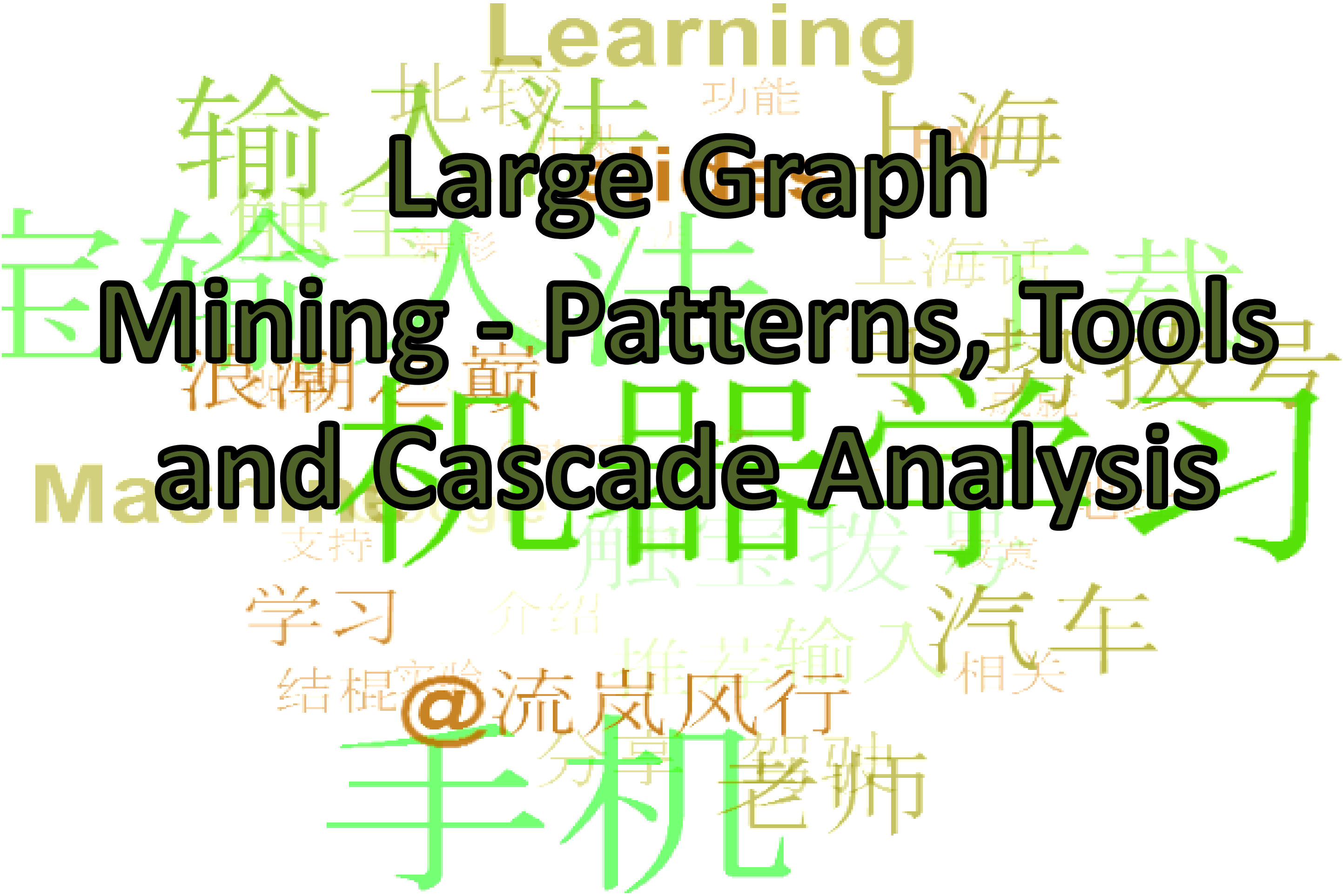
Large Graph Mining - Patterns, Tools and Cascade Analysis



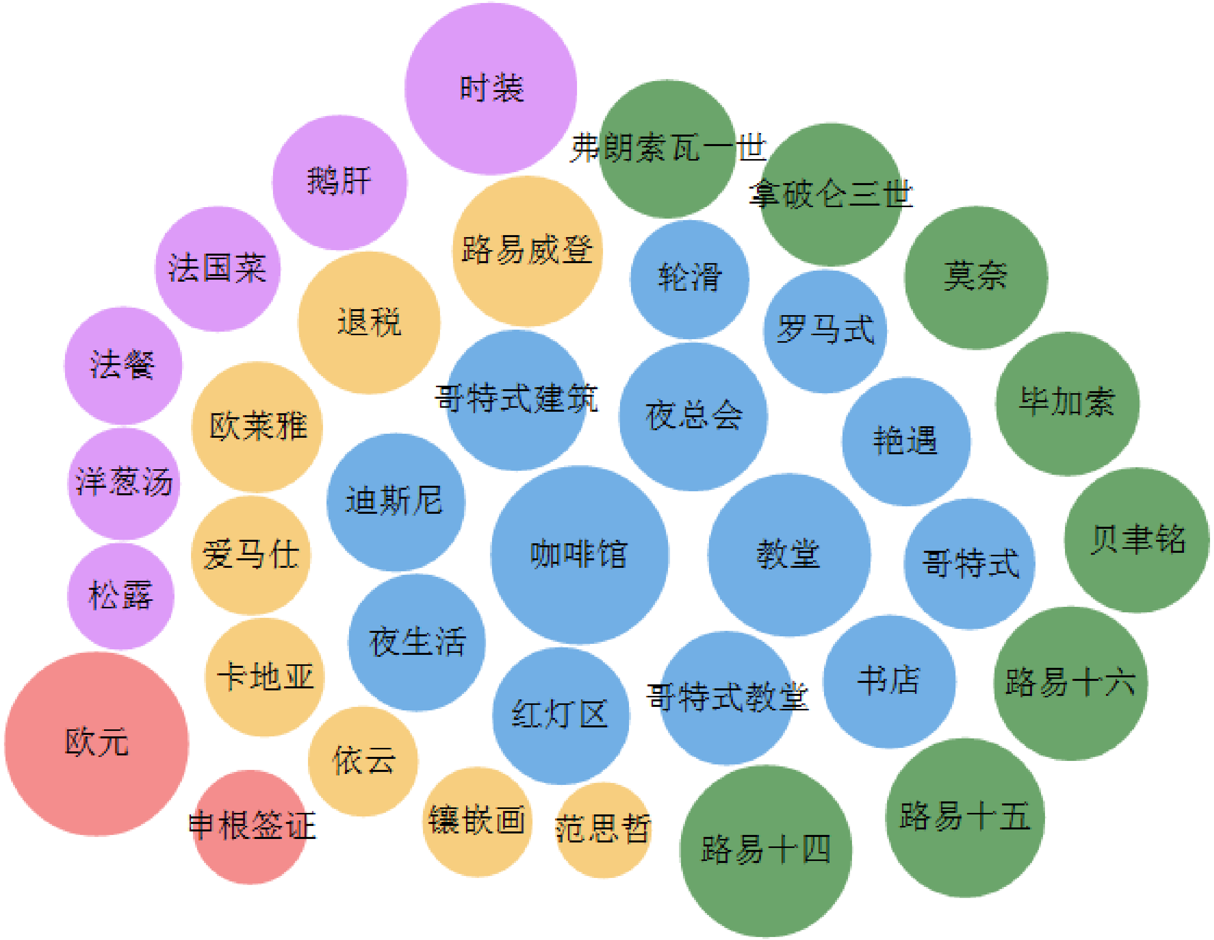
outline

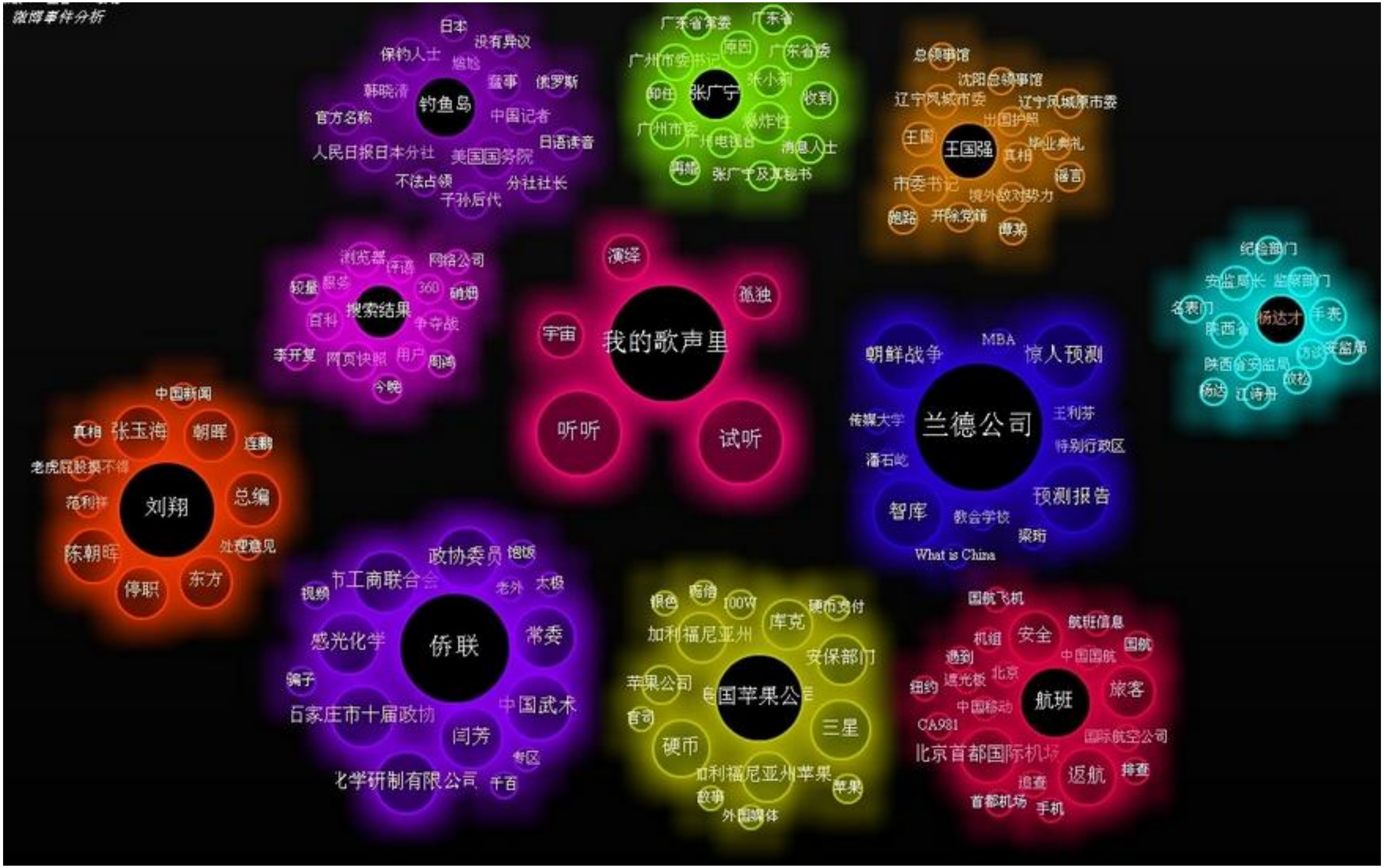
- simple tag cloud
- K-Means
- some python tools

simple tag cloud

$$freq_{norm} = \frac{word_{freq} - min}{\max - min}$$

$$freq_{norm} = \frac{(word_{freq} - min)(upper_{bound} - lower_{bound})}{\max - min} + lower_{bound}$$


时装
弗朗索瓦一世
鹅肝
拿破仑三世
路易威登
法国菜
轮滑
莫奈
退税
罗马式
法餐
哥特式建筑
夜总会
毕加索
欧莱雅
艳遇
洋葱汤
迪斯尼
贝聿铭
咖啡馆
教堂
爱马仕
哥特式
松露
夜生活
路易十六
卡地亚
书店
哥特式教堂
红灯区
欧元
依云
申根签证
镶嵌画
范思哲
路易十五
路易十四


微博事件分析
钓鱼岛
日本
没有异议
保钓人士
尴尬
蠢事
俄罗斯
韩晓清
官方名称
中国记者
日语读音
人民日报日本分社
美国国务院
不法占领
分社社长
子孙后代
张广宁
广东省常委
广东省
原因
广州市委书记
广东省委
张小莉
卸任
收到
爆炸性
广州市委
广州电视台
消息人士
调离
张广宁及其秘书
王国强
总领事馆
沈阳总领事馆
辽宁凤城市委
辽宁凤城原市委
出国护照
王国
毕业典礼
真相
谣言
市委书记
境外敌对势力
跑路
开除党籍
谭某
搜索结果
浏览器
评语
网络公司
较量
服务
360
硝烟
百科
争夺战
李开复
网页快照
用户
周鸿
今晚
我的歌声里
演绎
孤独
宇宙
听听
试听
兰德公司
朝鲜战争
MBA
惊人预测
传媒大学
王利芬
特别行政区
潘石屹
智库
教会学校
预测报告
梁珩
What is China
杨达才
纪检部门
安监局长
监察部门
名表门
陕西省
手表
访谈
安监局
陕西省安监局
放松
杨达
江诗丹
刘翔
中国新闻
真相
张玉海
朝晖
连翻
老虎屁股摸不得
范利祥
总编
陈朝晖
处理意见
停职
东方
侨联
政协委员
饱饭
市工商联合会
视频
老外
太极
感光化学
常委
骗子
石家庄市十届政协
中国武术
闫芳
特区
化学研制有限公司
千百
美国苹果公司
银色
赔偿
100W
库克
硬币支付
加利福尼亚州
安保部门
苹果公司
官司
三星
硬币
加利福尼亚州苹果
苹果
致辞
外国媒体
航班
国航飞机
航班信息
安全
机组
国航
中国国航
遇到
北京
遮光板
纽约
旅客
中国移动
CA981
国际航空公司
北京首都国际机场
返航
排查
追查
首都机场
手机

K-Means Algorithm

K = # of clusters (given);

One “mean” per cluster

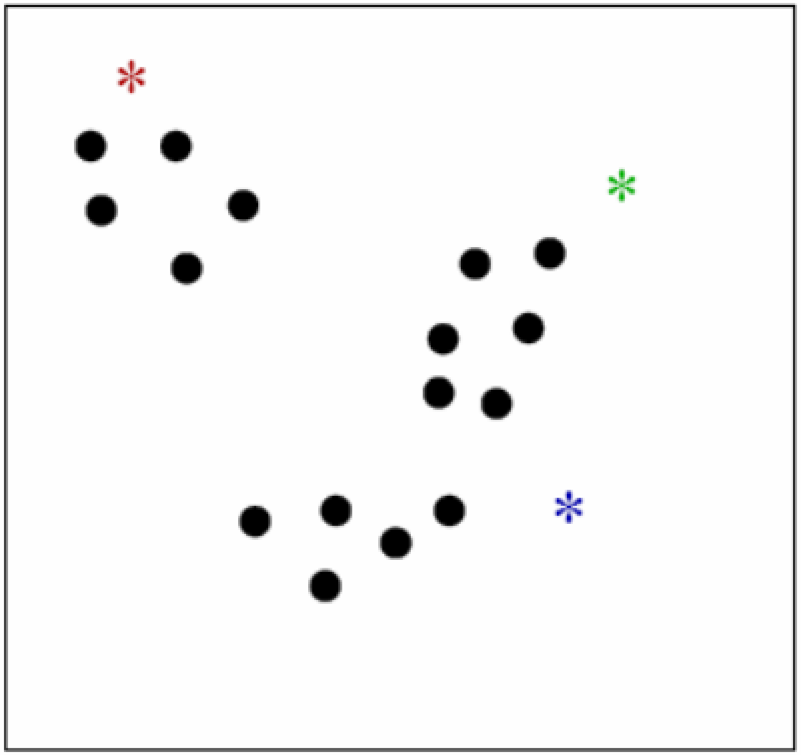
- Initialize means

(e.g. by picking k samples at random)

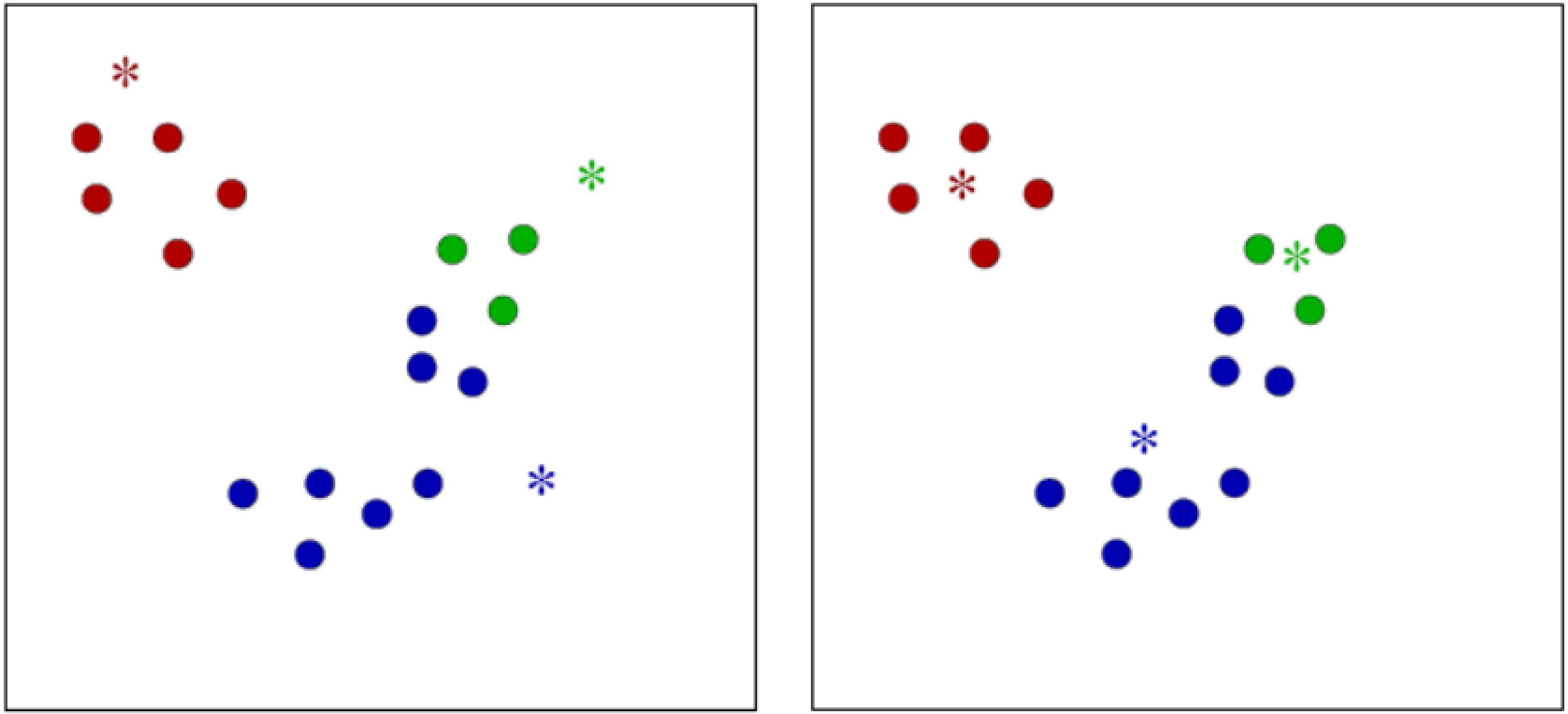
- Iterate:

(1) assign each point to nearest mean

(2) move “mean” to center of its cluster



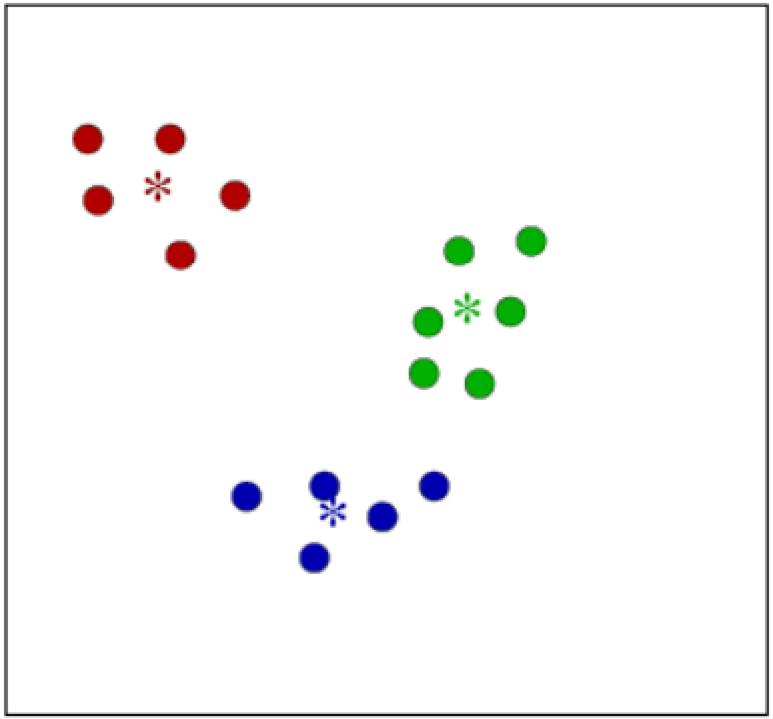
Means Update



K-Means Algorithm

- Complexity:

O(kn # of iterations)



- The object function is:

$$J = \min_{\{\mu_1,\cdots,\mu_k\}} \sum_{h=1} \sum_{x \in X_h} \|x - \mu_h\|^2$$

K-Means Algorithm

- **Initialize** $\mu_1, \cdots, \mu_k$
- **do** classify n samples according to nearist μ_h

  recompute μ_h
- **until** no change in μ_h

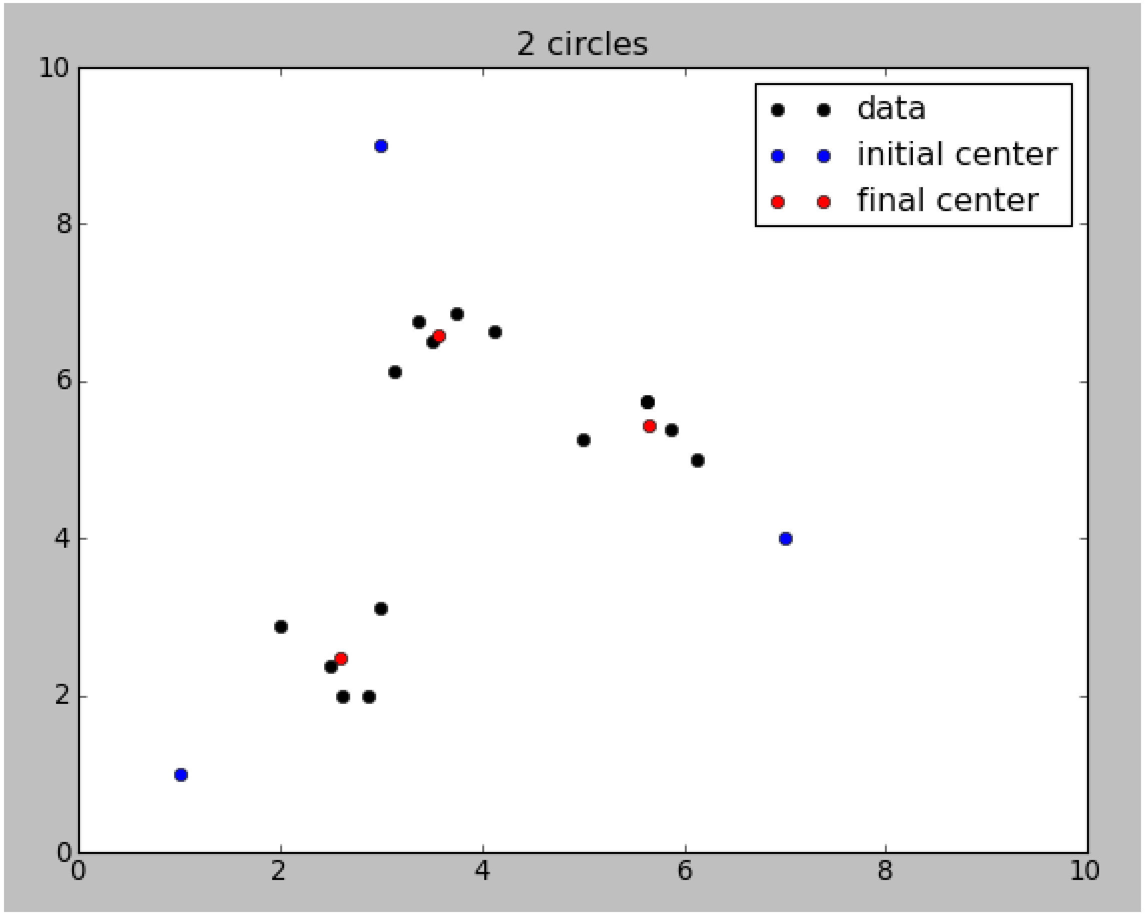
K-Means Algorithm

$$\frac{\partial}{\partial \mu_h} J = \frac{\partial}{\partial \mu_h} \sum_{i=1}^{k} \sum_{x \in X_h} \|x - \mu_h\|^2 = \sum_{i=1}^{k} \sum_{x \in X_h} \frac{\partial}{\partial \mu_h} \|x - \mu_h\|^2 = 0$$

$$\sum_{x \in X_h} 2(x - \mu_h) = 0 \rightarrow \mu_h = \frac{1}{n_h} \sum_{x \in X_h} x_h$$

K-Means in Python

Simple k-means code using numpy and matplotlib



Find related words

- 9.10 weibo corpus sample

Example:

- # 钓鱼岛 是 中国 的 #~ 我是 热血 爱国 好青年
- 这样的 第一个 教师节 也算是 难忘 了 。

100 cases

50：钓鱼岛是中国的

50：教师节

Find related words

- Use K-Means to cluster into two classed

Feature size : 1338

Feature ∈{0.0, 1.0}

Two classes

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum\limits_{i=1}^{n} A_i \times B_i}{\sqrt{\sum\limits_{i=1}^{n} (A_i)^2} \times \sqrt{\sum\limits_{i=1}^{n} (B_i)^2}}$$

Find related words

Pycluster used

```
labels, error, nfound = Pycluster.kcluster(weibo, 2)
```

Average accuracy: 0.933

Bad examples:

教师节 向 德艺双馨 的 人民 艺术家 ** 致以 最真诚的 祝福**
钓鱼岛 是 中国 的， ** 是 世界 的

Draw the result

0 钓鱼岛

1 中国

2 日本

3 领土

4 政府

5 我们

6 就是

7 有本事

8 日本人

9 小日本

10 明天

11 国有化

12 谴责

13 问题

14 退让

15 起来

16 垃圾

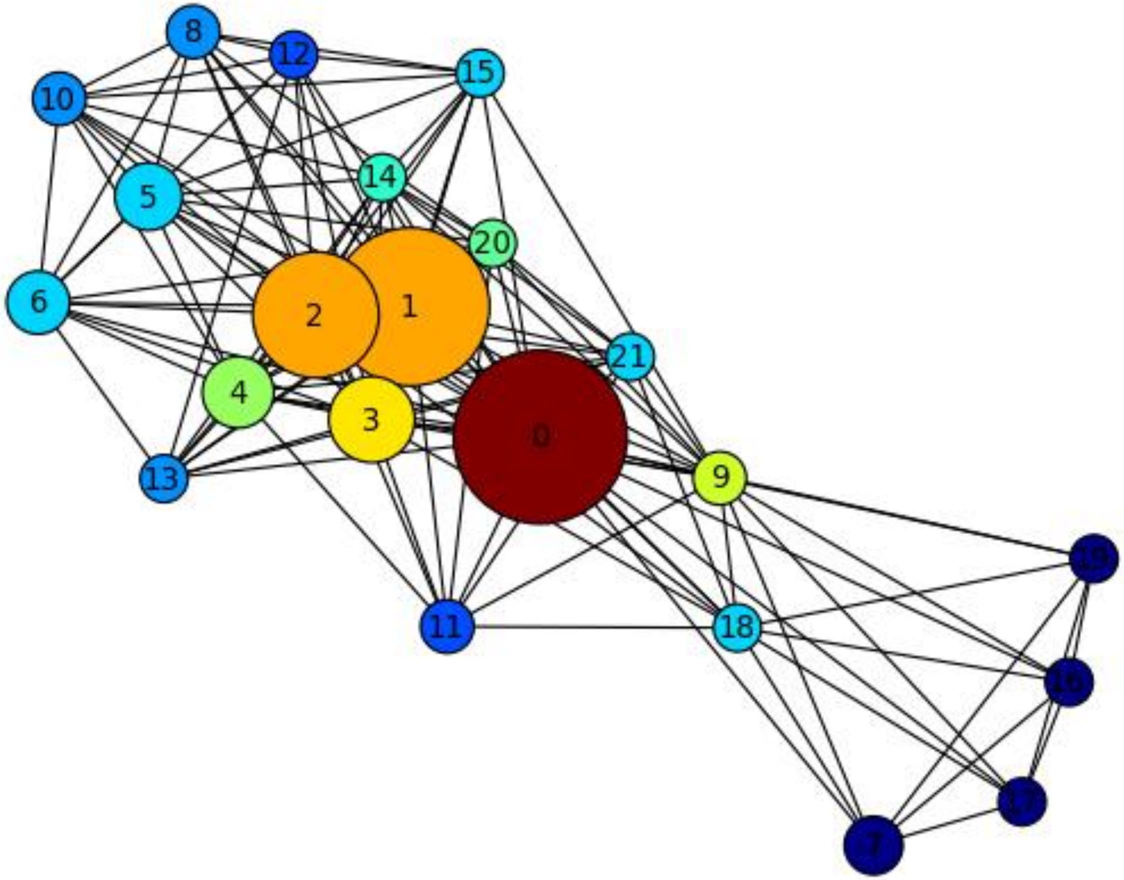
17 滚蛋

18 历史

19 破烂

20 固有

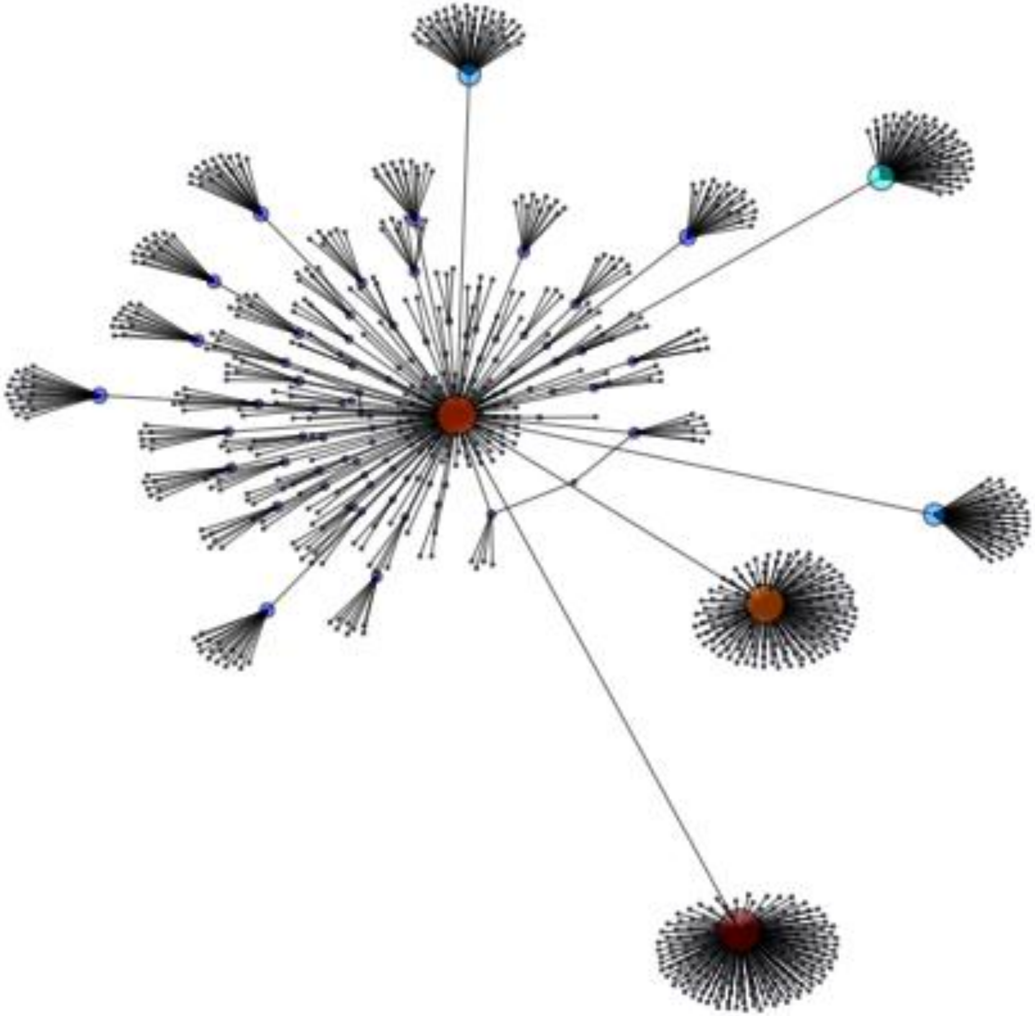
21 主权



networkx

- import networkx as nx
- G = nx.Graph()
- G.add_node(*)
- G.add_edge(*)
- nx.draw(G)

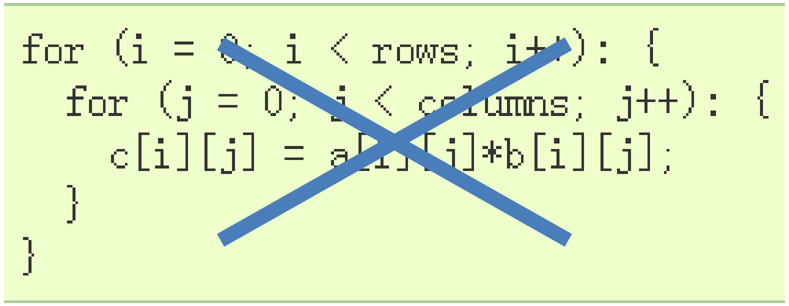
use networkx



Numpy and matplotlib

- NumPy is the fundamental package for scientific computing in Python.

- C = A*B



```
for (i = 0; i < rows; i++): {
  for (j = 0; j < columns; j++): {
    c[i][j] = a[i][j]*b[i][j];
  }
}
```

Install them!

1、安装ipython

```
sudo apt-get install ipython
```

2、安装matplotlib

```
sudo apt-get install python-matplotlib
```

3、启动绘图环境

```
ipython -pylab
```

4、安装pycluster

```
sudo pip install pycluster
```

5、安装networkx

```
sudo easy_install networkx
```