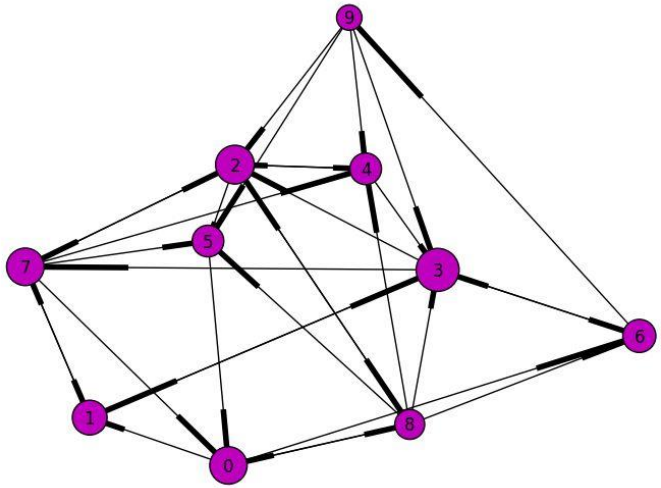
Python程序设计

陈远祥

chenyxmail@gmail.com

北京邮电大学 电子工程学院

机器学习



机器学习

机器学习的定义

- 机器学习(Machine Learning, ML)是一门多领域交叉学科，涉及概率论、统计学、逼近论、凸分析、算法复杂度理论等多门学科。专门研究计算机怎样模拟或实现人类的学习行为，以获取新的知识或技能，重新组织已有的知识结构（利用数据或经验等）使之不断改善自身的性能
- 它是人工智能的核心，是使计算机具有智能的根本途径，其应用遍及人工智能的各个领域，包括网络搜索、垃圾邮件过滤、推荐系统、广告投放、信用评价、欺诈检测、股票交易和医疗诊断等应用

机器学习

- 机器学习分类
  - ✓ 监督学习（supervised learning）
  - ✓ 无监督学习（unsupervised learning ）
  - ✓ 强化学习（reinforcement learning，增强学习）
  - ✓ 半监督学习（semi-supervised learning ）

机器学习

机器学习分类

- ✓ 监督学习：主要特点是要在训练模型时提供给学习系统训练样本以及样本对应的类别标签，因此又称为有导师学习。例：学生从老师那里获取知识、信息，老师提供对错指示、告知最终答案的学习过程
- ✓ 典型的监督学习方法：决策树、支持向量机（SVM）、监督式神经网络等分类算法和线性回归等回归算法

机器学习

- 机器学习分类
  - ✓ 无监督学习，主要特点是训练时只提供给学习系统训练样本，而没有样本对应的类别标签信息。例：没有老师的情况下，学生从书本或网络自学的过程
  - ✓ 典型的无监督学习方法：聚类学习、自组织神经网络学习

机器学习

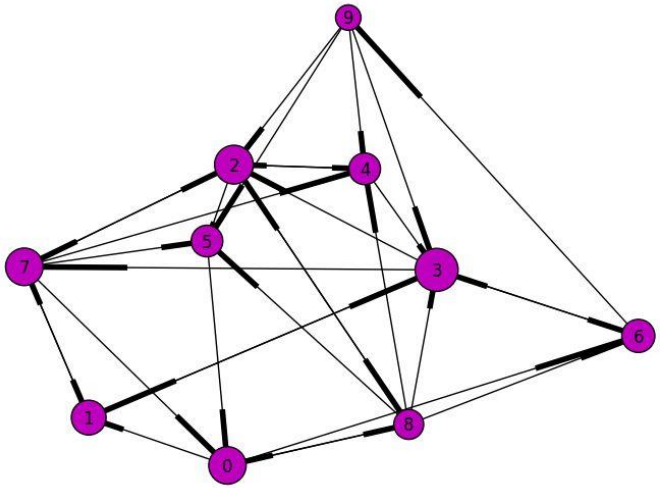
机器学习分类

- ✓ 半监督学习：在半监督学习方式下，训练数据有部分被标识，部分没有被标识，这种模型首先需要学习数据的内在结构，以便合理的组织数据来进行预测。算法上，包括一些对常用监督式学习算法的延伸，这些算法首先试图对未标识数据进行建模，在此基础上再对标识的数据进行预测
- ✓ 例：给学生很多未分类的书本与少量的清单，清单上说明哪些书属于同一类别，要求对其他所有书本进行分类

机器学习

- 机器学习分类
  - ✓ 强化学习：主要特点是通过试错来发现最优行为策略而不是带有标签的样本学习
  - ✓ 例：下棋（包括下围棋和象棋）、机器人、自动驾驶等

Scikit-learn



Scikit-learn

- Scikit-learn:
  - ✓ Scikit-learn项目最早由数据科学家David Cournapeau在2007年发起，需要NumPy和SciPy等其他包的支持，是Python语言中专门针对机器学习应用而发展起来的一款开源框架
  - ✓ 和其他众多的开源项目一样，Scikit-learn目前主要由社区成员自发进行维护。可能是由于维护成本的限制，Scikit-learn相比其他项目要显得更为保守。这主要体现在两个方面：一是Scikit-learn从来不做除机器学习领域之外的其他扩展，二是Scikit-learn从来不采用未经广泛验证的算法
  - ✓ http://scikit-learn.org/stable/index.html

Scikit-learn

- Scikit-learn的六大功能：
  - ✓ 基本功能主要被分为六大部分：分类，回归，聚类，数据降维，模型选择和数据预处理

Scikit-learn

- Scikit-learn的六大功能：
  - ✓ 分类是指识别给定对象的所属类别，属于监督学习的范畴，最常见的应用场景包括垃圾邮件检测和图像识别等。目前Scikit-learn已经实现的算法包括：支持向量机（SVM），最近邻，逻辑回归，随机森林，决策树以及多层感知器（MLP）神经网络等

Scikit-learn

- Scikit-learn的六大功能：
  - ✓ 回归是指预测与给定对象相关联的连续值属性，最常见的应用场景包括预测药物反应和预测股票价格等。目前Scikit-learn已经实现的算法包括：支持向量回归（SVR），脊回归，Lasso回归，弹性网络（Elastic Net），最小角回归（LARS ），贝叶斯回归，以及各种不同的鲁棒回归算法等
  - ✓ 回归算法几乎涵盖了所有开发者的需求范围，而且更重要的是，Scikit-learn还针对每种算法都提供了简单明了的用例参考

Scikit-learn

- Scikit-learn的六大功能：
  - ✓ 聚类是指自动识别具有相似属性的给定对象，并将其分组为集合，属于无监督学习的范畴，最常见的应用场景包括顾客细分和试验结果分组。目前Scikit-learn已经实现的算法包括：K-均值聚类，谱聚类，均值偏移，分层聚类，DBSCAN聚类等

Scikit-learn

- Scikit-learn的六大功能：
  - ✓ 数据降维是指使用主成分分析（PCA）、非负矩阵分解（NMF）或特征选择等降维技术来减少要考虑的随机变量的个数，其主要应用场景包括可视化处理和效率提升

Scikit-learn

- Scikit-learn的六大功能：
  - ✓ 模型选择是指对于给定参数和模型的比较、验证和选择，其主要目的是通过参数调整来提升精度。目前Scikit-learn实现的模块包括：格点搜索，交叉验证和各种针对预测误差评估的度量函数

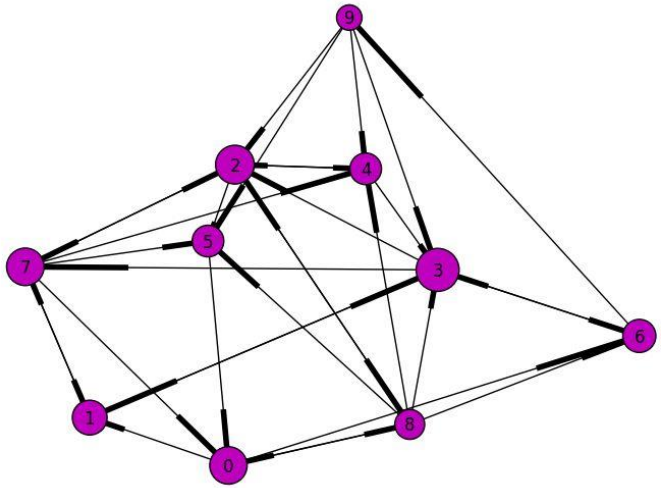
Scikit-learn

- Scikit-learn的六大功能：
  - ✓ 数据预处理是指数据的特征提取和归一化，是机器学习过程中的第一个也是最重要的一个环节
  - ✓ 归一化是指将输入数据转换为具有零均值和单位权方差的新变量，但因为大多数时候都做不到精确等于零，因此会设置一个可接受的范围，一般都要求落在0-1之间。而特征提取是指将文本或图像数据转换为可用于机器学习的数字变量

Scikit-learn

- Scikit-learn:
  - ✓ Scikit-learn针对每个算法和模块都提供了丰富的参考样例和详细的说明文档，据官方的统计大约有200多个。而且为了清晰明白，绝大多数样例都至少给出了一张由Matplotlib绘制的数据图表。这些都是官方提供的学习Scikit-learn框架最直接有效的学习材料

无监督学习

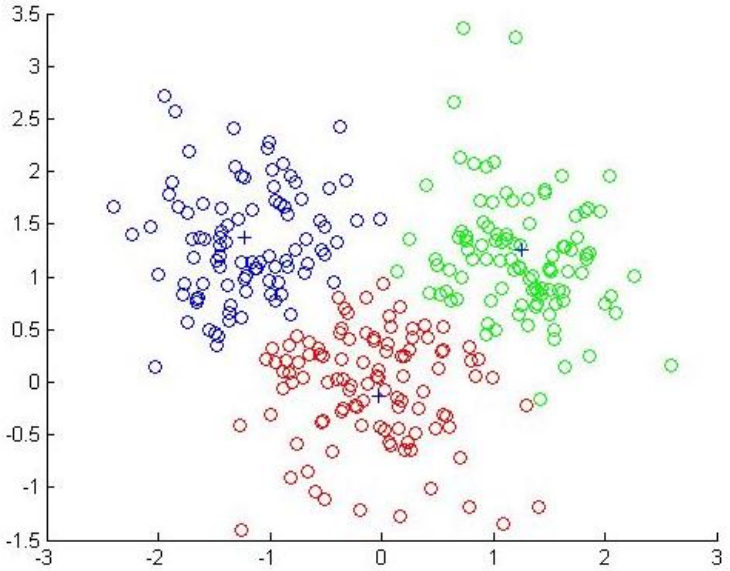


无监督学习

- 无监督学习的目标
  - ✓ 利用无标签的数据学习数据的分布或数据与数据之间的关系被称作无监督学习
  - ✓ 有监督学习和无监督学习的最大区别在于数据是否有标签
  - ✓ 无监督学习最常用的场景是聚类（clustering）和降维（dimension reduction）

无监督学习

- 聚类（clustering）
  - ✓ 根据数据的“相似性”将数据分为多类的过程
  - ✓ 评估两个不同样本之间的“相似性”，通常使用的方法就是计算两个样本之间的“距离”。使用不同的方法计算样本间的距离会关系到聚类结果的好坏



无监督学习

常用的距离概念及计算方法

欧氏距离

欧氏距离是最常用的一种距离度量方法，源于欧式空间中两点的距离。其计算方法如下：

$$d = \sqrt{\sum_{k=1}^{n} (x_{1k} - x_{2k})^2}$$

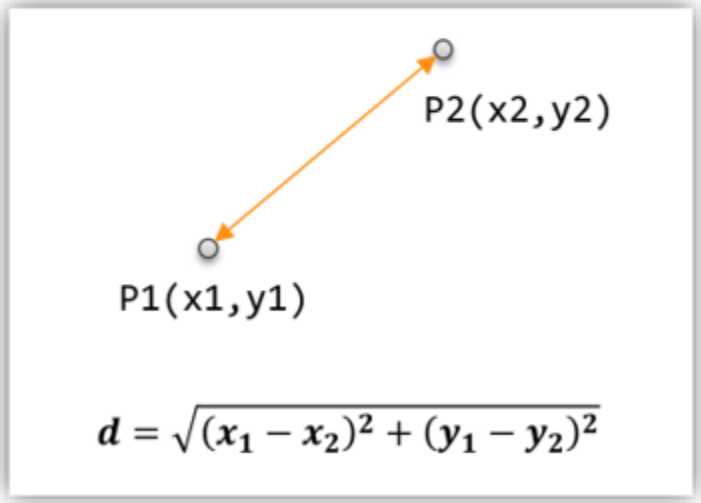


图. 二维空间中欧式距离的计算

无监督学习

- 常用的距离概念及计算方法

曼哈顿距离

曼哈顿距离也称作“城市街区距离”，类似于在城市之中驾车行驶，从一个十字路口到另外一个十字楼口的距离。其计算方法如下：

$$d = \sum_{k=1}^{n} |x_{1k} - x_{2k}|^2$$

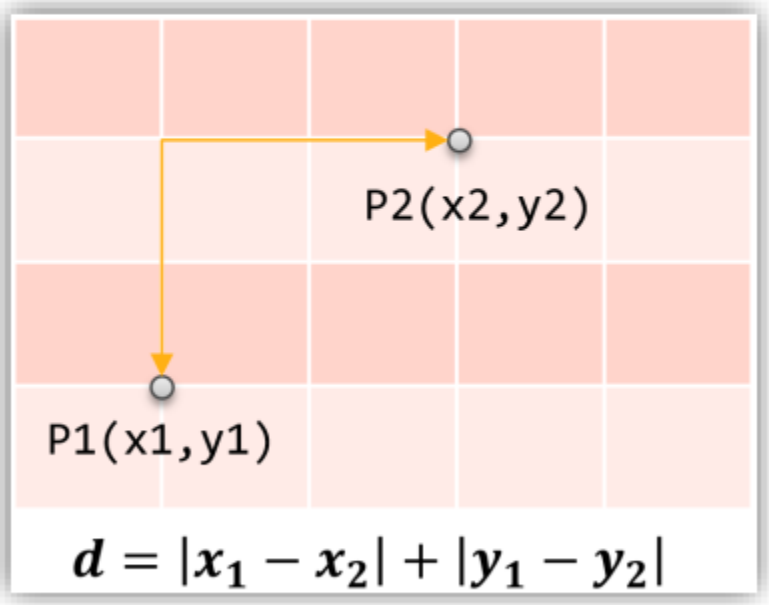


图. 二维空间中曼哈顿距离的计算

无监督学习

- 常用的距离概念及计算方法

马氏距离

马氏距离表示数据的协方差距离，是一种尺度无关的度量方式。也就是说马氏距离会先将样本点的各个属性标准化，再计算样本间的距离。其计算方式如下：（s是协方差矩阵，如图）

$$d(x_i, x_j) = \sqrt{(x_i - x_j)^T s^{-1} (x_i - x_j)}$$

图. 二维空间中的马氏距离

无监督学习

- 常用的距离概念及计算方法

夹角余弦

余弦相似度用向量空间中两个向量夹角的余弦值作为衡量两个样本差异的大小。余弦值越接近1，说明两个向量夹角越接近0度，表明两个向量越相似。其计算方法如下：

$$cos(\theta) = \frac{\sum_{k=1}^{n} x_{1k}x_{2k}}{\sqrt{\sum_{k=1}^{n} x_{1k}^2}\sqrt{\sum_{k=1}^{n} x_{2k}^2}}$$

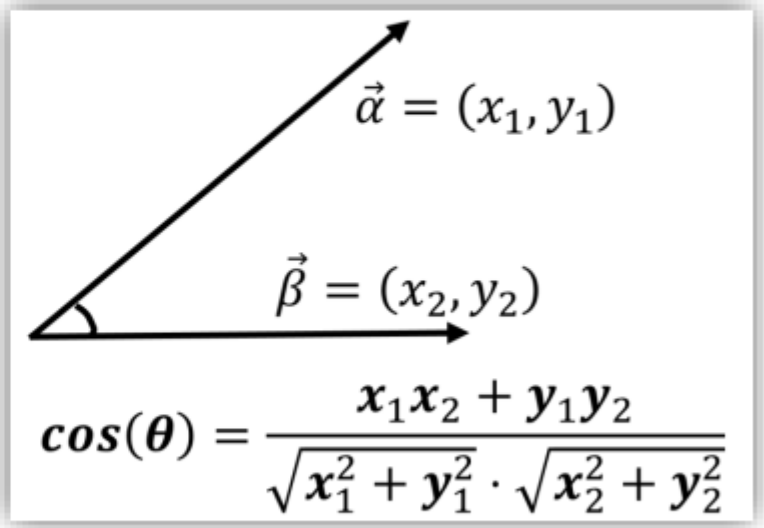
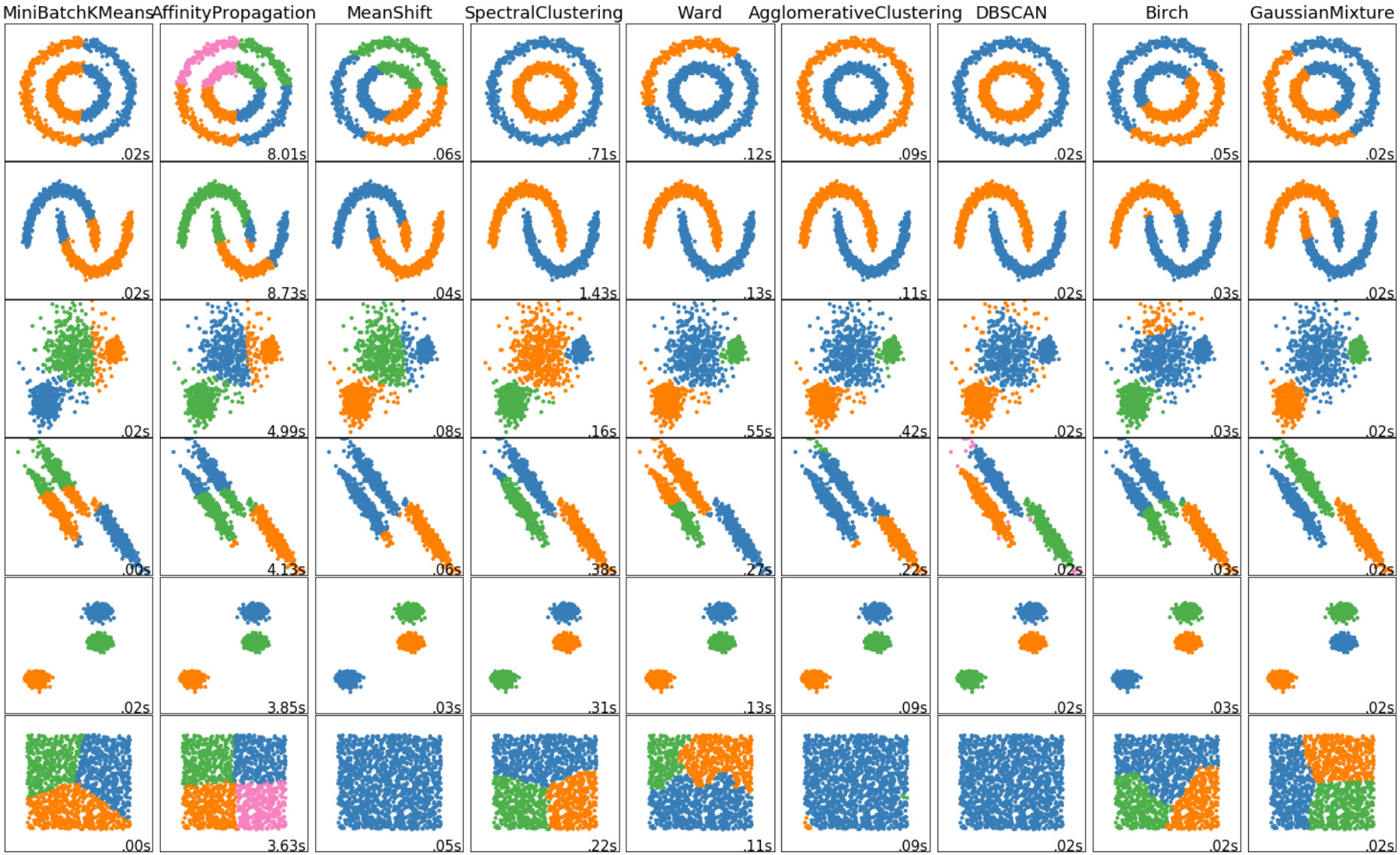


图. 二维空间中的夹角余弦

无监督学习

- Sklearn与聚类
  - ✓ sklearn中的聚类算法包含在sklearn.cluster这个模块中，如：K-Means，近邻传播算法，DBSCAN等
  - ✓ 以同样的数据集应用于不同的算法，可能会得到不同的结果，算法所耗费的时间也不尽相同，这是由算法的特性决定的

无监督学习


MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
Birch
GaussianMixture
.02s 8.01s .06s .71s .12s .09s .02s .05s .02s
.02s 8.73s .04s 1.43s .13s .11s .02s .03s .02s
.02s 4.99s .08s .16s .55s .42s .02s .03s .02s
.00s 4.13s .06s .38s .27s .22s .02s .03s .02s
.02s 3.85s .03s .31s .13s .09s .02s .03s .02s
.00s 3.63s .05s .22s .11s .09s .02s .02s .02s

无监督学习

- Sklearn与聚类
  - ✓ sklearn.cluster模块提供的各聚类算法函数可以使用不同的数据形式作为输入：
  - ✓ 标准数据输入格式：[样本个数，特征个数]定义的矩阵形式
  - ✓ 相似性矩阵输入格式：即由[样本数目，样本数目]定义的矩阵形式，矩阵中的每一个元素为两个样本的相似度，如DBSCAN，AffinityPropagation接受这种输入。如果以余弦相似度为例，则对角线元素全为1．矩阵中每个元素的取值范围为[0,1]

无监督学习

- Sklearn与聚类

算法名称	参数	可扩展性	相似性度量
K-means	聚类个数	大规模数据	点间距离
DBSCAN	邻域大小	大规模数据	点间距离
Gaussian Mixtures	聚类个数及其他超参	复杂度高，不适合处理大规模数据	马氏距离
Birch	分支因子，阈值等其他超参	大规模数据	两点间的欧式距离

无监督学习

- 降维（dimension reduction）
  - ✓ 降维，就是在保证数据所具有的代表性特性或者分布的情况下，将高维数据转化为低维数据的过程，如数据的可视化、精简数据

无监督学习

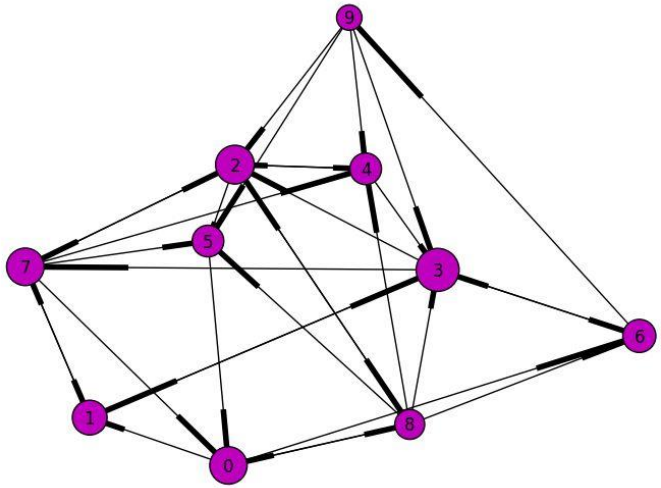
- Sklearn与降维
  - ✓ 降维是机器学习领域的一个重要研究内容，有很多被工业界和学术界接受的典型算法，截止到目前sklearn库提供7种降维算法
  - ✓ 降维过程也可以被理解为对数据集的组成成份进行分解（decomposition）的过程，因此sklearn为降维模块命名为decomposition，在对降维算法调用需要使用sklearn.decomposition模块

无监督学习

- Sklearn与降维

算法名称	参数	可扩展性	适用任务
PCA	所降维度及其他超参	大规模数据	信号处理等
FastICA	所降维度及其他超参	超大规模数据	图形图像特征提取
NMF	所降维度及其他超参	大规模数据	图形图像特征提取
LDA	所降维度及其他超参	大规模数据	文本数据，主题挖掘

K-means方法及应用

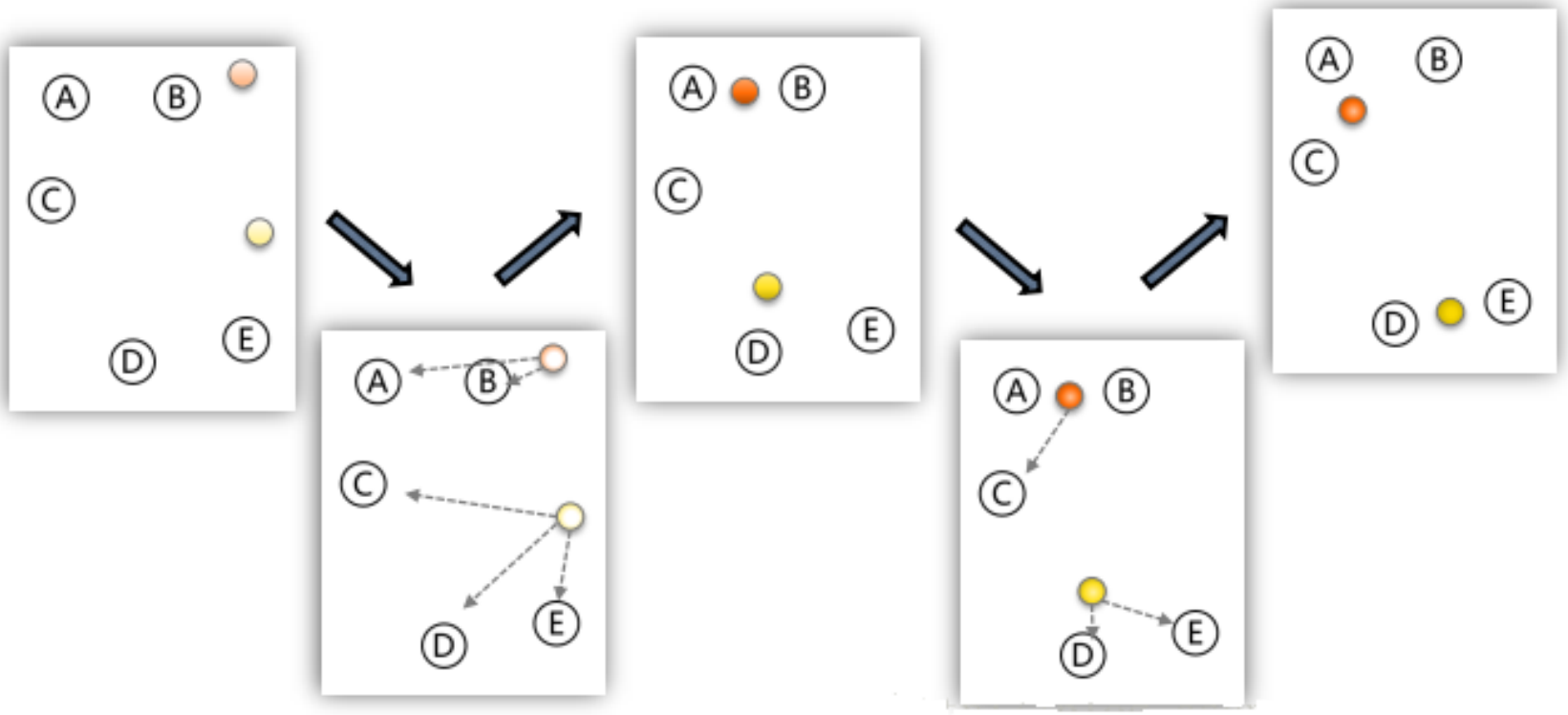


K-means

- K-means算法原理：
  - ✓ k-means算法以k为参数，把n个对象分成k个簇，使簇内具有较高的相似度，而簇间的相似度较低
- 其处理过程如下：
  - ✓ 随机选择k个点作为初始的聚类中心
  - ✓ 对于剩下的点，根据其与聚类中心的距离，将其归入最近的簇
  - ✓ 对每个簇，计算所有点的均值作为新的聚类中心
  - ✓ 重复2、 3直到聚类中心不再发生改变

K-means

- K-means算法原理：



K-means

- K-means的应用：通过聚类，了解1999年各个省份的消费水平在国内的情况
- 数据介绍：
  - ✓ 现有1999年全国31个省份城镇居民家庭平均每人全年消费性支出的八个主要变量数据，这八个变量分别是：食品、衣着、家庭设备用品及服务、医疗保健、交通和通讯、娱乐教育文化服务、居住以及杂项商品和服务。利用已有数据，对31个省份进行聚类

K-means

1999年全国31个省份城镇居民家庭平均每人全年消费性支出数据

城市	食品	衣着	家庭设备用品及服务	医疗保健	交通和通讯	娱乐教育文化服务	居住	杂项商品和服务
北京	2959	730.79	749.41	513.34	467.87	1141.82	478.42	457.64
天津	2460	495.47	697.33	302.87	284.19	735.97	570.84	305.08
河北	1496	515.9	362.37	285.32	272.95	540.58	364.91	188.63
山西	1406	477.77	290.15	208.57	201.5	414.72	281.84	212.1
内蒙古	1304	524.29	254.83	192.17	249.81	463.09	287.87	192.96
辽宁	1731	553.9	246.91	279.81	239.18	445.2	330.24	163.86
吉林	1562	492.42	200.49	218.36	220.69	459.62	360.48	147.76
黑龙江	1410	510.71	211.88	277.11	224.65	376.82	317.61	152.85
上海	3712	550.74	893.37	346.93	527	1034.98	720.33	462.03
江苏	2208	449.37	572.4	211.92	302.09	585.23	429.77	252.54
浙江	2629	557.32	689.73	435.69	514.66	795.87	575.76	323.36
安徽	1845	430.29	271.28	126.33	250.56	513.18	314	151.39
福建	2709	428.11	334.12	160.77	405.14	461.67	535.13	232.29
江西	1564	303.65	233.81	107.9	209.7	393.99	509.39	160.12
山东	1676	613.32	550.71	219.79	272.59	599.43	371.62	211.84
河南	1428	431.79	288.55	208.14	217	337.76	421.31	165.32
湖南	1942	512.27	401.39	206.06	321.29	697.22	492.6	226.45
湖北	1783	511.88	282.84	201.01	237.6	617.74	523.52	182.52
广东	3055	353.23	564.56	356.27	811.88	873.06	1082.82	420.81
广西	2034	300.82	338.65	157.78	329.06	621.74	587.02	218.27
海南	2058	186.44	202.72	171.79	329.65	477.17	312.93	279.19
重庆	2303	589.99	516.21	236.55	403.92	730.05	438.41	225.8
四川	1974	507.76	344.79	203.21	240.24	575.1	430.36	223.46
贵州	1674	437.75	461.61	153.32	254.66	445.59	346.11	191.48
云南	2194	537.01	369.07	249.54	290.84	561.91	407.7	330.95
西藏	2647	839.7	204.44	209.11	379.3	371.04	269.59	389.33
陕西	1473	390.89	447.95	259.51	230.61	490.9	469.1	191.34
甘肃	1526	472.98	328.9	219.86	206.65	449.69	249.66	228.19
青海	1655	437.77	258.78	303	244.93	479.53	288.56	236.51
宁夏	1375	480.89	273.84	317.32	251.08	424.75	228.73	195.93
新疆	1609	536.05	432.46	235.82	250.28	541.3	344.85	214.4

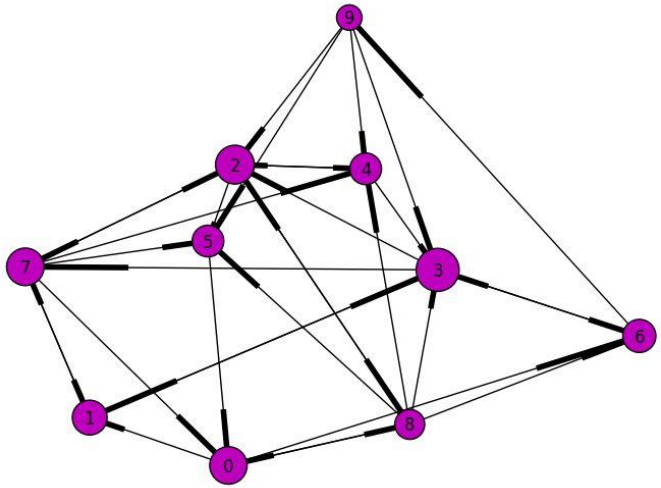
K-means

- 调用KMeans方法所需参数：
  - ✓ n_clusters：用于指定聚类中心的个数
  - ✓ init：初始聚类中心的初始化方法
  - ✓ max_iter：最大的迭代次数
  - ✓ 一般调用时只用给出n_clusters即可，init默认是k-means++，max_iter默认是300

K-means

- Sklearn中KMeans算法的改进
  - ✓ 计算两条数据相似性时，Sklearn的K-Means默认用的是欧式距离（euclidean_distances），可以考虑修改euclidean_distances，实现基于其他距离的聚类

DBSCAN方法及应用



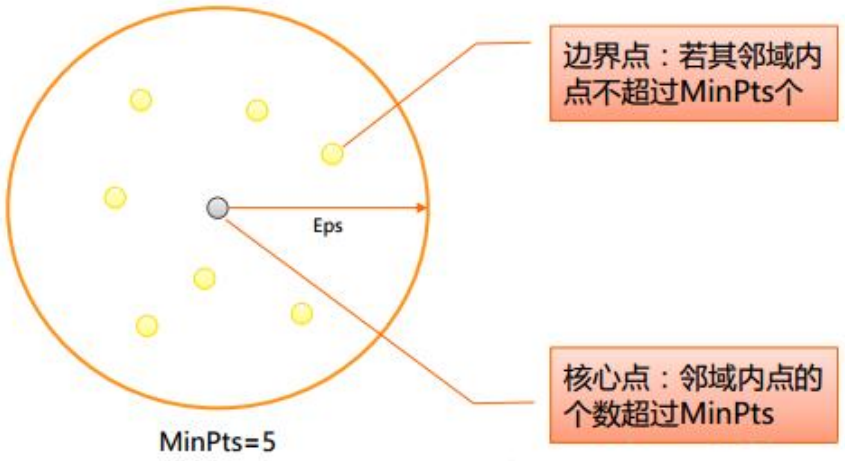
DBSCAN

- DBSCAN介绍
  - ✓ DBSCAN（Density-Based Spatial Clustering of Applications with Noise）算法是一种基于密度的聚类算法
  - ✓ 聚类的时候不需要预先指定簇的个数
  - ✓ 最终的簇的个数不定

DBSCAN

DBSCAN算法原理

- ✓ DBSCAN算法将数据点分为三类：
- ✓ 核心点：在半径Eps内含有超过MinPts数目的点
- ✓ 边界点：在半径Eps内点的数量小于MinPts，但是落在核心点的邻域内
- ✓ 噪音点：既不是核心点也不是边界点的点



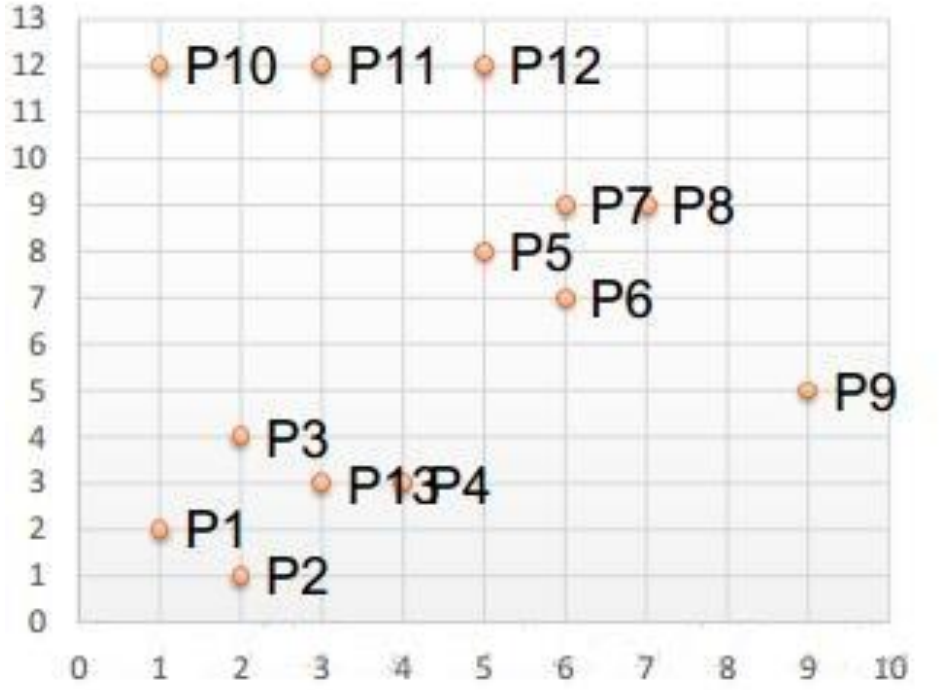
DBSCAN

- DBSCAN算法流程
  - ✓ 将所有点标记为核心点、边界点或噪声点
  - ✓ 删除噪声点
  - ✓ 为距离在Eps之内的所有核心点之间赋予一条边
  - ✓ 每组连通的核心点形成一个簇
  - ✓ 将每个边界点指派到一个与之关联的核心点的簇中

DBSCAN

- DBSCAN算法举例

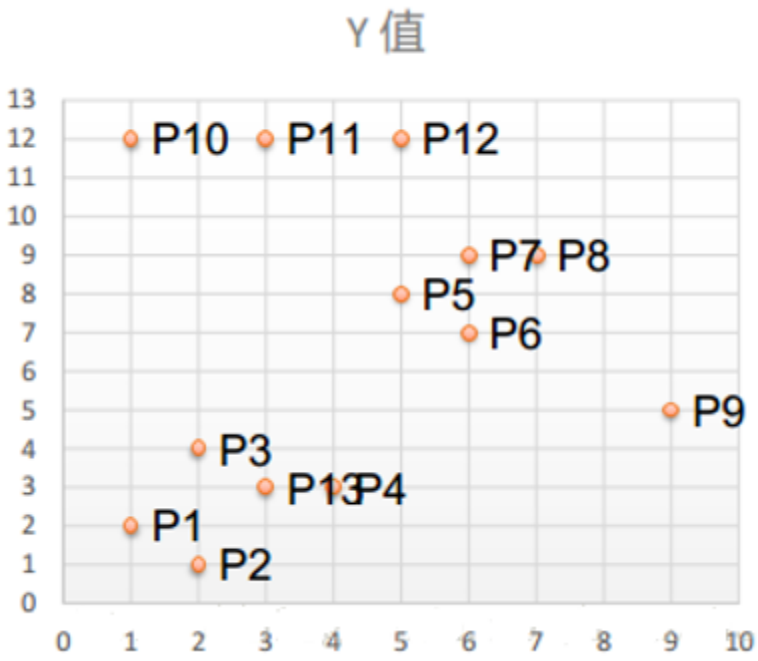
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
X	1	2	2	4	5	6	6	7	9	1	3	5	3
Y	2	1	4	3	8	7	9	9	5	12	12	12	3



DBSCAN

- DBSCAN步骤1：

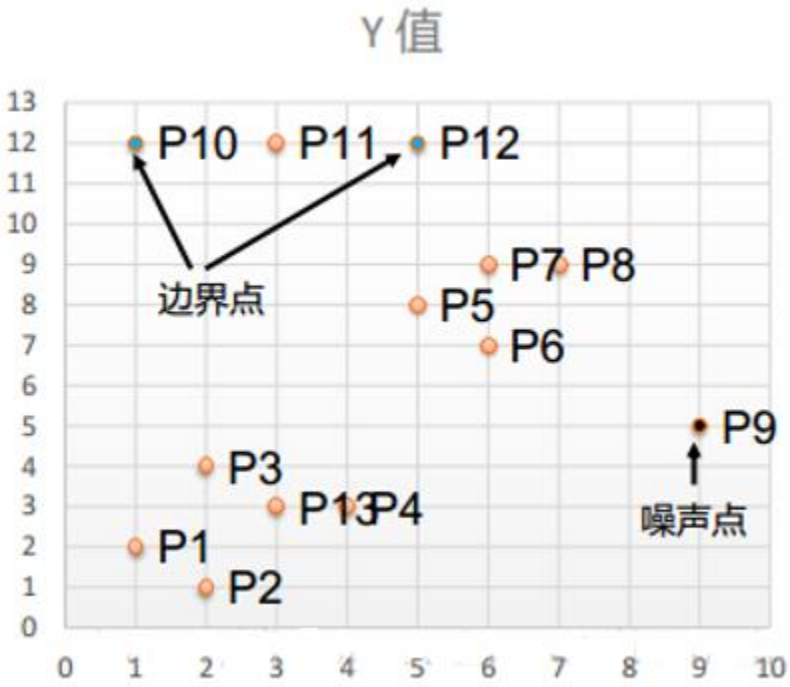
取Eps=3，MinPts=3，依据DBSACN对所有点进行聚类（曼哈顿距离）。



DBSCAN

DBSCAN步骤2：

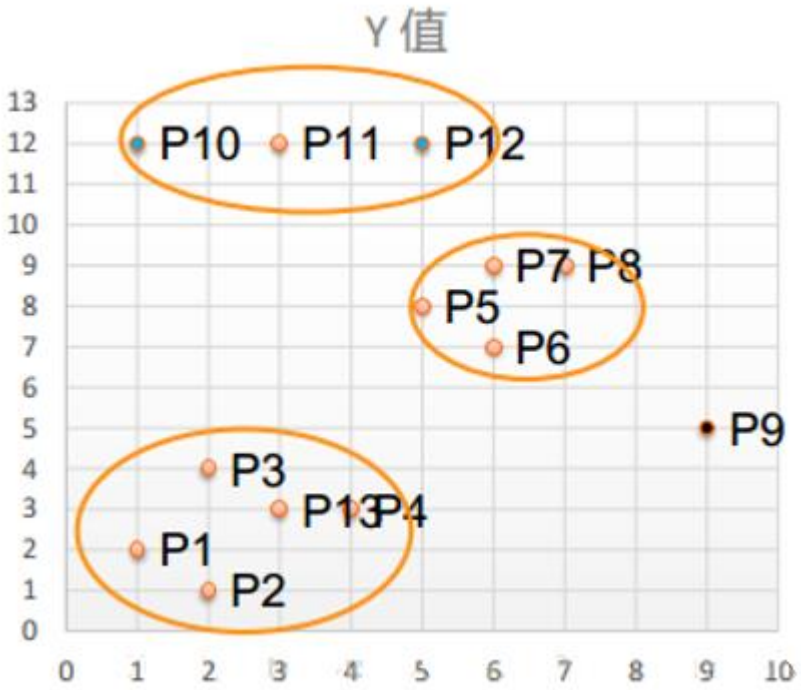
- 对每个点计算其邻域Eps=3内的点的集合。
- 集合内点的个数超过MinPts=3的点为核心点
- 查看剩余点是否在核心点的邻域内，若在，则为边界点，否则为噪声点。



DBSCAN

- DBSCAN步骤3：

将距离不超过Eps=3的点相互连接，构成一个簇，核心点邻域内的点也会被加入到这个簇中。则右侧形成3个簇。



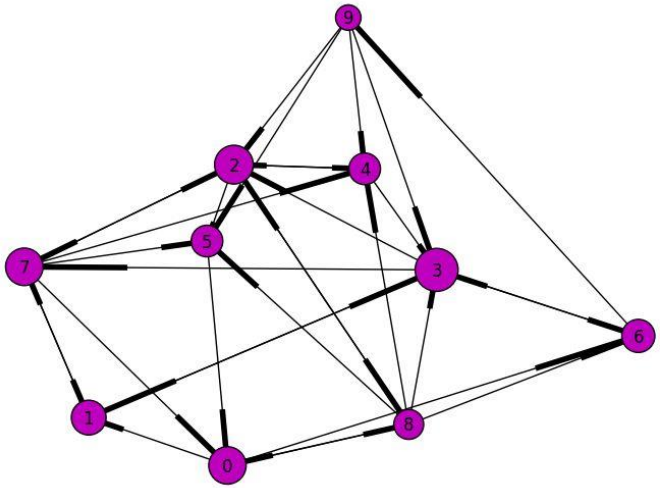
DBSCAN

- DBSCAN应用
  - ✓ 数据介绍：现有大学校园网的日志数据，100条大学生的校园网使用情况数据，数据包括用户ID，设备的MAC地址，IP地址，开始上网时间，停止上网时间，上网时长，校园网套餐等。 利用已有数据，分析学生上网的模式

DBSCAN

学生上网日志（单条数据格式）	
记录编号	2c929293466b97a6014754607e457d68
学生编号	U201215025
MAC地址	A417314EEA7B
IP地址	10.12.49.26
开始上网时间	2014-07-20 22:44:18.540000000
停止上网时间	2014-07-20 23:10:16.540000000
上网时长	1558

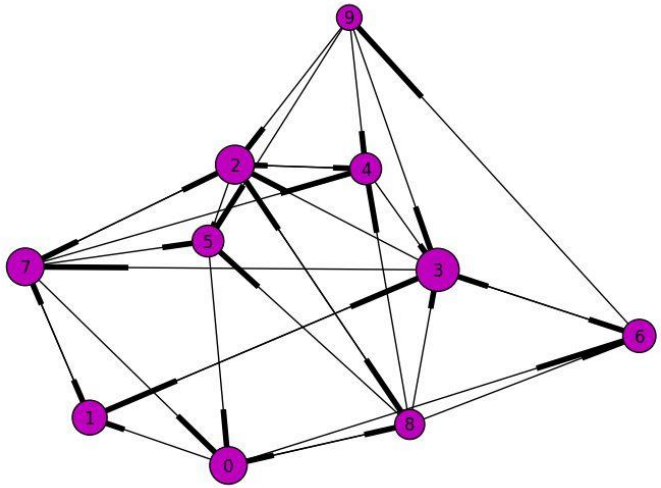
聚类方法比较



聚类方法比较

- #clustering algorithms comparing

主成分分析及其应用



主成分分析

- 主成分分析（Principal Component Analysis，PCA）是最常用的一种降维方法，通常用于高维数据集的探索与可视化，还可以用作数据压缩和预处理等
- PCA可以把具有相关性的高维变量合成为线性无关的低维变量，称为主成分。主成分能够尽可能保留原始数据的信息

主成分分析

- 方差：是各个样本和样本均值的差的平方和的均值，用来度量一组数据的分散程度

$$S^2 = \frac{\sum_{i=1}^{n}(x_i - x)^2}{n-1}$$

- 协方差：用于度量两个变量之间的线性相关性程度，若两个变量的协方差为0，则可认为二者线性无关

$$Cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{n-1}$$

主成分分析

- 协方差矩阵：协方差矩阵则是由变量的协方差值构成的矩阵（对称阵）
- 特征向量和特征值：矩阵的特征向量是描述数据集结构的非零向量，并满足如下公式，A是方阵，v是特征向量，lamda是特征值

$$A\vec{v} = \lambda\vec{v}$$

主成分分析

PCA原理

- ✓ 矩阵的主成分就是其协方差矩阵对应的特征向量，按照对应的特征值大小进行排序，最大的特征值就是第一主成分，其次是第二主成分，以此类推

输入： 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
低维空间维数 d'.

过程：

1: 对所有样本进行中心化: $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i$;
2: 计算样本的协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$;
3: 对协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ 做特征值分解;
4: 取最大的 d' 个特征值所对应的特征向量 $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'}$.

输出： 投影矩阵 $\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'})$.

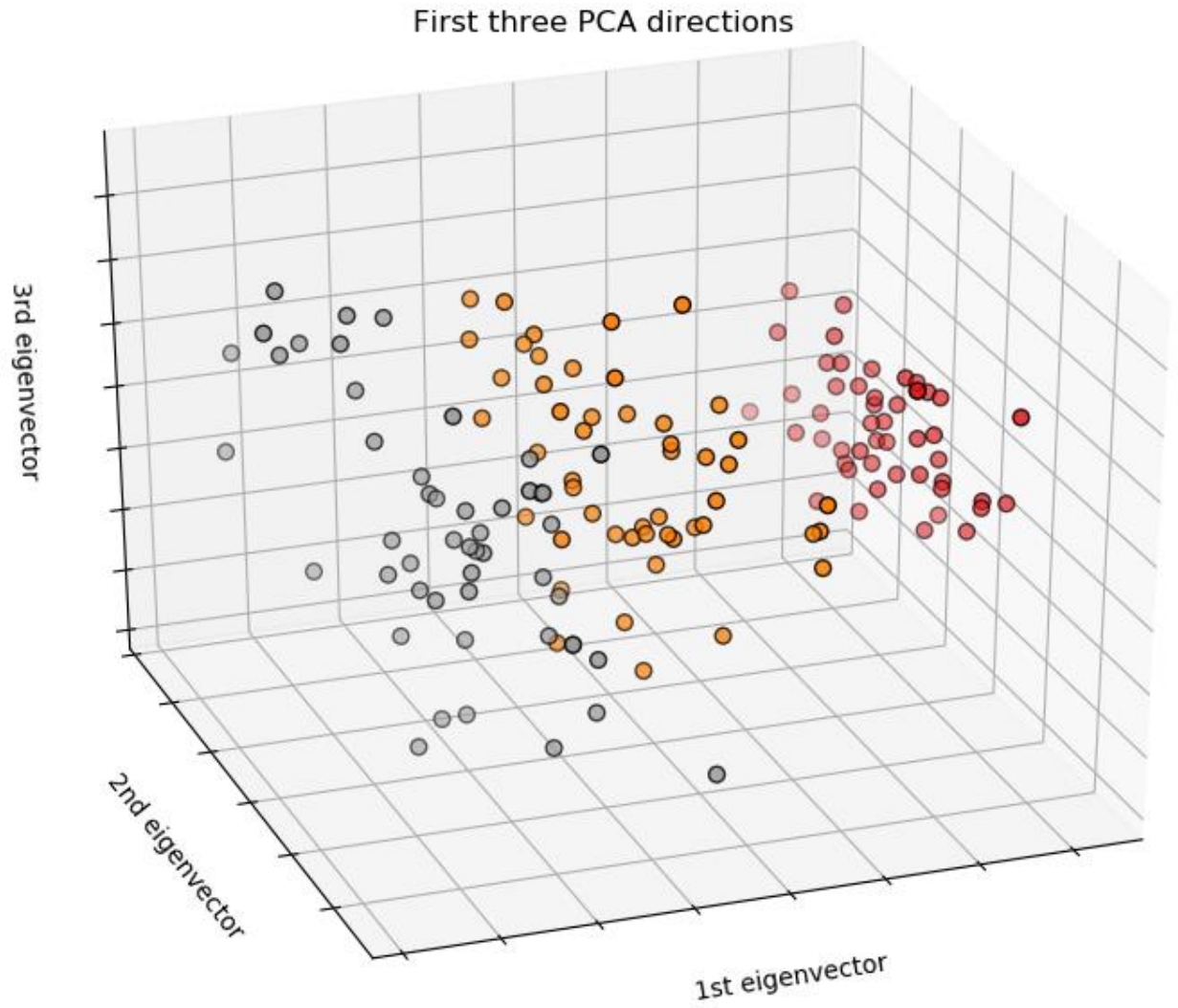
主成分分析

- sklearn中主成分分析
  - ✓ 在sklearn库中，可以使用sklearn.decomposition.PCA加载PCA进行降维
  - ✓ 主要参数有：
  - ✓ n_components：指定主成分的个数，即降维后数据的维度
  - ✓ svd_solver：设置特征值分解的方法，默认为‘auto’，其他可选有‘full’，‘arpack’，‘randomized’

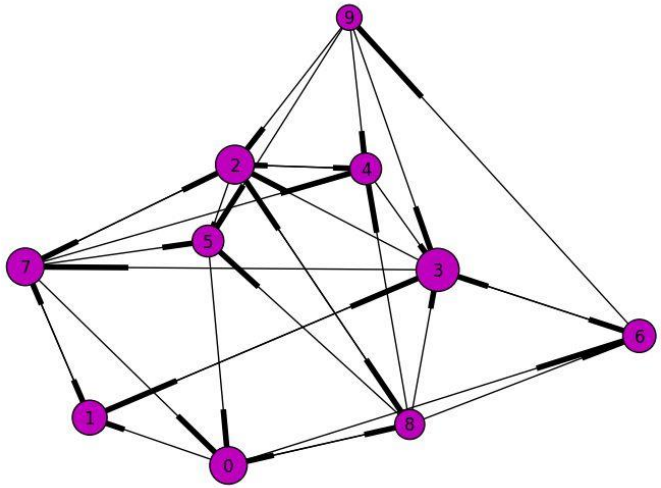
主成分分析

- sklearn中主成分分析
  - ✓ 已知鸢尾花数据是4维的，共三类样本。使用PCA实现对鸢尾花数据进行降维，实现在二维平面上的可视化
  - ✓ 数据集包含150个数据集，分为3类，每类50个数据，每个数据包含4个属性。可通过花萼长度，花萼宽度，花瓣长度，花瓣宽度4个属性预测鸢尾花卉属于（Setosa，Versicolour，Virginica）三个种类中的哪一类
  - ✓ #PCA_1

主成分分析


First three PCA directions
3rd eigenvector
2nd eigenvector
1st eigenvector

NMF方法及其应用



NMF

- NMF
  - ✓ 非负矩阵分解（Non-negative Matrix Factorization，NMF）是在矩阵中所有元素均为非负数约束条件之下的矩阵分解方法

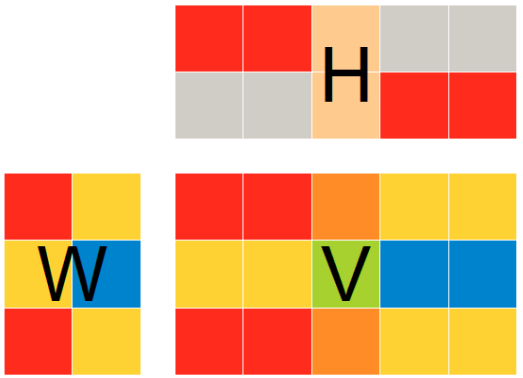
NMF

NMF基本思想

- ✓ 给定一个非负矩阵V，NMF能够找到一个非负矩阵W和一个非负矩阵H，使得矩阵W和H的乘积近似等于矩阵V中的值

$$V_{n*m} = W_{n*k} * H_{k*m}$$

- ✓ W矩阵：基础图像矩阵，相当于从原矩阵V中抽取出来的特征。H矩阵：系数矩阵
- ✓ 对应关系如图所示：



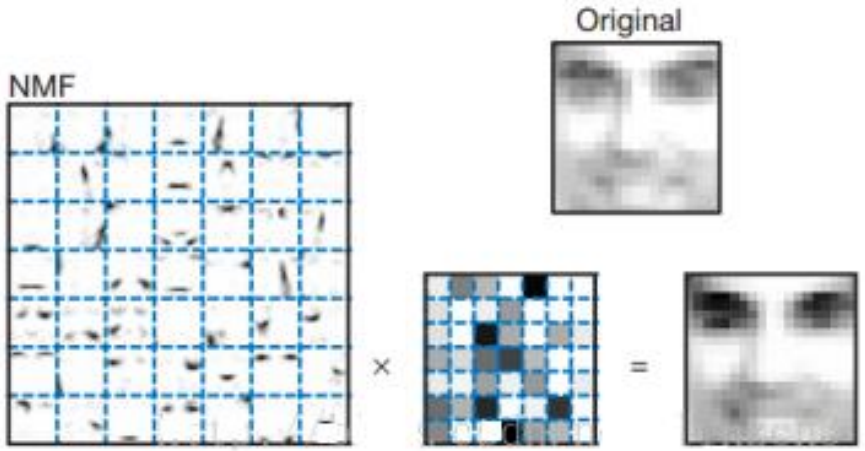
$V \approx WH;$

- $\mathbf{W} = [w_{fk}]$ s.t. $w_{fk} \geq 0$ and
- $\mathbf{H} = [h_{kn}]$ s.t. $h_{kn} \geq 0$.

NMF

NMF基本思想

- ✓ 左侧为W矩阵，可以看出从原始图像中抽取出来的特征。中间的是H矩阵，表示系数
- ✓ 可以发现乘积结果与原结果是很像的



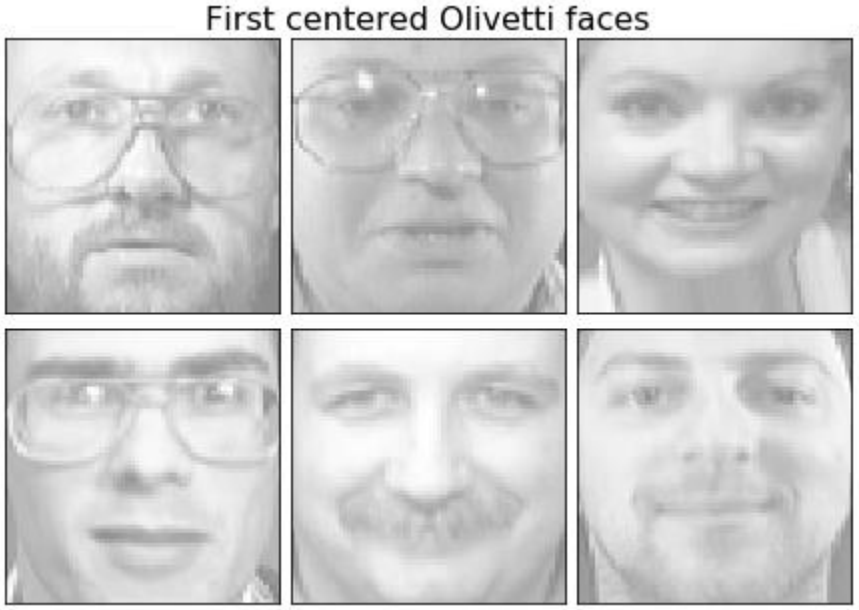
- ✓ 矩阵分解优化目标：最小化W矩阵H矩阵的乘积和原始矩阵之间的差别

NMF

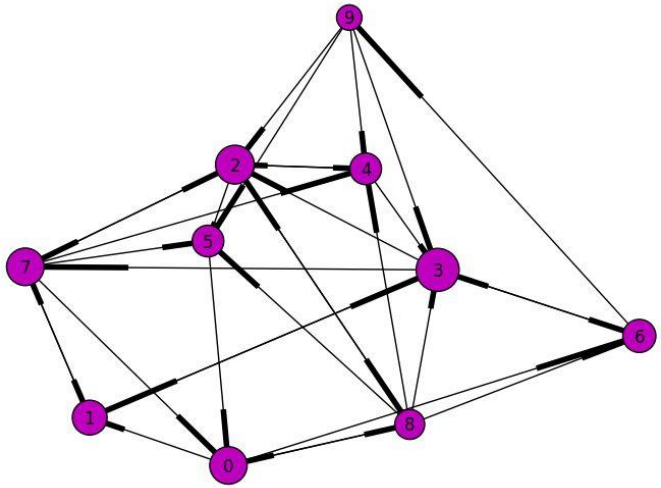
- sklearn中非负矩阵分解
  - ✓ 在sklearn库中，可以使用sklearn.decomposition.NMF加载NMF算法，主要参数有：
  - ✓ n_components：用于指定分解后矩阵的单个维度k
  - ✓ init：W矩阵和H矩阵的初始化方式，默认为‘nndsvdar’

NMF

- sklearn中非负矩阵分解
  - ✓ 数据集：已知Olivetti人脸数据共400个，每个数据是64*64大小。由于NMF分解得到的W矩阵相当于从原始矩阵中提取的特征，那么就可以使用NMF对400个人脸数据进行特征提取



基于聚类的整图分割实例



基于聚类的整图分割

- 图像分割：
  - ✓ 利用图像的灰度、颜色、纹理、形状等特征，把图像分成若干个互不重叠的区域，并使这些特征在同一区域内呈现相似性，在不同的区域之间存在明显的差异性。然后就可以将分割的图像中具有独特性质的区域提取出来用于不同的研究

基于聚类的整图分割

- 图像分割用途：
  - ✓ 图像分割技术已在实际生活中得到广泛的应用。例如：在机车检验领域，可以应用到轮毂裂纹图像的分割，及时发现裂纹，保证行车安全；在生物医学工程方面，对肝脏CT图像进行分割，为临床治疗和病理学研究提供帮助

基于聚类的整图分割

- 图像分割常用方法：
  - ✓ 阈值分割：对图像灰度值进行度量，设置不同类别的阈值，达到分割的目的
  - ✓ 边缘分割：对图像边缘进行检测，即检测图像中灰度值发生跳变的地方，则为一片区域的边缘
  - ✓ 直方图法：对图像的颜色建立直方图，而直方图的波峰波谷能够表示一块区域的颜色值的范围，来达到分割的目的
  - ✓ 特定理论：基于聚类分析、小波变换等理论完成图像分割

基于聚类的整图分割

- 图像分割
  - ✓ 目标：利用K-means聚类算法对图像像素点颜色进行聚类实现简单的图像分割
  - ✓ 输出：同一聚类中的点使用相同颜色标记，不同聚类颜色不同
  - ✓ 技术路线：sklearn.cluster.Kmeans
  - ✓ # knn_pic

谢谢