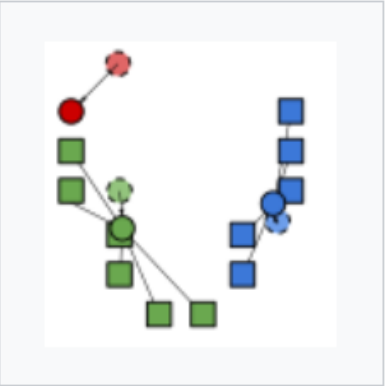
K-means og PageRank

FYS-2021, 18.10.24

Bakgrunn for algoritme

k "gjennomsnitt" ved start (i dette tilfellet *k* = 3) blir tilfeldig generert innenfor data-domenet (vist i fargene rød, grønn og blå).

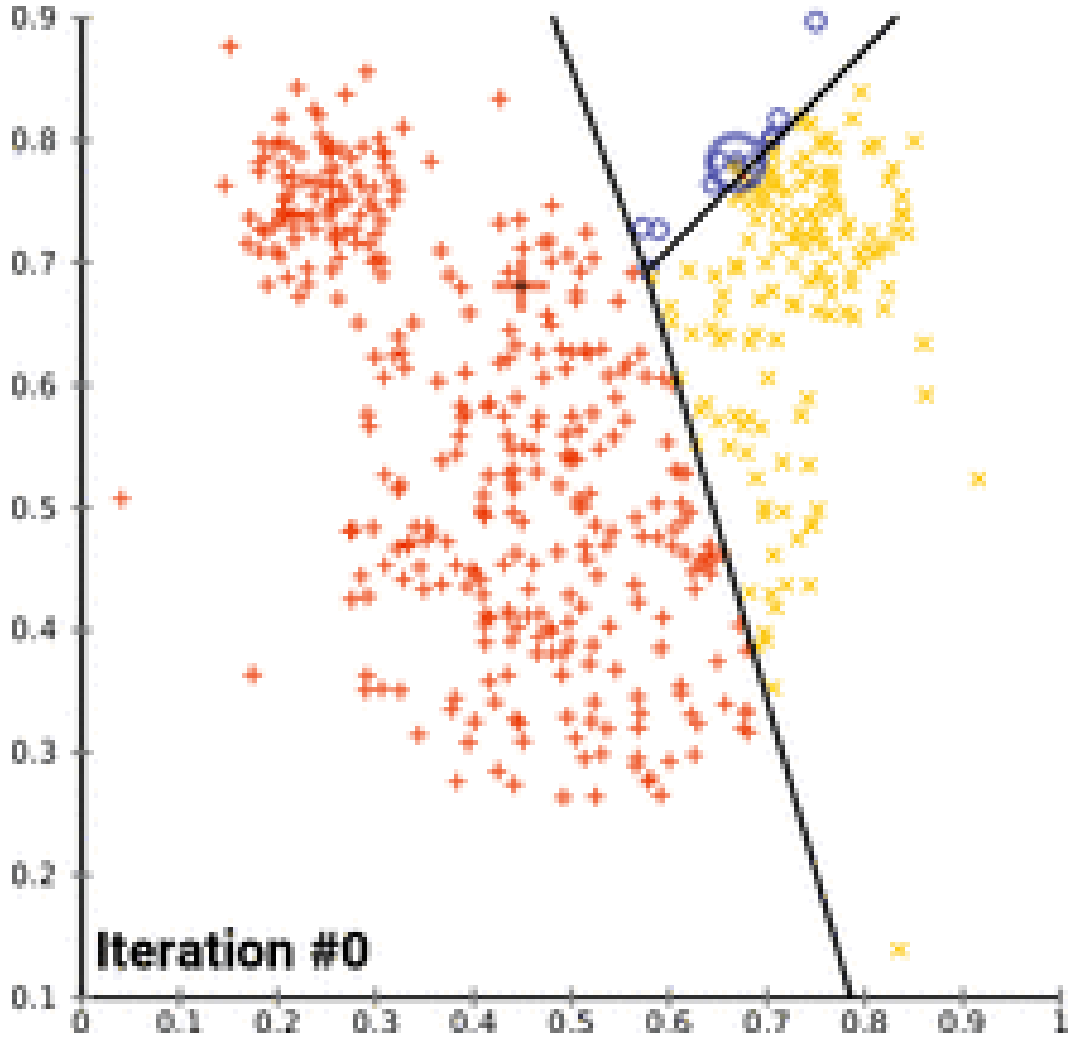
k grupper lages ved å assosiere hver observasjon med det nærmeste gjennomsnittet. Partisjonene her representerer Voronoi-diagrammet generert av gjennomsnittene.



Sentroiden av hvert av de *k* gruppene blir det nye gjennomsnittet.

Steg 2 og 3 repeteres til konvergens har blitt nådd (sentroidene flytter seg ikke).

Animasjon



NB: Legg merke til sentroide-punktet som er plassert i tyngdepunketet til hver cluster

Anta k clusters m. tilhørende sentroider $\{\underline{m}_k\}_{k=1}^{K}$

Vi har N samples $\{x_i\}_{i=1}^{N}$

For hver kluster k, definer en binær maske

$$b_k^i = \begin{cases} 1, & \underline{x}^i \text{ tilhører kluster } k \\ 0, & \text{dersom ikke.} \end{cases}$$

kluster →

$$b = \begin{matrix} \downarrow \\ x \end{matrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \leftarrow$$

Algoritme

1) Initialiser $\{m_k\}$ to k random $\underline{x}^i$ punkter

Iterer til "lik" endring eller max iterasjoner:

2a) For alle $\underline{x}^i$ i datasett:

a) Initialiser $\underline{b}^i = 0$

b) Finn k slik at $\min_k \|x^i - m_k\|$ c) Sett $b_k^i = 1$

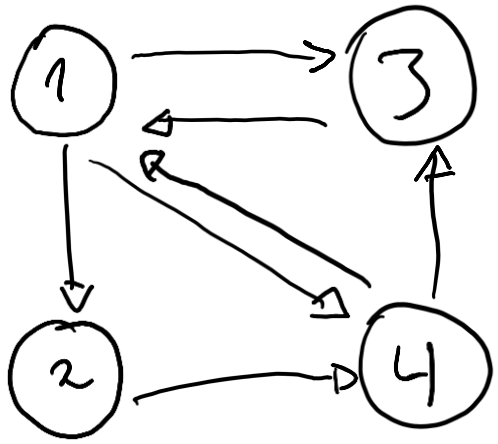
2b) For alle $\underline{m}_k$, $k = 1, \ldots, K$ opdater kluster centre (centroider)

$k = 1, 2, \ldots$ $\quad \underline{m}_k = \sum_{i=0}^{N} b_k^i x^i \Big/ \sum_{i=1}^{N} b_k$ ⟵ Mean value coordinates

Pagerank algorithm

- Agents follow hyperlinks randomly $Y_{t+1} = GY_t$
  - Add +1 when a page is visited
- To avoid being stuck with pages without links -> Add a probability to randomly jump to another page
  - $G = (1 - \alpha)H + \alpha B$
  - $B = \frac{1}{n}\begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$
  - H is the transition matrix
  - α is the “weight” of random walk. Small value means little probability of visiting a random page

Problem 2: Adjacency matrix



$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Problem 2: transition matrix

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Degree matrix

$$D = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

$$B = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Transition matrix

$$H = \begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 \end{pmatrix}$$

Google Matrix :

$$G = (1-\alpha)H + \alpha B$$

↑ Random jump

Power method

- Take an initial vector Y_0, also called π (can be random generated)
- Apply the Google matrix G n times (follows from $Y_{m+1} = GY_m$)
  - After each iteration, normalize Y ($Y_{n+1} = \frac{GY_m}{\|GY_m\|}$)
  - Then this will converge: $Y_m \to S$, where $S = GS$ (steady state)
  - S is then the eigenvector with eigenvalue 1
- The entries in the S vector will rank the various pages!