# 基本数据清理方法

## Sorted-Neighborhood Method

给定两个或多个数据集合，我们首先将它们合并到一个有N个记录的list中，然后在其上应用Sorted-Neighborhood Method

Sorted-Neighborhood Method解决数据合并、净化大体包括一下步骤：

1. **生成keys**：通过提取相关字段或部分字段，为list中的每个记录生成一个key。Key的选择依赖一个“错误模型”，该模型是知识加强或领域相关。Sorted-Neighborhood Method的效率极大取决于正确选择的key，普遍且大量的数据应该密切的匹配这些key
2. **排序**：利用1中生成的key对记录进行排序
3. **合并**：移动固定大小的窗口顺序通过list中的记录，限制比较匹配记录只在滑动窗口中。如果窗口的大小为w条记录，则进入窗口的一条新记录只是和窗口的前w-1条记录比较去发现匹配记录。窗口的第一条记录滑出窗口。

## Keys的选择

Sorted-Neighborhood Method 的效率依赖Keys的选择，这里keys是属性或属性子串的子集的序列。

考虑：

* 应该有足够的识别差异的能力，相同或相近的记录能排列在相邻位置
* 常见、不易误解、不易记录错误（e.g. 外国名字中，firstname不容易拼错，因为读音大小写等，lastname比较容易拼错，）

## 等式理论（Equational theory）

等式理论：问题域的逻辑相等，不是简单的值相等或字符串相等

轻微差异：distance(rec1, rec2) < threshold

Notes：不需要比较相同属性或相同域的值（可以A属性和B属性比，或A的一部分和B的一部分比）

比较三个阶段：

1. 窗口中的2个记录相比较，判断是否有相似的属性，e.g. 名字、社会保险号码、地址等
2. 将第1阶段收集的信息综合判断是否可以进行合并
3. 对于2中无法合并的记录，采用更精确的edit-distance函数来再次尝试是否能够合并

## 计算独立运行结果的传递闭包

* multi-pass approach

单个key不够获得所有的匹配。Key中前面的属性要比后面的优先级别高，如果前面的属性错误，就无法匹配。

解决方案：

1. 加大窗口的尺寸w
2. 执行几次独立的Sorted-Neighborhood Method过程，每次使用不同的keys和相对小的窗口 ---

* 提高单个pass的精度

如果按照相等规则，a和b相等，b和c相等，则a和c相等，即使a和c不在滑动窗口内。--- 使用较小的滑动窗口，也可以发现匹配的记录

# 增量合并

Delta(i) = 第i次增量数据，

C(i) = 合并过程定义的clusters的主代表，初始值为空

For 每次增量输入Delta(i)

1. I(i) = C(i-1)+ 本次增量Delta(i)，其中C(i-1)是上一次的结果Clusters的主代表
2. 对I(i)执行所有的合并过程，结果是将I划分为新的一群clusters
3. 将I(i)中的每条记录都分配到2中新的clusters中去
4. 如果需要，对本次结果clusters的每个cluster，从其中的记录中选择1或多个记录作为该cluster的主代表。新的C(i) = 本次clusters的所有主代表

End

注意：

1. 主代表是从clusters中选取的能够代表clusters中信息的记录，例如在kmeans中的cluster的中点
2. 算法的精确性，关键在于cluster的主代表的选择，这个与领域知识相关
3. 对于某次结果clusters中的某些cluster，最好的主代表是没有主代表
4. 对于某些预计不会被加入新记录的cluster，可以考虑从结果中去掉，例如按时间划分的clusters，10年前的cluster中应该不会有新的记录加入，这是就可以不选择主代表，把它从结果中删除
5. 本算法的假设：
   1. 已经使用的记录不会被删除
   2. 开始后，合并的规则都不变

选择主代表的可选策略：

1. 随机抽样
2. 最近N个
3. 生成法：通过从能够代表本cluster的正记录中以某种算法生成主记录
4. 最大的或更加复杂记录
5. 选择最频繁匹配其他记录的记录

如果过程中删除一个主代表记录，算法补充如下：

1. 将所有的删除操作延迟到增量算法的step 3（将记录划分到新clusters中）之后。
2. 执行删除操作，并记录所有被影响的clusters的clusterID
3. 重新计算被影响的cluster，可能需要划分已有的clusters