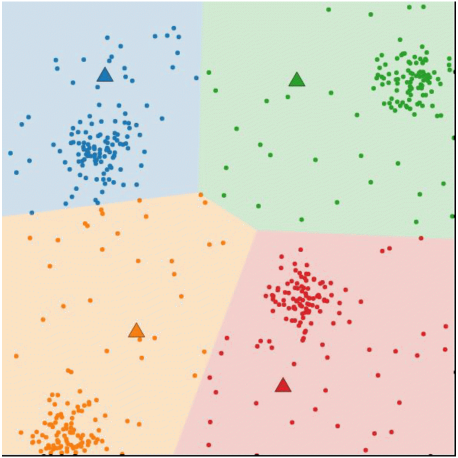
  2. Kiszámítjuk az új centroidokat

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j$$

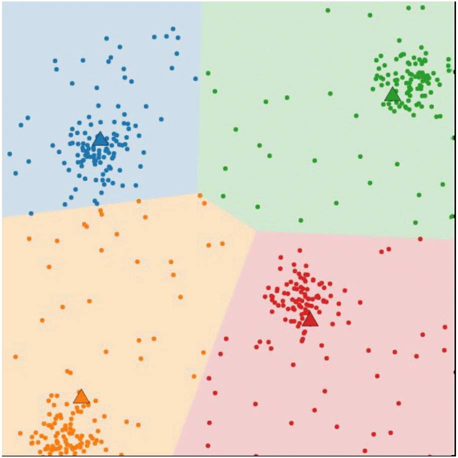
Algoritmus



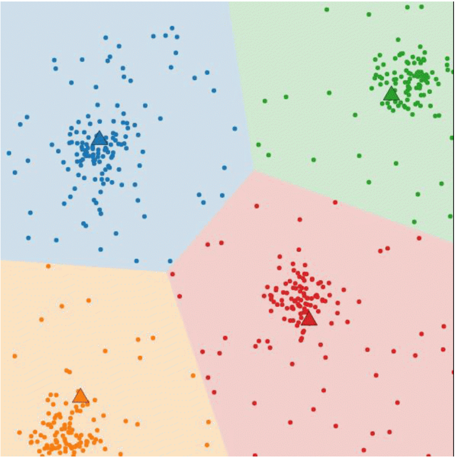
Algoritmus



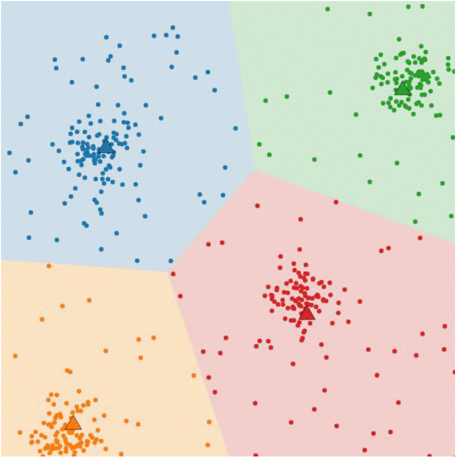
Algoritmus



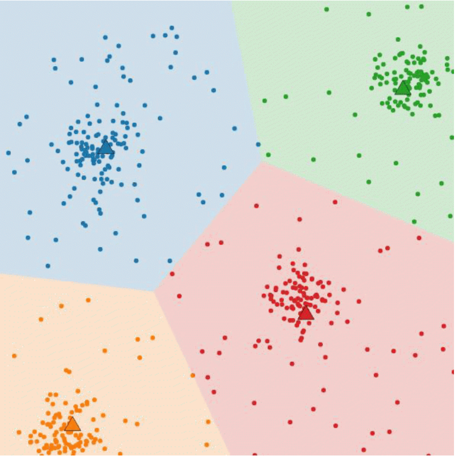
Algoritmus



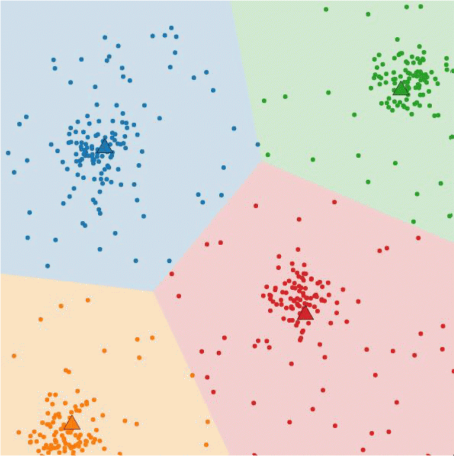
Algoritmus



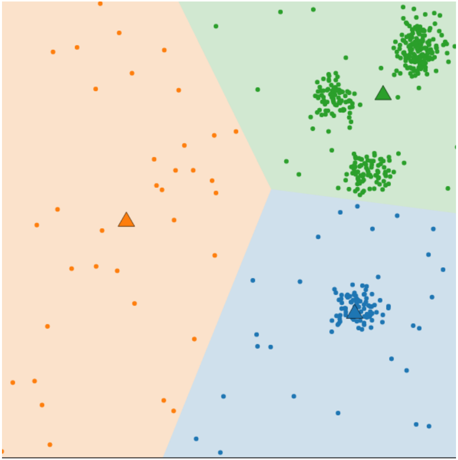
Algoritmus



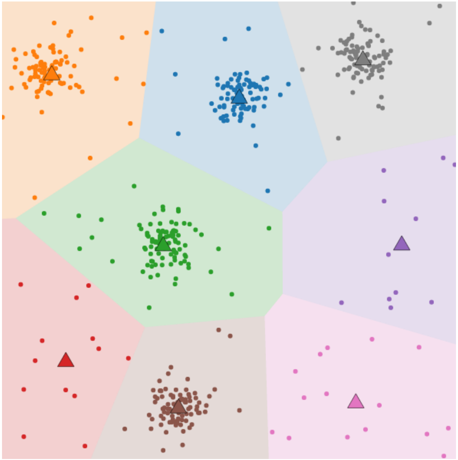
Algoritmus



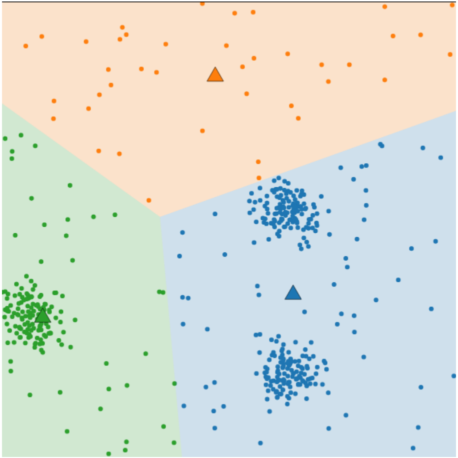
Problémák – túl kicsi k-t adunk meg



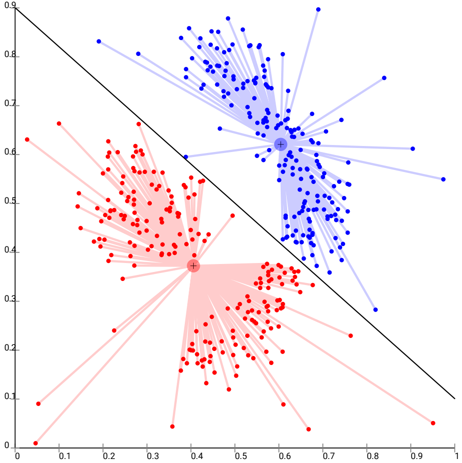
Problémák – túl nagy k-t adunk meg



Problémák – rossz inicializáció



Problémák – sűrűség alapúak a klaszterek



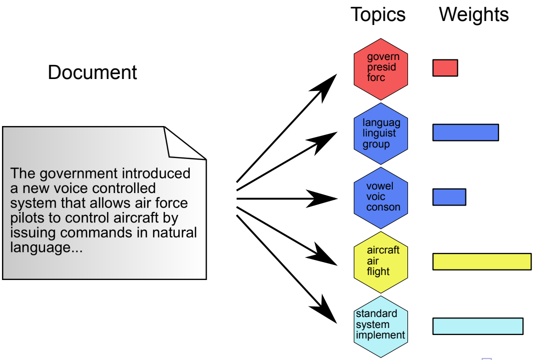
Python példák

- Kép kvantálás: http://scikit-learn.org/stable/auto_examples/cluster/plot_color_quantization.html
- Dokumentumok klaszterezése: http://scikit-learn.org/stable/auto_examples/text/document_clustering.html

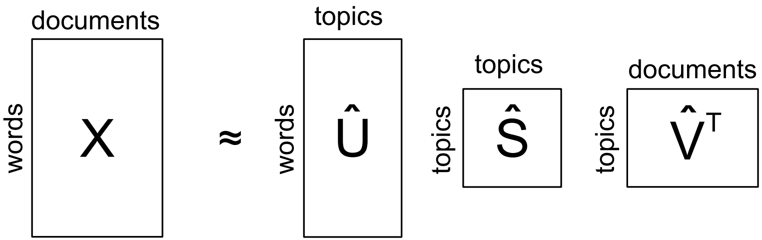
Tartalom

Témamodellek

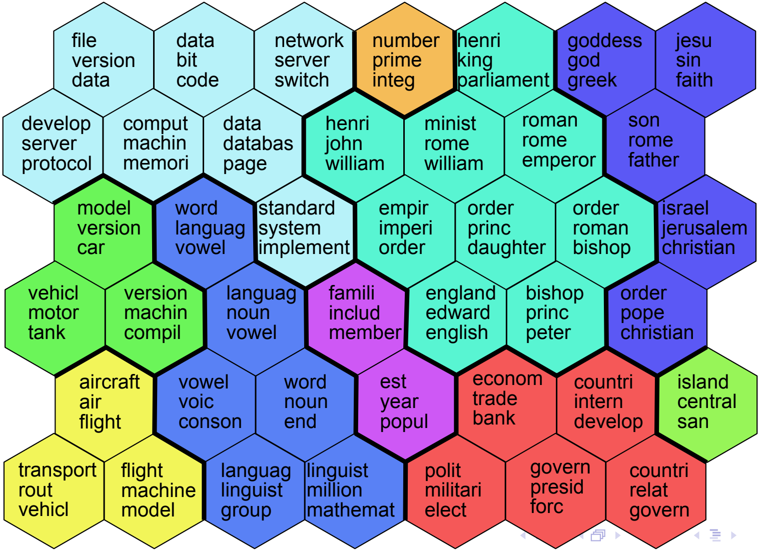
- Soft clusteringre példa: témamodellek
  - Egy dokumentum mennyire szól az egyes témákról
  - Egy témába milyen szavak tartoznak?
  - Pl. Latent Semantic Analysis (SVD)



Latent Semantic Analysis



Kiterjesztés csoportritka regularizációval



Példa: kakukktojás játék

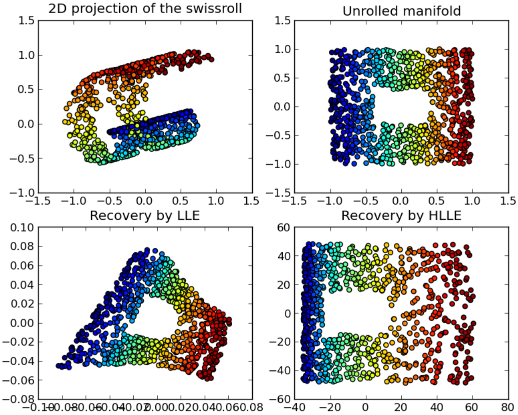
Egybetartozó szavak				Kakukktojás
cao	wei	liu	emperor	**king**
superman	clark	luthor	kryptonite	**batman**
devil	demon	hell	soul	**body**
egypt	egyptian	alexandria	pharaoh	**bishop**
singh	guru	sikh	saini	**delhi**
language	dialect	linguistic	spoken	**sound**
mass	force	motion	velocity	**orbit**
voice	speech	hearing	sound	**view**
athens	athenian	pericles	corinth	**ancient**
data	file	format	compression	**image**
function	problems	polynomial	equation	**physical**

Tartalom

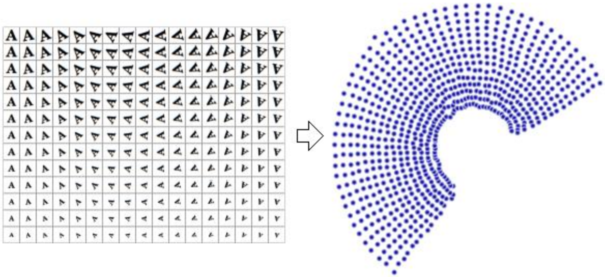
Miért dimenziócsökkentünk?

- Az adatok valójában alacsonyabb dimenziósak, csak magasabb dimenziós térben vannak
- Láttatjuk az adatokat
- Eltüntetjük a zajt
- Csökkentjük a tanulási feladat bonyolultságát (jobb eredmények, kisebb futási idő, ...)
- A csökkentett dimenziójú adatokon új törvenyszerűségeket, sejtéseket láthatunk meg
- A probléma megoldásához kisebb dimenziós és/vagy sűrű reprezentációra van szükségünk

Példa: Swiss roll

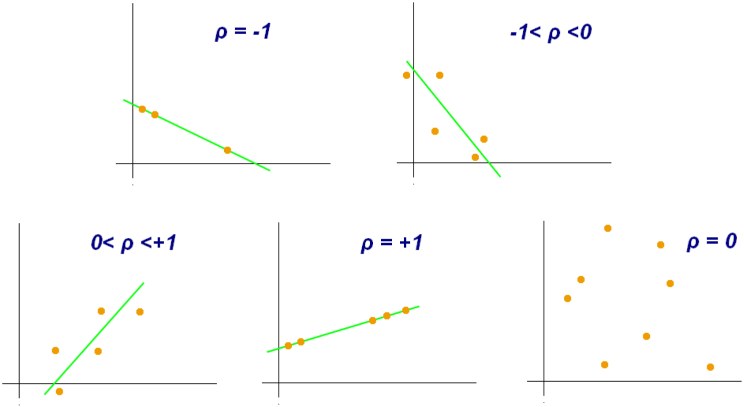


Példa: A betű forgatása

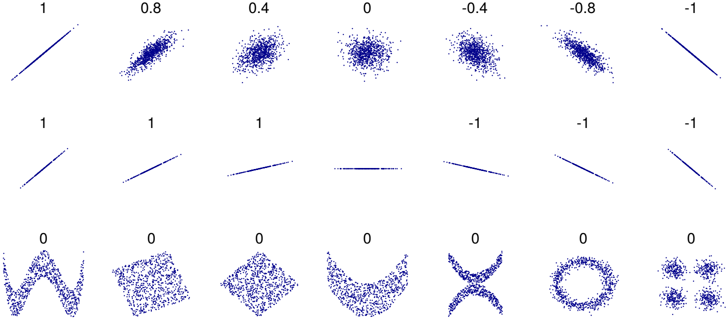


Tartalom

Kovariancia, korreláció



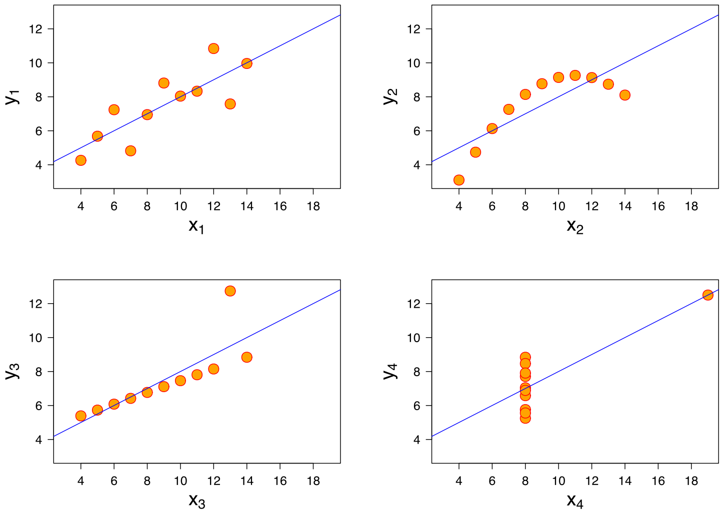
Kovariancia, korreláció



Kovariancia, (Pearson) korreláció

- Azt mérik, hogy X, Y val. változók mennyire mozognak együtt
- Lineáris kapcsolatot mutatnak
- $\mathrm{Cov}(X, Y) = \mathrm{E}\left[(X - \mathrm{E}[X])(Y - \mathrm{E}[Y])\right]$
- Pl.: Pozitív: Ha $X > E(X)$, akkor $Y > E(Y)$, ha $X < E(X)$, akkor $Y < E(Y)$
- $\mathrm{Cov}(X, Y) = \mathrm{E}\left[XY\right] - \mathrm{E}[X]\,\mathrm{E}[Y]$
- Korreláció: „Normalizált” kovariancia, -1 és 1 között
- $\rho_{X,Y} = \mathrm{corr}(X, Y) = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$
- $r_{xy} = \frac{\sum\limits_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum\limits_{i=1}^{n}(x_i-\bar{x})^2 \sum\limits_{i=1}^{n}(y_i-\bar{y})^2}}$

Mind a négy adathalmaz korrelációs együtthatója 0.816

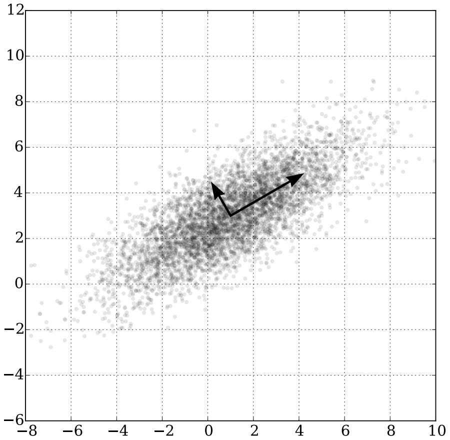


Kovariancia mátrix

- $\mathbf{X}$ egy vektor, aminek az elemei val. változók
- A kovariancia mátrix elemei X_i, X_j közti kovarianciák
- $\Sigma_{ij} = \mathrm{cov}(X_i, X_j) = \mathrm{E}\left[(X_i - \mu_i)(X_j - \mu_j)\right]$
- $\mu_i = \mathrm{E}(X_i)$
- $$\begin{bmatrix} \mathrm{E}[(X_1-\mu_1)(X_1-\mu_1)] & \mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)] \\ \mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)] & \mathrm{E}[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)] & \mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}$$
- A főátlóban a szórások vannak.
- Ekvivalens: $\Sigma = \mathrm{E}(\mathbf{X}^\top\mathbf{X}) - \mu^\top\mu$

Tartalom

Példa - 2d normáleloszlás



Főkomponens analízis

- Principal component analysis (PCA)
- Demo: `http://setosa.io/ev/principal-component-analysis/`
- Az adathalmazt egy új koordinátarendszerben ábrázoljuk, a tengelyek merőlegesek
- Az adathalmaz vetítései közül a legnagyobb szórású az első tengelyen (főkomponensen) van
- A második legnagyobb szórású a második főkomponensen, ...
- Új változók/adatok: a főkomponensekre vetítjük le az eredeti változókat. Ezek már korrelálatlanok
- Dimenziócsökkentés: eldobjuk azokat a tengelyeket (és koordinátákat), amiken kicsi a szórás

Főkomponens analízis

- $\mathbf{X} \in \mathbb{R}^{n \times p}$: adathalmaz, egy sor egy adatpont
- $\mathbf{t}_{(i)} = (t_1, \ldots, t_l)_{(i)}$: az adatpontok az új koordinátarendszerbe transzformálva $\mathbf{w}_{(k)} = (w_1, \ldots, w_p)_{(k)}$-val

$$t_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(k)} \qquad \text{for} \qquad i = 1, \ldots, n \qquad k = 1, \ldots, l$$

- Szórás maximalizálása

$$\mathbf{w}_{(1)} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \sum_i (t_1)^2_{(i)} \right\} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \sum_i \left(\mathbf{x}_{(i)} \cdot \mathbf{w} \right)^2 \right\}$$

Főkomponens analízis

- Ugyanez mátrixosan:

$$\mathbf{w}_{(1)} = \underset{\|\mathbf{w}\|=1}{\arg\max} \{\|\mathbf{X}\mathbf{w}\|^2\} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}\right\}$$

- Mivel **w** egységvektor:

$$\mathbf{w}_{(1)} = \arg\max \left\{\frac{\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}}{\mathbf{w}^T\mathbf{w}}\right\}$$

- Ez a Rayleigh-hányados, a legnagyobb lehetséges érték az $\mathbf{X}^T\mathbf{X}$ legnagyobb sajátértéke lesz, ahol **w** a hozzá tartozó sajátvektor
- A többi komponensre is így van $\rightarrow$ a főkomponensek az $\mathbf{X}^T\mathbf{X}$ sajátvektorai

Főkomponens analízis – algoritmus

- Az $\mathbf{X}$ mátrixban vannak az adataink
- Nulla átlagúra hozzuk az adatokat (kivonjuk az átlagot)
- Kiszámoljuk a $\mathbf{Q} = \mathbf{X}^T\mathbf{X}$ kovariancia mátrixot
- Meghatározzuk ennek a mátrixnak a sajátértékeit, és a sajátvektorait
- A sajátvektorok a főkomponensek, a belőlük álló bázis az új koordinátarendszer
- A legnagyobb sajátértékhez tartozó főkomponens a legnagyobb szórású, és így tovább
- Dimenziócsökkentés: csak a k legnagyobb sajátértékű főkomponenst tartjuk meg

PCA és SVD

SVD

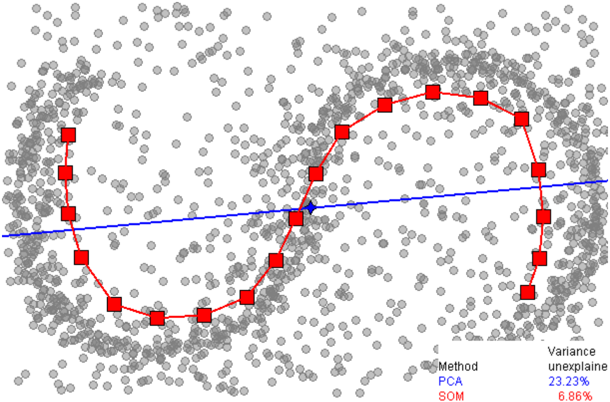
$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T$$

PCA SVD-vel

$$\begin{aligned}\mathbf{X}^T\mathbf{X} &= \mathbf{W}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{W}^T \\ &= \mathbf{W}\mathbf{\Sigma}^T\mathbf{\Sigma}\mathbf{W}^T \\ &= \mathbf{W}\hat{\mathbf{\Sigma}}^2\mathbf{W}^T\end{aligned}$$

- $\mathbf{W}$-ben már $\mathbf{X}^T\mathbf{X}$ sajátvektorai vannak. A szinguláris értékek a sajátértékek négyzetgyökei.

A PCA is lineáris

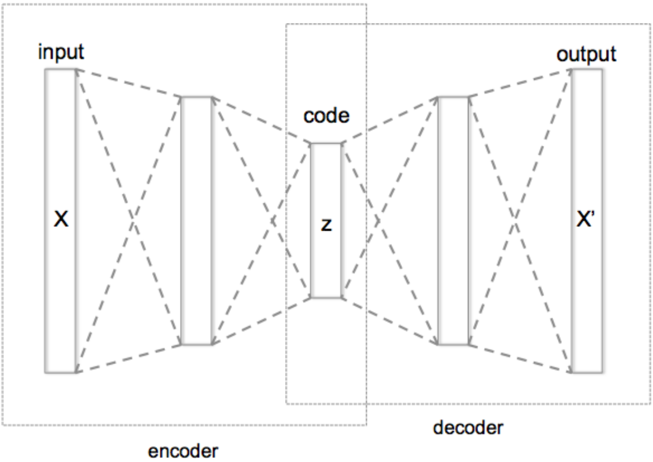


Python példák

- A feature scaling fontossága: http://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html

Tartalom

Autoenkóderek



Autóenkóderek

Egyszerű autóenkóder

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2 = \|\mathbf{x} - \sigma'(\mathbf{W}'(\sigma(\mathbf{W}\mathbf{x} + \mathbf{b})) + \mathbf{b}')\|^2$$

- Ez az egyszerű autoenkóder a PCA alterébe projektál
- Flexibilis, sokféle variáció létezik
  - Denoising autoencoder: zajos inputból kell zajtalan outputot előállítani
  - Sparse autoencoder: csak néhány egység lehet aktív a rejtett reprezentációban
  - VAE: Egy valószínűségi modellt feltételez, a poszterior eloszlást approximálja
- Sokszor fontosak egy felügyelt mély háló előtanításában

Köszönöm a figyelmet!

Köszönöm a figyelmet!