AP 聚类算法

1.分类与聚类

1.1 分类算法简介

分类(classification)是找出描述并区分数据类或概念的模型(或函数)，以便能够使用模型预测类标记未知的对象类。

在分类算法中输入的数据，或称训练集（Training Set），是一条条的数据库记录（Record）组成的。每一条记录包含若干条属性（Attribute），组成一个特征向量。训练集的每条记录还有一个特定的类标签（Class Label）与之对应。该类标签是系统的输入，通常是以往的一些经验数据。一个具体样本的形式可为样本向量:(v_1, v_2, ..., v_n; c)。在这里 v_i 表示字段值,c 表示类别。

分类的目的是：分析输入的数据，通过--在训练集中的数据表现出来的特性，为每一个类找到一种准确的描述或者模型。这种描述常常用谓词表示。由此生成的类描述用来对未来的测试数据进行分类。尽管这些未来的测试数据的类标签是未知的，我们仍可以由此预测这些新数据所属的类。注意是预测，而不能肯定。我们也可以由此对数据中的每一个类有更好的理解。也就是说：我们获得了对这个类的知识。

下面对分类流程作个简要描述：

训练：训练集——>特征选取——>训练——>分类器

分类：新样本——>特征选取——>分类——>判决

常见的分类算法有：**决策树、KNN 法(K-Nearest Neighbor)、SVM 法、VSM 法、Bayes 法、神经网络**等。

1.2 聚类算法简介

聚类(clustering)是指根据“物以类聚”的原理，将本身没有类别的样本聚集成不同的组，这样的一组数据对象的集合叫做簇，并且对每一个这样的簇进行描述的过程。

与分类规则不同，进行聚类前并不知道将要划分成几个组和什么样的组，也不知道根据哪些空间区分规则来定义组。

它的目的是使得属于同一个簇的样本之间应该彼此相似，而不同簇的样本应

该足够不相似。

聚类分析的算法可以分为：**划分法（Partitioning Methods）、层次法（Hierarchical Methods）、基于密度的方法（density-based methods）、基于网格的方法（grid-based methods）、基于模型的方法（Model-Based Methods）**。

经典的 K-means 和 K-centers 都是划分法。

分类与聚类的区别

聚类分析也称无监督学习或无指导学习，聚类的样本没有标记，需要由聚类学习算法来自动确定;在分类中，对于目标数据库中存在哪些类是知道的，要做的就是将每一条记录分别属于哪一类标记出来。聚类学习是观察式学习，而不是示例式学习。

可以说**聚类分析可以作为分类分析的一个预处理步骤**。

2.K-MEANS 算法

k-means 算法接受输入量 k；然后将 n 个数据对象划分为 k 个聚类以便使得所获得的聚类满足：同一聚类中的对象相似度较高；而不同聚类中的对象相似度较低。簇的相似度是关于簇中对象的均值度量，可以看作簇的质心(centriod)或重心(center of gravity)。

k-means 算法的工作过程说明如下：首先从 n 个数据对象任意选择 k 个对象作为初始聚类中心；而对于所剩下其它对象，则根据它们与这些聚类中心的相似度（距离），分别将它们分配给与其最相似的（聚类中心所代表的）聚类；然后再计算每个所获新聚类的聚类中心（该聚类中所有对象的均值）；不断重复这一过程直到标准测度函数开始收敛为止。一般都采用均方差作为标准测度函数，其定义如下：

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} | p - m_i |^2 \qquad (1)$$

其中，E 是数据集中所有对象的平方误差和，p 是空间中的点，表示给定对象，m_i 是簇 C_i 的均值（p 和 m_i 都是多维的）。换句话说，对于每个簇中的每个对象，求对象到其簇中心距离的平方，然后求和。这个准则试图使生成的 k 个结果簇尽可能的紧凑和独立。

例 1:我们在二维空间中随机的生成 20 个数据点，将聚类数目指定为 5 个，并随机生成一个聚类中心(用“×”来标注)，根据对象与簇中心的距离，每个对象分成于最近的簇。初始示例图如下：

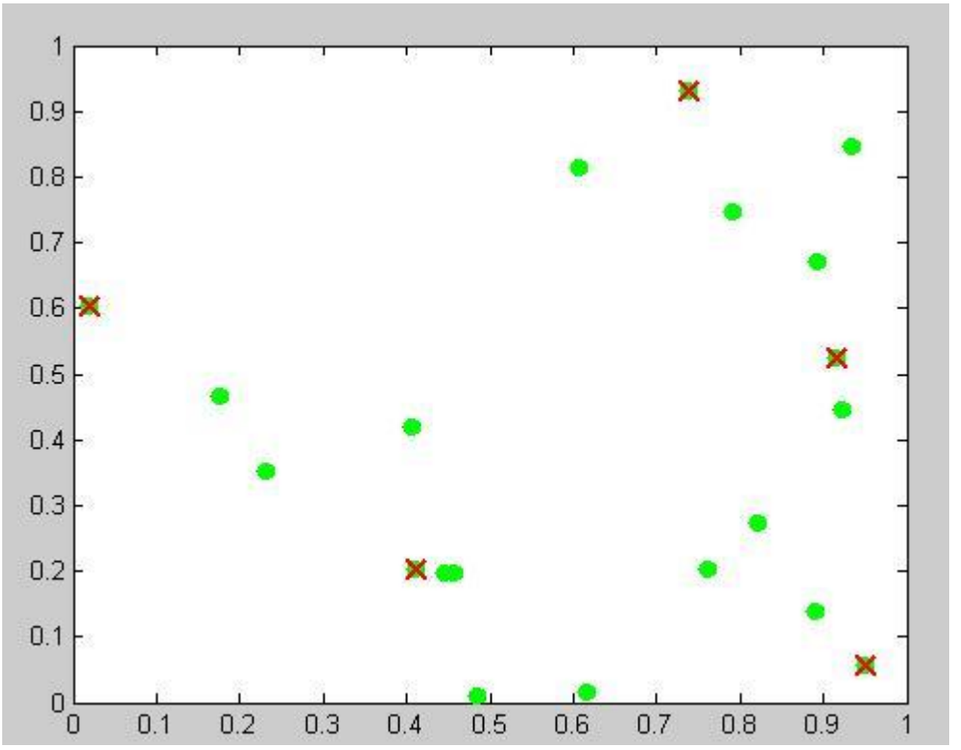

图 1.随机生成的数据点及初始聚类中心示例图

下一步，更新簇中心。也就是说，根据簇中的当前对象，重新计算每个簇的均值。使用这些新的簇中心，将对象重新分成到簇中心最近的簇中。

不断迭代上面的过程，直到簇中对象的重新分布不再发生，处理结束。最终的聚类结果示例图如下：

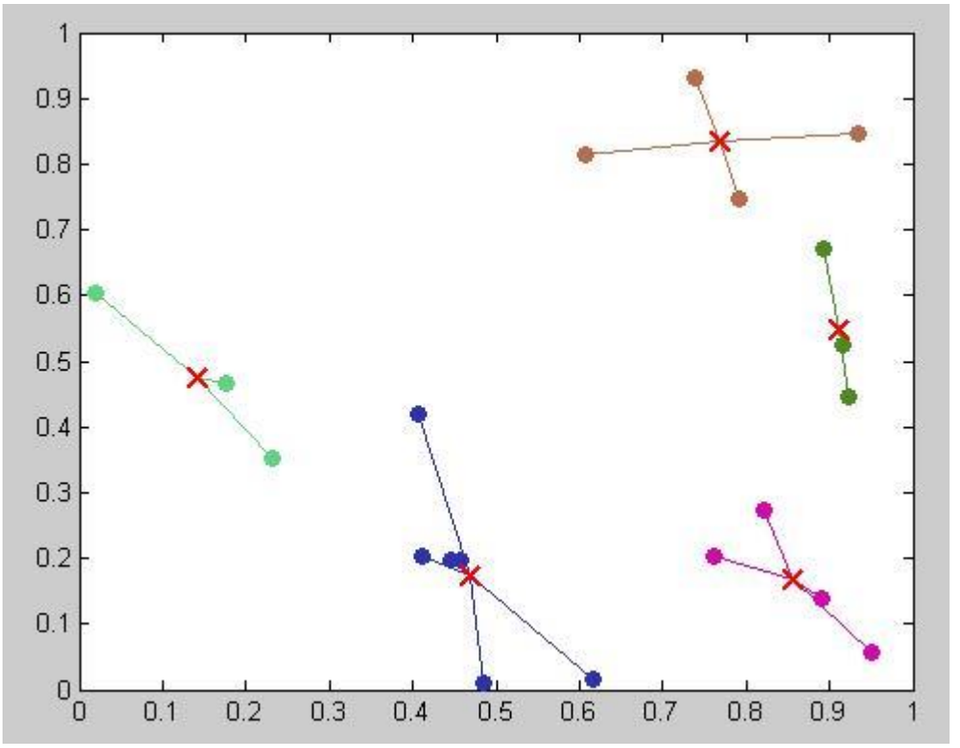

图 2. 最终聚类结果示例图

从上图中我们可以看到，最终的聚类结果受初始聚类中心的影响很大，而且最后的簇质心点不一定是在数据点上。

K 均值算法试图确定最小化平方误差的 k 个划分。当结果簇是紧凑的，并且簇与簇之间明显分离时，它的效果较好。**对处理大数据集，该算法是相对可伸缩的和有效率的**，因为它的计算复杂度是 O（nkt），其中 n 是对象的总数，k 是簇的个数，t 是迭代的次数。通常地，k<<n 并且 t<<n。该方法经常终止于局部最优解。

然而，只有**当簇均值有定义的情况下 k 均值方法才能使用**。在某些应用中，例如当涉及具有分类属性的数据时，均值可能无定义。用户必须事先给出要生成的簇的数目 k 可以算是该方法的缺点。**K 均值方法不适合于发现非凸形状的簇**，或者**大小差别很大的簇**。此外，它**对于噪声和离群点数据是敏感**的，因为少量的这类数据能够对均值产生极大的影响。

3.AP 算法

Affinity Propagation(AP)聚类是最近在Science杂志上提出的一种新的聚类算法。它根据N个数据点之间的相似度进行聚类,这些相似度可以是对称的,即两个数据点互相之间的相似度一样(如欧氏距离);也可以是不对称的,即两个数据点互相之间的相似度不等。这些相似度组成N ×N的相似度矩阵S(其中N为有N个数据点)。AP

算法不需要事先指定聚类数目,相反它将所有的数据点都作为潜在的聚类中心,称之为**exemplar**。

以S矩阵的对角线上的数值s(k,k)作为k点能否成为聚类中心的评判标准,这意味着该值越大,这个点成为聚类中心的可能性也就越大,这个值又称作**参考度p (preference)** 。聚类的数量受到参考度p的影响,如果认为每个数据点都有可能作为聚类中心,那么p就应取相同的值。如果取输入的相似度的均值作为p的值,得到聚类数量是中等的。如果取最小值,得到类数较少的聚类。AP算法中传递两种类型的消息,(responsiility)和(availability)。r(i,k)表示从点i发送到候选聚类中心k的数值消息,反映**k点是否适合作为i点的聚类中心**。a(i,k)则从候选聚类中心k发送到i的数值消息,反映**i点是否选择k作为其聚类中心**。r(i,k)与a (i,k)越强,则k点作为聚类中心的可能性就越大,并且i点隶属于以k点为聚类中心的聚类的可能性也越大。**AP算法通过迭代过程不断更新每一个点的吸引度和归属度值,直到产生m个高质量的exemplar,同时将其余的数据点分配到相应的聚类中**。

在这里介绍几个文中常出现的名词：

exemplar：指的是聚类中心。

similarity：数据点i和点j的相似度记为S(i，j)。是指点j作为点i的聚类中心的相似度。一般使用欧氏距离来计算，如$-\|(x_i-x_j)^2+(y_i-y_j)^2\|$。文中，所有点与点的相似度值全部取为负值。因为我们可以看到，相似度值越大说明点与点的距离越近，便于后面的比较计算。

preference：数据点i的参考度称**为P(i)**或**S(i,i)**。是指点i作为聚类中心的参考度。一般取S相似度值的中值。

Responsibility:R(i,k)用来描述点k适合作为数据点i的聚类中心的程度。

Availability:A(i,k)用来描述点i选择点k作为其聚类中心的适合程度。

两者的关系如下图：

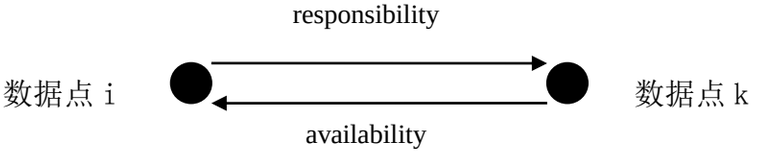

图3. 数据点之间传递消息示意图

下面是R与A的计算公式：

$$R(i,k)=S(i,k)-\max\{A(i,j)+S(i,j)\}(j\in\{1,2,\ldots\ldots,N,但j\neq k\}) \quad (2)$$

$$A(i,k)=\min\{0,R(k,k)+\sum_{j}\{\max(0,R(j,k))\}\} \quad (j\in\{1,2,\ldots\ldots,N,但j\neq i且j\neq k\}) \quad (3)$$

$$R(k,k)=P(k)-\max\{A(k,j)+S(k,j)\}\ (j\in\{1,2,\ldots\ldots,N,但j\neq k\}) \quad (4)$$

由上面的公式可以看出，当P(k)较大使得R(k, k)较大时, A(i, k)也较大,从而类代表k作为最终聚类中心的可能性较大;同样, 当越多的P(i)较大时,越多的类代表倾向于成为最终的聚类中心。因此,增大或减小P可以增加或减少AP输出的聚类数目。

Damping factor(阻尼系数):主要是起收敛作用的。文中讲述，每次迭代，吸引度 R_i 和归属度 A_i 要与上一次的 R_{i-1} 和 A_{i-1} 进行加权更新。公式如下：

$$R_i=(1\text{-lam})*R_i+\text{lam}*R_{i-1} \quad (5)$$

$$A_i=(1\text{-lam})*A_i+\text{lam}*A_{i-1} \quad (6)$$

其中，$\text{lam}\in[0.5,1)$。

AP算法的具体工作过程如下：先计算N个点之间的相似度值，将值放在S矩阵中，再选取P值(一般取S的中值)。设置一个最大迭代次数(文中设默认值为1000)，迭代过程开始后，计算每一次的R值和A值，根据R(k,k)+A(k,k)值来判断是否为聚类中心(文中指定当(R(k,k)+A(k,k))＞0时认为是一个聚类中心)，当迭代次数超过最大值(即maxits值)或者当聚类中心连续多少次迭代不发生改变(即convits值)时终止计算(文中设定连续50次迭代过程不发生改变是终止计算)。

例2：我们在二维空间中随机的生成20个数据点，将P值设为S矩阵的中值，将convits值设置成20, maxits值设置成1000，用AP算法进行计算，最终聚类结果如下图：

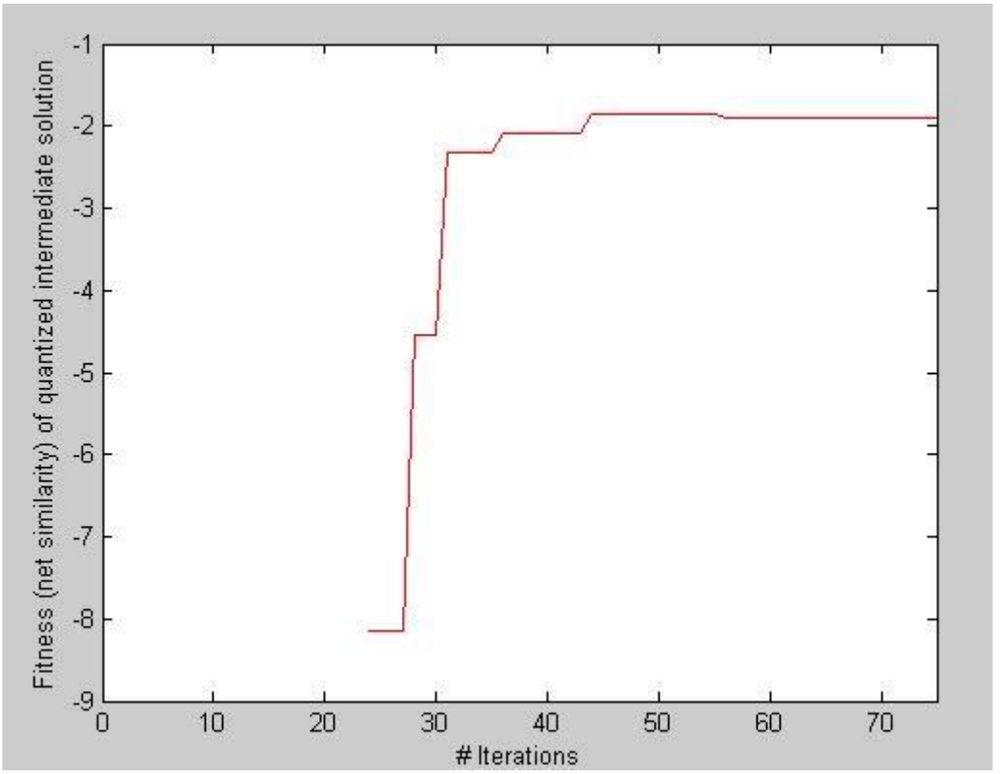

图4. AP算法迭代过程示意图

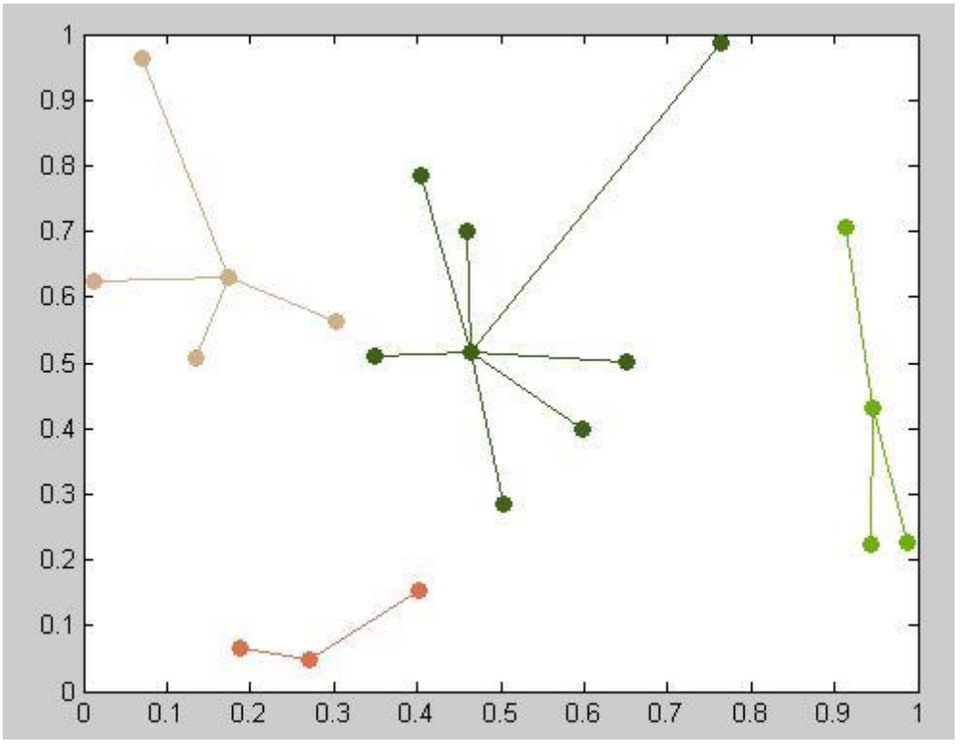

图5 最终聚类结果示意图

在AP 算法中，迭代次数和聚类数目主要受到两个参数的影响。其中，聚数数目主要受preference值（负值）的影响。下面对同一组数据集(200个数据点)进行计

算，取不同的preference值得到的聚类数目如下：

Preference值	聚类数目
$\frac{median(S)}{2}$	16
median(S)	11
2×median(S)	8

表1.不同的preference得到的聚类数目比较

由表1，我们可以看出，当preference越大时，得到的聚类数目越多。

当取不同的**lam（阻尼系数）**值时，迭代次数和迭代过程中数据的摆动都会有很大的不同，下面同样是对同一组数据集(200个数据点)进行计算，取有代表性的两个值（0.5和0.9）进行比较结果如下：

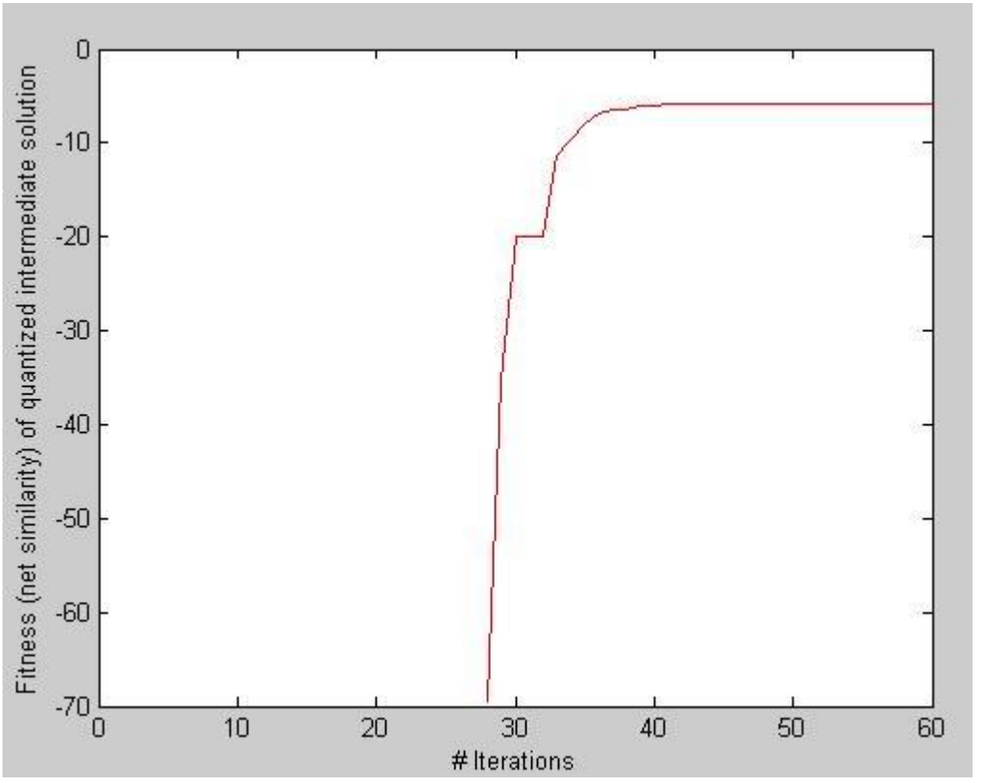

图6 lam取0.9时的迭代示意图

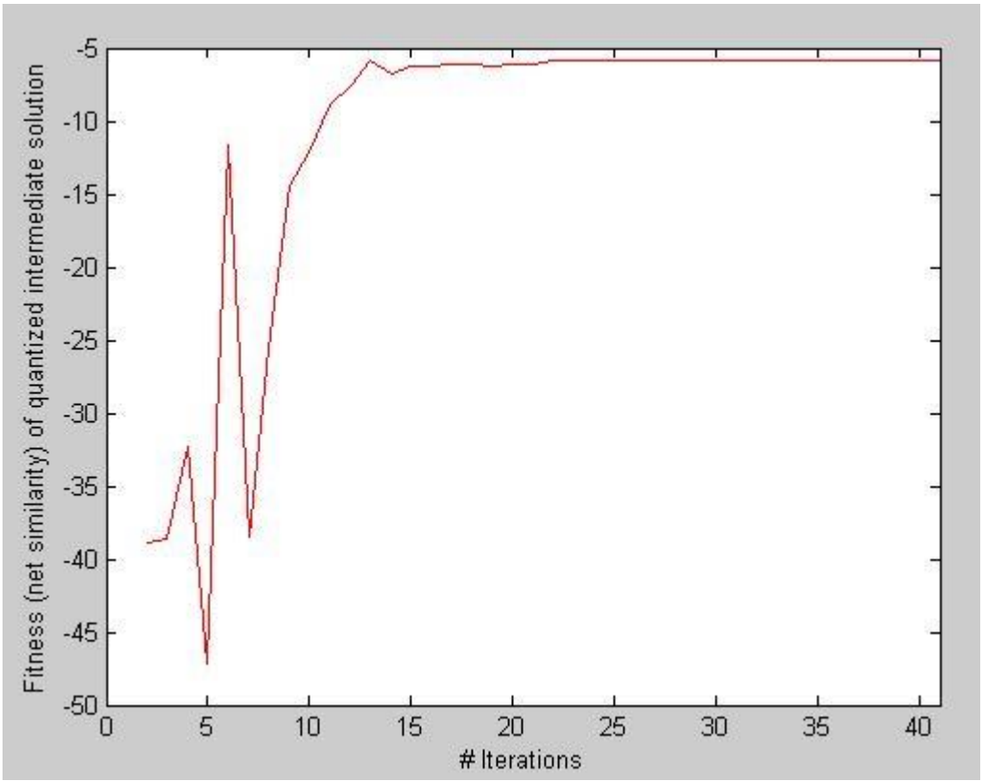

图 7.lam取0.5时的迭代示意图

从上面两个图对比中我们可以发现，当lam值越小时，迭代次数会减少，但是迭代过程中net Similarity值波动会很大，当要聚类的数据点比较大时，这样难于收敛。当lam值较大时，迭代次数会增加，但是总的net Similarity比较平稳。

根据式（5）和式（6），我们也可以看到，每一次迭代的 R_i 和 A_i 受到lam的影响。当lam取较小的值时，R_i 和 A_i 相比上一次迭代的 R_{i-1} 和 A_{i-1} 会发生较大的变化，这也是为什么net Similarity值摆动比较大的原因;当lam取较大值时，R_i 和 A_i 和上一次迭代的 R_{i-1} 和 A_{i-1} 比较接近，这也是导致迭代次数比较多的原因。

正是因为如此，有人提出了**自适应仿射传播聚类**（在文献2中可以看到），文中主要提出了如何**根据数据集自动生成preference值和lam值**的方法。

4. k-means 算法与 AP 算法比较

例 3：下面，我们随机在二维空间中生成 50 个数据点，分别用上面讲述的两种聚类算法进行聚类计算。

我们先进行 AP 算法聚类，将生成的聚类数量用于 k-means 算法中，将结果示意图进行比较，具体结果如下：

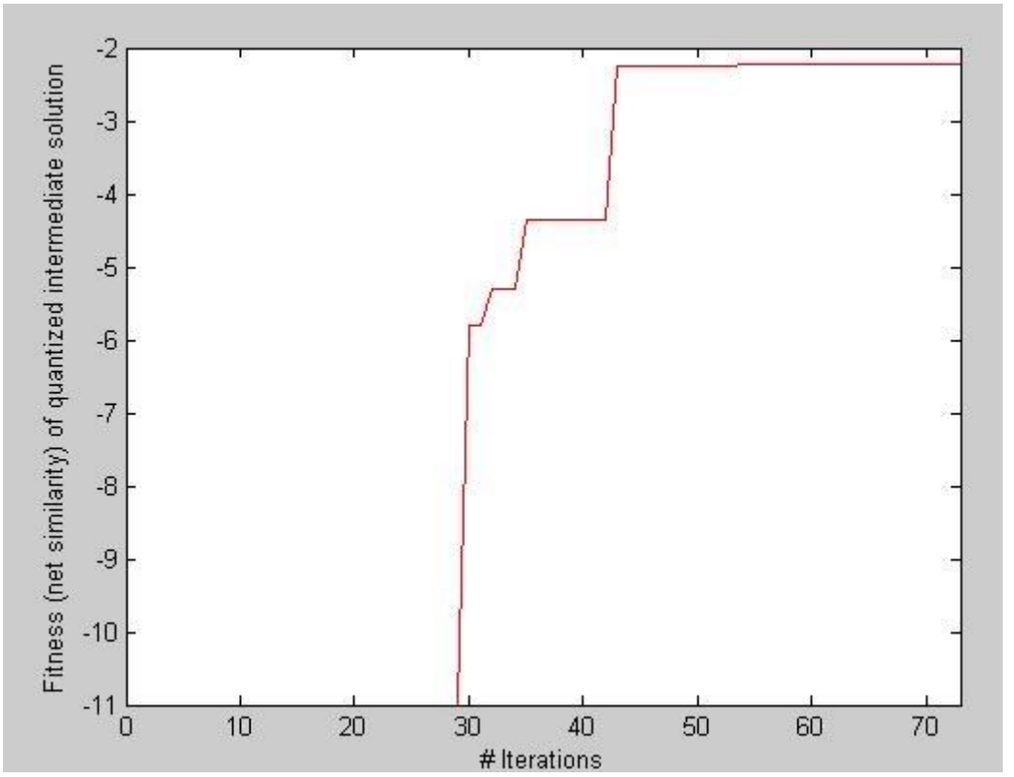

图 8.AP 算法迭代过程

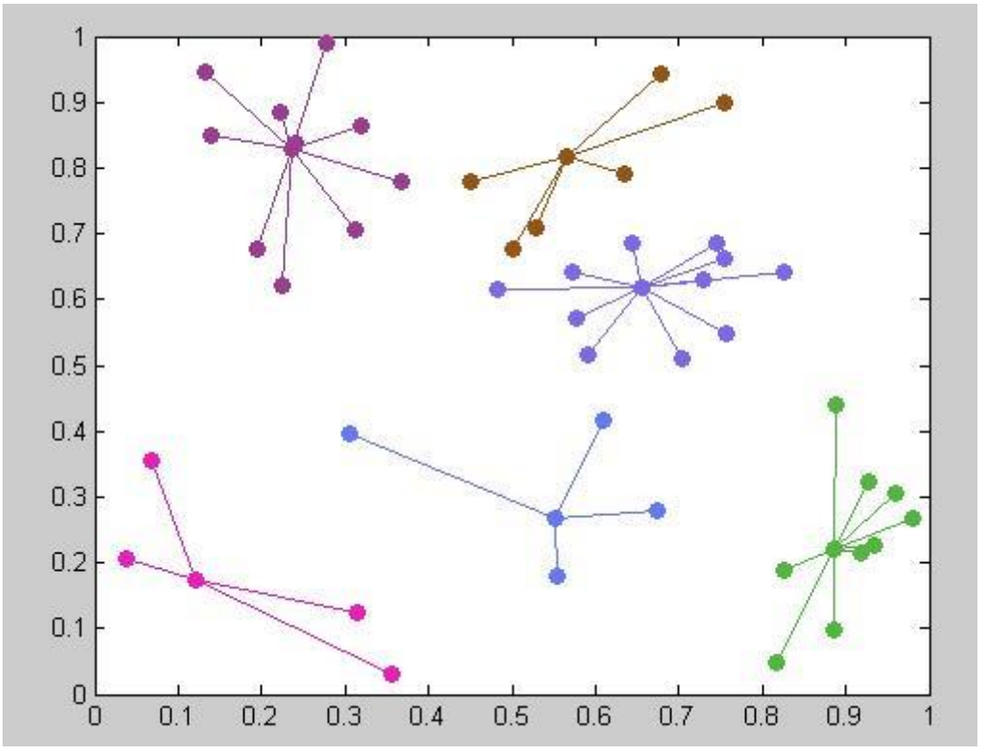

图 9.AP 算法最终计算结果

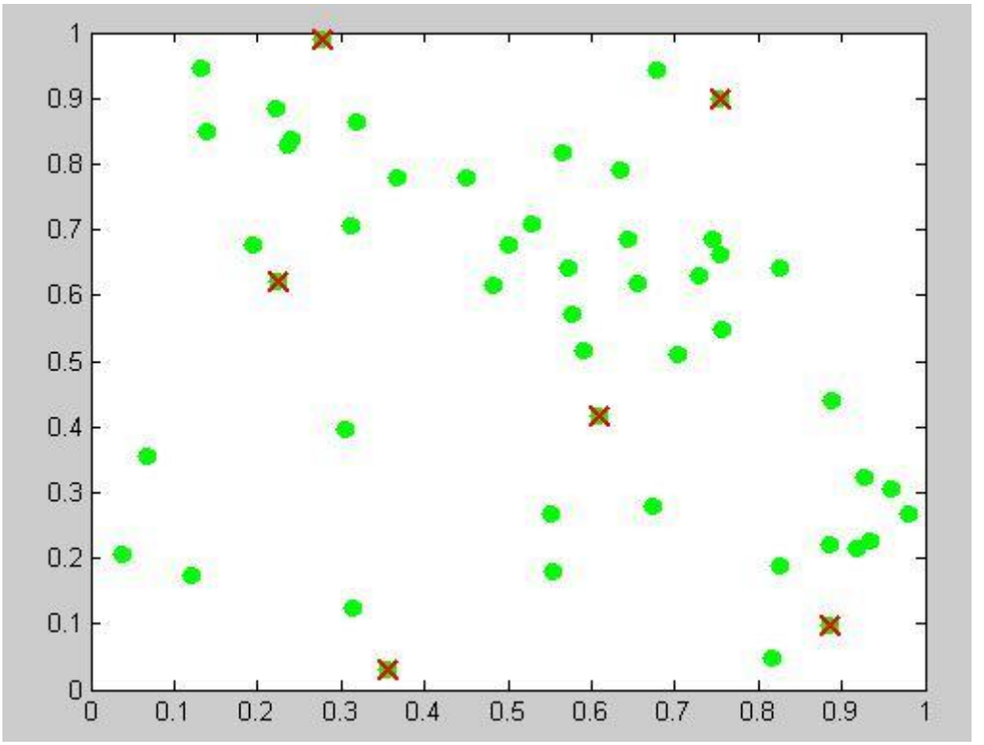

图 10. k-means 算法初始聚类中心示意图

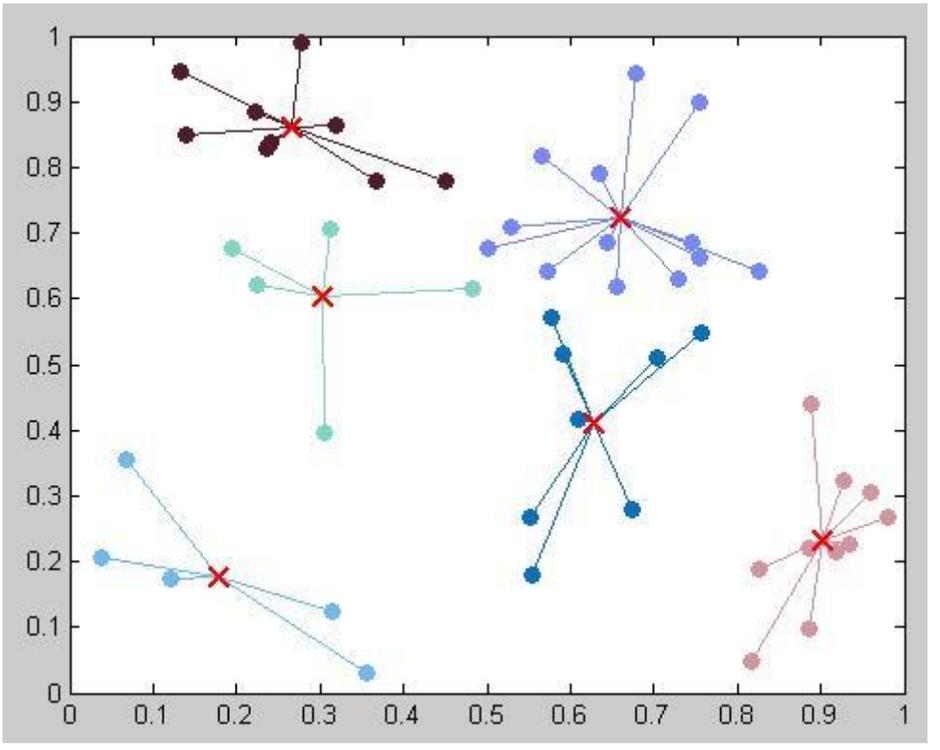

图 11.k-means 算法最终聚类结果

5.总结与展望

k-means 算法对于离散和噪声数据比较敏感，对于初始聚类中心的选择很关

键，因为初始聚类中心选择的好坏直接影响到聚类结果，而且这个算法要求进行聚类时输入聚类数目，这也可以说是对聚类算法的一种限制。不过，这种算法**运行速度相对于 AP 算法要快一些**，因此，对于那些**小**而且数据比较密集的数据集来说，这种聚类算法还是比较好的。

AP 算法对于 **P 值**的选取比较关键，这个值的大小，直接影响都最后的聚类数量。值越大，生成的聚类数越多，反之如此。而且，那个**阻尼系数(lam)**迭代也是很关键的。在文献[2]中有人提及，此算法可能会出现数据震荡现象，即迭代过程中产生的聚类数不断发生变化不能收敛。增大 lam 可消除震荡现象。但根据式[5]和式[6]来看，一味的增大 lam 会使 R 和 A 的更新变的缓慢，增加了计算时间。因此，如何选取一个合适 lam 的来进行计算也成了一个提升算法运行速度的重要因素。

将 AP 算法运用于**图像检索系统**中来进行**初始分类**，我觉得挺有用的。这个想法还有待实现。

6. 参考文献

[1] Frey B J, Dueck D. Clustering by passing messages between data points. Science, 2007, 315(5814): 972~976.

[2] 王开军 张军英等 自适应仿射传播聚类 自动化学报 2007年12月 第33 卷第12 期

[3] Jiawei Han Micheline Kamber 范明 孟小峰 译 数据挖掘概念与技术 机械工业出版社 2008年 251~265.