Affinity Propagation 聚类简介

尹卓
北京工业大学
MichaelYin777@outlook.com

2018 年 5 月 25 日

目录

1 引言

对数据进行聚类,识别出数据的子结构对于数据处理和模式发现是一项重要的工作。*Affinity propagation* 聚类，中文又称为近邻传播聚类，是一种最近被提出来的优秀的聚类算法。它通过迭代更新的方式，不需要预先指定聚类的个数，发现数据中的子结构。迭代的过程当中，信息在样本点之间进行传播。并且实验证明，这种聚类算法的误差，要远小于诸如 *K-centers* 之类的算法。这里，我们先简要介绍一下算法中定义的 3 种矩阵，随后介绍一下算法的迭代与决策过程，最后我们给出一个改造的 *Sklearn* 库里的例子展示各个参数对聚类效果的影响。

2 Affinity Propagation

基于相似性度量的聚类是数据分析的关键一步。实现这种聚类的一般性方法，就是让算法从数据中学习出一系列的“中心”点，使得样本点到这些中心点的距离平方和最小。当这些中心点是实际的样本点的时候，这些中心点被称为*代表点 (Examplars)*。原始的 *K-centers* 算法对于这样的过程有很强的局限性，2007 年，有人在 Science 提出了一种的新的聚类算法，叫做*近邻传播聚类 (Affinity propagation)*，并用实验证明了其优秀的效果 [1]。

2.1 相似度矩阵

AP 算法的思想基于的是近邻信息传播 [2]。对于平面上的样本点，我们首先对它们构成所谓的*相似性矩阵 (Similarity Matrix)*，我们记这个矩阵为 S 矩阵，其中 $s(i,k)$ 代表第 i 行的第 k 个元素。其中 S 矩阵的构造使用的是负欧式距离度量：

$$s(i,k) = -||x_i - x_k||^2 \tag{1}$$

由式子可得，$s(i,k) \in (-\infty, 0]$，代表了样本 i 与 k 之间的相似度，其值越大，越接近 0，说明样本之间越相似。其中，S 矩阵的主对角线上的元素 $s(k,k)$ 被叫做*偏向参数 (Perference)*，代表着第 k 个样本点做聚类中心的合适程度，这个值越大，说明样本点 k 越适合作为类的代表。传统的 AP 聚类算法将这个值设置为一样的值，即假设所有点成为类代表的可能性相同，这个值的大小将会对聚类个数产生非常大的影响，一般来说，这个值被设置为输入的 S 矩阵的所有元素的中值（将会产生中等数目的聚类）或者是最小值（产生比较少的聚类数目）[1]。

2.2 Responsibility Matrix

AP 聚类算法有两种样本点与样本点之间信息的传播方式，每种方式都相当于在“竞争”。第一种方式被叫做*吸引度 (Responsibility)*，代表支持某个点成为代表点的累积证据；另外一种被叫做*归属度 (Availability)*，代表选择某个点当做代表点的合适程度，这两个矩阵分别用 R 矩阵和 A 矩阵来代表，AP 算法的核心思想就是迭代更新这两个矩阵。

吸引度 *(Responsibility)* 矩阵的元素 $r(i,k)$ 代表样本点 k 作为 i 的代表的累积证据。当我们在计算 $r(i,k)$ 时，我们不仅仅要考虑 x_k 做 x_i 代表的合适程度，也要考虑样本点 x_i 选择其他代表的合适程度，二者相互竞争，于是定义吸引度矩阵如下：

$$r(i,k) = s(i,k) - max\{a(i,k^{'}) + s(i,k^{'})\}(k^{'} \neq k) \tag{2}$$

初始迭代时，A 矩阵被设置为零矩阵，那么首次迭代时，我们可以看出，$s(i,k)$ 代表 k 作为 i 的代表的合适程度，$s(i,k^{'})$ 代表 $k^{'}$ 做其他点代表的合适程度，但是这里我们取其中相似度最大者来做相减，这是一个很自然的做法。$r(i,k)$ 越大说明 k 越能代表 i，越小则说明其他样本点更加适合做 i 的代表。在后续的迭代中，$a(i,k^{'})$ 代表 i 选择 $k^{'}$ 作为聚类代表的合适程度，$a(i,k^{'}) + s(i,k^{'})$ 总的来说就是 $k^{'}$ 作为 i 的代表的合适程度，道理上和之前也是一样的。特别说明的是，$r(k,k)$ 代表着 k 样本点做聚类代表的合适程度，若 $s(k,k)$ 初始设定的越高，我们越认为 k 适合做聚类的代表。

2.3 Availability Matrix

归属度 *(Availability)* 矩阵的元素 $a(i,k)$ 代表 i 选择 k 作为其聚类代表的合适程度的累积证据。它的公式如下：

$$a(i,k) = \min\{0, r(k,k) + \sum_{i^{'}\ s.t.\ i^{'} \neq \{k,i\}}^{m} \max\{0, r(i^{'},k)\}\} \tag{3}$$

在计算过程中，不仅要考虑 $r(k,k)$ 自身适不适合做聚类的代表，同时我们也需要考虑样本点 k 做其他样本点的代表的合适性，考虑这一点也是为了说明 k 是不是真的能够代表一个群体，但是这里我们只取其中的正值。这里应当注意到，若是 $r(k,k)$ 取到负值，那么就说明 k 更加适合当做从属，而不是代表，但是同时，其他的样本点对于 $r(k,k)$ 作为代表的和有可能会是正值，若这个值超过了 $r(k,k)$ 的绝对值，说明 i 选择 k 作为代表点是合适的，但是这与 $r(k,k)$ 更适合当从属点刚好相矛盾。为了限制这样的情况，我们定义了归属度的值最大不能超过 0，取了一个极小值的算子。k 对 k 的归属度我们定义为：

$$a(k,k) = \sum_{i^{'}\ s.t.\ i^{'} \neq k}^{m} \max\{0, r(i^{'},k)\} \tag{4}$$

这个信息反映了 k 作为聚类核心的累积证据，利用的是从别的样本点发送的正的归属度。

2.4 决策过程

上面提到的就是 Affinity Propagation 聚类的核心思想，信息借助已知的相似性矩阵，在每对样本点之间传递。在任何一次迭代中，都可以将吸引度矩阵和归属度矩阵结合起来，判断哪些点是代表点。比如对于任意点 i 来说，计算使得 $a(i,k) + r(i,k)$ 最大的点的坐标作为这

个点的类代表。若是 $k = i$，则说明 i 是一个代表点，反之则说明 i 是一个从属点。一般来说，迭代终止的条件，一般设定为固定的迭代次数，或者是样本点之间的传递的信息低于某一个阈值，或是选出来的代表点经过一定的迭代次数不再发生变化。AP 算法在信息更新的过程当中，为了防止迭代发生震荡从而导致无法收敛，还引入了一个阻尼系数 $\lambda(\lambda \in [0.5, 1])$，有时候又叫 *Damping factor*。在每次迭代过程中，$a(i,k)$ 与 $r(i,k)$ 的更新结果都是由当前迭代过程中的更新值与上一次迭代的结果加权获得的:

$$r^{(t)}(i,k) := (1-\lambda)r^{(t)}(i,k) + \lambda r^{(t-1)}(i,k) \tag{5}$$

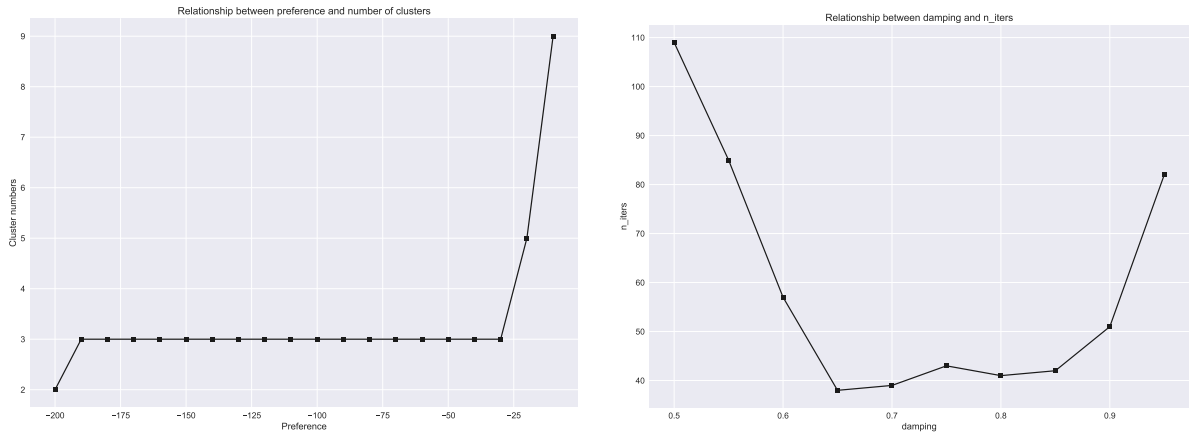
$$r^{(t)}(k,k) := (1-\lambda)r^{(t)}(k,k) + \lambda r^{(t-1)}(k,k) \tag{6}$$

$$a^{(t)}(i,k) := (1-\lambda)a^{(t)}(i,k) + \lambda a^{(t-1)}(i,k) \tag{7}$$

$$a^{(t)}(k,k) := (1-\lambda)a^{(t)}(k,k) + \lambda a^{(t-1)}(k,k) \tag{8}$$

3 例子

我们这里改造了 *Sklearn* 官方库里面的例子，来展示各个参数对于 AP 聚类的影响。



(a) (b)

图 1: (a) 聚类数目和偏向参数大小的关系 (b) 迭代收敛次数与 damping 大小的关系，其中 preference=-50

由 (a) 图可以看到，随着偏向参数越接近 0，我们越认为每个点作为类代表的可能性是一致的，从而产生了越多的聚类。由 (b) 图我们可以看到，随着阻尼系数 λ 变化，迭代的收敛次数先变小后变大。

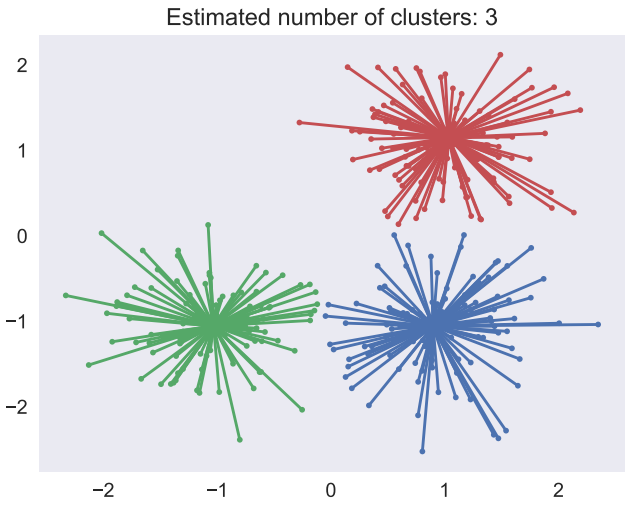


图 2: 聚类结果，preference=-50

最终聚类结果我们可以看到，AP 聚类具有比较优异的效果，并且自动发现了平面上的聚类个数。

4 总结

Affinity Propagation 聚类算法算是一种新颖的聚类算法，它能够自动选择聚类的数目，并且是一种迭代的聚类方式，聚类算法本身速度比较快，并且聚类精度高 [1]。但是 AP 聚类算法需要构建相似性矩阵，构建相似性矩阵的运行时间上会较长；其次虽然说聚类数目由算法自动选择，但是提前需要给定的 preference 参数，一定程度上就代表了选择聚类的数目；同时，AP 聚类算法使用传统欧氏距离作为度量，聚类的效果也容易受到量纲的影响 [2]。

参考文献

[1] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[2] 唐丹. 改进的近邻传播聚类算法及其应用研究. PhD thesis, 南京理工大学, 2017.