计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(3):172−178 [doi: 10.15888/j.cnki.csa.006835]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于划分的海量数据相似重复记录检测①

李　莉, 张晓雯

(江苏大学 计算机科学与通信工程学院, 镇江 212013)

摘　要: 针对目前社工库存储的海量数据, 数据冗余、查询效率低下的质量问题, 本文提出了一种有效的基于划分的近邻排序算法. 对不同渠道采集、以不同存储方式存储的社工数据进行整合形成能以二维表形式存储的海量数据集, 采用划分思想, 对大数据集进行分割, 形成簇; 采用改进的近邻排序算法对各个簇中的小数据集进行检测得到最终的相似重复记录检测结果. 实验和对比分析结果表明, 划分和近邻排序算法的结合使用不仅提高了海量数据相似重复记录检测的时间效率, 检测准确率也有所提升.

关键词: 数据质量; 数据清洗; 相似重复记录; 划分; SNM 算法

引用格式: 李莉,张晓雯.基于划分的海量数据相似重复记录检测.计算机系统应用,2019,28(3):172–178. http://www.c-s-a.org.cn/1003-3254/6835.html

Similar Duplicate Record Detection of Massive Data Based on Partition

LI Li, ZHANG Xiao-Wen

(School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China)

Abstract: Aiming at solving problems of data redundancy and low query efficiency in the storage of mass social work data, this study proposed an effective partition-based neighbor sorting algorithm. The social data collected by different channels and stored in different storage methods were integrated to form a massive data set that can be stored in a two-dimensional form. The partitioning idea was used to segment the massive data set to clusters; the improved neighbor sorting algorithm was used for each cluster to obtain the final similar duplicate record detection results. The experimental and comparative analysis results show that the combination of partitioning and neighbor sorting algorithm not only improves the time efficiency of similar duplicate records detection of massive data, but also improves the detection accuracy.

Key words: data quality; data cleaning; similar duplicate records; partition; SNM algorithm

1　引言

随着互联网、移动互联网及智能手机的快速发展, 人们每时每刻都产生大量的数据, 如个人登录某网站的账号、密码、邮箱、分享的照片、信用卡记录、订的机票记录、通话记录等个人行为数据. 而现代社会数据成为了决策者, 在社工库的海量数据中挖掘出有价值的信息, 即让数据资源转化为信息, 将对做出合理的应对决策和精确的任务计划, 甚至业务方面的收益等都有着重要的意义. 社工库中的数据采集方式有人工录入、网络爬取等, 囊括了来自不同数据源的数据. 多数据源集成时, 存在着对同一个概念有不同表示方法的问题, 如同一个人可以在两个不同的数据源有两种不同表示[1]. 数据源的数据格式也较复杂, 有.csv 格式、.sql 格式、.excel 格式等. 数据的完整性也存在着一定的欠缺, 如数据的重复、缺失、错误等问题, 而这将直接影响数据挖掘与分析的结果[2]. 因此对相似重复

① 收稿时间: 2018-10-04; 修改时间: 2018-10-23; 采用时间: 2018-11-05; csa 在线出版时间: 2019-02-22

记录的检测便成了数据清洗中的一个关键环节.

近年来, 国内外学者在相似重复记录检测研究中取得不错的成绩, 相似重复记录清洗通过排序、相似检测与合并/删除, 相似检测算法是清洗的核心. 利用编辑距离[3]、N-gram 字符串匹配度量[4]等方法, 进行相似记录比较. 利用优先权队列算法[5]、近邻排序算法[6]、多趟近邻排序算法[7]等检测相似重复记录. 针对海量数据的特点, 杨巧巧等人使用网格法对海量数据进行分组, 并为各属性设立对应的权值, 提高了算法的检测效率以及准确度, 但网格划分的大小是根据经验设定的[8], 而网格划分效果很大程度依赖于网格步长的选取, 并且只对密度分布较为均匀数据进行采样, 未充分考虑不同密度的簇、噪声和密度阈值的关系对划分结果的影响. 针对相似重复记录检测中记录属性维度过高导致的查准率和时间效率低下的问题, 文献利用信息熵, 通过过滤噪声属性, 降低属性维度, 提高了相似重复记录检测算法的效率[9], 但随着待检测记录数量的增多, 算法耗时会迅速增多. 在 CURE 算法的改进方面也取得了成就, 时念云等人使用预抽样改进代表点选择方法, 采用基于距离影响因子的代表点选取策略, 既可以根据数据集的密度进行代表点的选取, 又能适当选取有一定意义的边缘点作为代表点, 提高了代表点选取的合理性[10], 但在预抽样中, 未考虑到相邻样本集可能出现交叉重复记录的情况. 在 SNM 算法的研究上亦取得了较大的进步. 刘雅思等人使用长度过滤法对数据进行预处理, 根据两条记录的长度比例和属性缺失情况, 排除部分不可能构成相似重复记录的数据; 进一步使用动态容错法, 校准字段相似度评判结果, 解决了因属性缺失而误判的问题[11],但对于属性权重的设置存在主观性, 并且未能处理文字不同而语义相似的重复记录. Miao Li 等人使用余弦相似度算法进行属性匹配, 提高匹配精度, 并且采用 Top-k 有效权重过滤算法, 选择权值较高的 Top-k 个属性进行匹配, 最后计算 k 个相似度值和权重的总和, 减少了属性的匹配次数[12]. 陈爽等人使用变步长伸缩窗口, 动态检测窗口大小, 减少不必要的匹配, 并采用动态调整等级法, 根据记录相似度调整字段等级, 通过等级法将字段等级转换为权重, 解决了人为赋予固定权重主观性强、不准确问题[13]. 但文献[12,13]均未对海量数据进行预处理, 而直接采用相关改进算法检测, 未能解决海量数据相似重复检测时间效率低下的问题.

针对 SNM 算法的缺点以及海量数据的特点, 本文提出了一种有效的基于划分的近邻排序算法. 算法主要步骤为: 首先根据属性对海量大数据集进行划分, 形成小数据集; 然后对划分后的小数据集, 采用改进的近邻排序算法进行清洗. 提高了在海量数据库中查找相似重复记录的时间效率以及检测精度.

2 近邻排序算法 (SNM)

相似重复记录检测中比较有效的方法是近邻排序算法. 近邻排序 (SNM) 算法的基本思路是:

1) 创建排序关键字. 抽取记录的一个或一组属性字符串, 计算数据集中每一条记录的键值.

2) 排序. 根据排序关键字对整个数据集进行排序. 潜在的相似重复记录将被调整到一个临近的区域, 从而可以将匹配限定到一定的范围之内[11].

3) 合并. 在排序后的数据集上滑动一个大小为 Q 的窗口, 窗口内的第一条数据仅与窗口内剩余的 Q-1 条记录进行比较. 比较结束后, 最先进入窗口内的记录滑出窗口, 最后一条记录的下一条记录移入窗口, 再把此 Q 条记录作为下一轮比较对象. 如此反复, 直到所有记录比较完毕[11,14].

SNM 算法通过滑动窗口减少了比较次数, 提高了匹配效率, 算法的时间复杂度为 O(QN)[12](Q 为窗口大小, N 为数据中的记录总数). 但是 SNM 存在两个主要的缺陷, 一是排序关键字难以选取, 排序关键字选取的好坏不仅直接影响检测效率, 对测重结果的精度也有很大影响. 二是滑动窗口大小难以设置, 如果窗口过大会导致不必要的记录比较, 如果窗口过小, 会出现漏配现象, 降低检测精度.

3 PSNM 算法思想

针对传统近邻排序 (SNM) 算法的缺点, PSNM 算法的思想是, 首先对海量数据集进行划分, 形成小数据集; 其次对每个小数据集采用等级综合评价法为属性设置权重, 以权重排序关键字; 最后采用可伸缩的滑动窗口, 进行相似重复记录检测.

1) 数据划分

把海量大数据集划分为若干个不相交的小数据集. 划分方法如下:

步骤 1. 选取具有代表性的某个属性, 以该属性把大数据集分割为若干个不相交的小数据集, 称为簇. 记

大数据集 D, 属性 $P=\{p_1, p_2,\cdots, p_m\}$, 划分为 n 个不相交的子集, D: $\{D_1, D_2,\cdots, D_n\}$.

步骤 2. 若某些划分后的数据集的数据量仍比较大, 则对该数据集再次划分. 对数据集 $D_i(i=1, 2,\cdots, n)$ 再次划分, 选取属性 $P_i=\{p_{i1}, p_{i2},\cdots, p_{ik}\}$,依据属性划分为 k 个不相交的子集, D_i: $\{D_{i1}, D_{i2},\cdots, D_{ik}\}$.

步骤 3. 若仍存在某些数据集比较大, 可重复步骤 2, 直到数据集划分较为合理为止.

数据划分依赖于问题的求解, 为了求解的精确度, 可将只选取单一属性进行数据划分, 扩展到多属性选取进行数据划分.

2) 关键字选取

为选取恰当的关键属性, 本文采用等级综合评价. 等级综合评价法结合了客观的数理统计方法和主观的专家经验, 综合考虑了各个属性对记录的贡献程度不同[2].

数理统计方法: 每个属性都有值域. 多次在数据集中随机选取相同大小的样本数据, 统计属性的长度, 称为属性值. 为减少随机取样时样本质量差异对属性值的影响, 因此以均值法确定数据集中各属性值, 若属性值越大, 此属性的记录差异越大, 该属性所占权重也就越大. 属性值统计如表 1 所示.

表 1 属性值统计表

随机取样次数	属性 p_1	属性 p_2	属性 p_3	…	属性 p_m
1	Y_{11}	Y_{12}	Y_{13}	…	Y_{1m}
2	Y_{21}	Y_{22}	Y_{23}	…	Y_{2m}
…	…	…	…	…	…
n	Y_{n1}	Y_{n2}	Y_{n3}	…	Y_{nm}
属性值	Y_1	Y_2	Y_3	…	Y_m
权重	W_1	W_2	W_3	…	W_m

采用均值法确定属性 p_j 的属性值 Y_j:

$$Y_j = \frac{1}{n}\cdot\sum_{i=1}^{n} Y_{ij} \tag{1}$$

其中, $Y_{ij}(1\le i\le n, 1\le j\le m)$ 表示第 i 次取样第 j 个属性 p_j 的属性长度.

将属性值从小到大进行排序, 并分别赋予属性权重 $W_i(1\leqslant i\le m)$, 其中 $W_1+W_2+\cdots+W_m=1$.

经验值设定: 结合领域知识让用户根据个人经验为各属性进行等级分配, 即为每个属性指定经验等级. 为降低各专家经验认知不同对属性打分的影响, 仍采用均值法就算各个属性最终的经验等级 $G_j(1\le j\le m)$. 经验等级表如表 2 所示.

表 2 属性经验等级表

用户	属性 p_1	属性 p_2	属性 p_3	…	属性 p_m
1	G_{11}	G_{12}	G_{13}	…	G_{1m}
2	G_{21}	G_{22}	G_{23}	…	G_{2m}
…	…	…	…	…	…
n	G_{n1}	G_{n2}	G_{n3}	…	G_{nm}
经验等级	G_1	G_2	G_3	…	G_m
权重	W_1	W_2	W_3	…	W_m

采用均值法确定属性 p_j 的属性值 G_j:

$$G_j = \frac{1}{n}\cdot\sum_{i=1}^{n} G_{ij} \tag{2}$$

其中, $G_{ij}(1\le i\le n, 1\le j\le m)$ 表示第 i 个用户为第 j 个属性 p_j 的分配的经验等级.

将属性值从小到大进行排序, 并分别赋予属性权重 $W_i(1\le i\le m)$, 其中 $W_1+W_2+\cdots+W_m=1$.

根据上述分析, 属性的等级越高, 其重要性越高, 所对应的属性的权重也就越大, 将数理统计方法与经验等级法相结合, 计算得到最终的综合属性权值, 公式如下[2]:

$$W_j = \frac{1}{2}\left(\frac{Y_j}{\sum_{j=1}^{m} Y_j} + \frac{G_j}{\sum_{j=1}^{m} G_j}\right) \tag{3}$$

其中, $\sum_{j=1}^{m} W_j = 1$, 而 m 表示记录中有 m 个属性.

等级综合评价法伪代码如下:

Input: 用户数 n, 用户经验 G, 选取次数 time, 随机选取记录数 count

Output: $(W_1, W_2,\cdots, W_m)$(m 表示记录的 m 个属性)

```
computeWeight(){
Y=statisticsWeight(time, count);//统计
G=gradeWeight(n, E);//经验
for(i=1…m){//计算最终的权重
  Wi=1/2*(Yi+Gi);
  }
  return W;
}
```

3) 排序

近邻排序算法很大程度上依赖于排序关键字的选取. 依据上述等级综合评价法, 首先操作数据集以属性对应的权重 W 从大到小进行排序, 选出前四个属性作

为数据集的排序关键属性. 如对特定的社工数据集, 划分为小数据集后, 经上述方法最终选取“Firstname”、“Lastname”、“家庭住址”、“所在城市”四个属性作为排序关键属性, 对各小数据集进行排序.

4) 可伸缩的滑动窗口

传统近邻排序算法滑动窗口大小难以设置, 窗口过大或者过小都会出现一系列的问题, 从而影响最终的检测效果, 因此滑动窗口大小的设置也极为重要. 本文根据窗口内记录间的相似程度动态地调整滑动窗口大小. 记窗口最大值为 $Q_{\max}$, 窗口最小值为 $Q_{\min}$, 当前滑动窗口大小在 $Q_{\min}$ 和 $Q_{\max}$ 之间变化, 滑动窗口大小根据记录相似度的计算值与阈值的比较进行灵活调整. 可伸缩滑动窗口需设置 3 个参数, 窗口最小值 $Q_{\min}$, 窗口最大值 $Q_{\max}$, 窗口最小阈值 $LowT$, 以及变量当前滑动窗口大小 Q_i. 窗口初始值 Q_i 设定为 $Q_{\min}$, 窗口中的第 1 条记录 R_1 首先在 $Q_{\min}$ 范围内与其他记录进行匹配, 当匹配到记录 $R_{Q\min}$ 时, 如果相似度 $Sim(R_1, R_{Q\min})>LowT$, 说明此时窗口内记录间相似程度较高, 应扩大窗口继续进行匹配, 窗口扩大为:

$$Q_i = Q_i + Q_{\min} \times \frac{Sim(R_1, R_{Q\min}) - LowT}{1 - LowT} \tag{4}$$

如果相似度 $Sim(R_1, R_{Q\min})<LowT$, 记录对的相似度较低, 应缩小窗口, 减少不必要的比较. 此时窗口 Q_i 缩小为:

$$Q_i = Q_i - Q_{\min} \times \frac{LowT - Sim(R_1, R_{Q\min})}{LowT} \tag{5}$$

经过多次实验, 最终确定窗口最小阈值 $LowT$ 以及相似度最小阈值为 0.75 时效果最佳, 若两条记录已经匹配过的属性相似度之和大于等于相似度最小阈值 0.75, 即可确定为相似重复记录, 对后续属性可不进行匹配. 窗口最小阈值以及相似度最小阈值是针对本数据集进行多次实验比较后得出的, 并不具有普遍性, 若对其他数据集采用该算法, 还需根据查准率、召回率等对阈值进行调整.

PSNM 算法根据属性对海量数据集进行划分, 大大降低了数据量等级, 提高后续的测重效率; 采用等级综合评价法为各属性设置权重, 使关键属性的选取以及各属性权重的分配更为客观合理, 提高了算法检测重复记录的准确性; 滑动窗口大小的伸缩, 使窗口大小随窗口内记录间的相似程度动态调整, 在增强算法测重能力的同时减少了不必要的匹配, 提高算法运行效率.

4 实验结果与分析

4.1 实验环境

实验计算机配置: CPU Core(TM)3.40 GHz, 内存 16 GB, 显存 8 GB; 操作系统: Windows7; 软件环境: Python2.7, MySQL5.7.

4.2 评价标准

衡量相似重复记录检测算法的性能指标通常用查准率 (*precision*)、召回率 (*recall*) 和平均数 *F* 值来加以描述. 查准率是指正确识别的相似重复记录与实际的相似重复记录的比值, 查准率越高, 表明该算法识别相似重复记录精度越高. 召回率是指相似重复记录被正确识别的百分率, 召回率越高, 表明该算法识别相似重复记录的能力越强. 查准率和召回率定义如下:

$$precision = \frac{tp}{tp + fp} \times 100\% \tag{6}$$

$$recall = \frac{tp}{tp + fn} \times 100\% \tag{7}$$

其中, *tp* 表示正确识别的相似重复记录数, *fp* 表示错误识别的相似重复记录数, *fn* 表示未识别的相似重复记录数[15].

由于 *precision* 和 *recall* 有时会出现矛盾的情况, 因此采用求它们的调和平均数 *F* 值的方法, 来综合考虑算法的性能, *F* 值越高表明该算法的综合性能越高. *F* 值的定义如下:

$$F = \frac{1}{\frac{1}{2}\left(\frac{1}{precision} + \frac{1}{recall}\right)} = \frac{2 \times precision \times recall}{precision + recall} \tag{8}$$

4.3 实验结果与分析

本实验数据为非洲地区人口社工数据, 其中包含: 姓名, 性别, 家庭住址, 所在城市, 所在州编号, 电话号码, 邮箱, 密码, 受教育等级, 工作等级等 24 个属性. 分析数据源, 不同地区同名同姓同地址的两条记录有可能是相似重复记录, 而相同地区同名同姓同地址的两条记录有极大的可能是相似重复记录. 因此选择“所在州编号”属性对大数据集进行划分有重大意义. 将大数据集 *D* 按照属性“所在州编号”分割形成相应的簇, 划分成了 $\{D_1, D_2, \cdots, D_{69}\}$ 69 个小数据集. 再对 $\{D_1, D_2, \cdots, D_{69}\}$ 各个小数据集进行相似重复记录的检测.

选取 2.5 万条数据按照其“所在州编号”属性进行划分, 结果如图 1 所示.

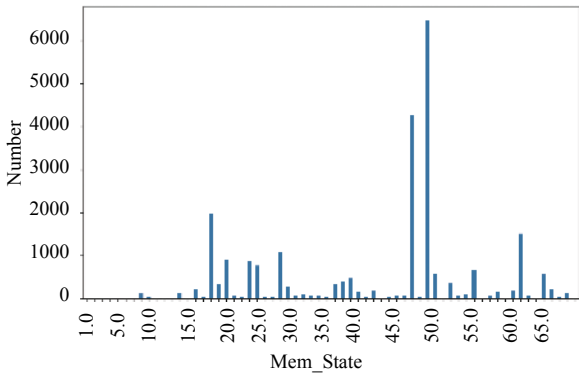

图 1 数据划分结果图

观察图 1 可知, 通过划分将大数据集划分成了各个有意义的簇, 而在各个簇中, 仅有一个簇中的数据量是大于五千的. 这一操作将万级数据量瞬间转化为千级, 并且对于簇中仅有一条数据的簇, 不需要对其进行检测, 只需要检测簇中数据量大于等于 2 的簇. 这大大提高了后续算法检测的检测效率.

分别选取其中 5000 条、1 万条、1.5 万条、2 万条、2.5 万条数据作为测试集, 首先根据“所在州编号”属性对其进行划分, 选取“Firstname”、“Lastname”、“家庭住址”、“所在城市”四个属性作为排序关键字, 对小数据集进行排序, 这四个属性的权重经过计算分别为: 0.0552, 0.0448, 0.0366, 0.0301, 其他属性权重这里就不加以赘述了. 为本文提出的基于划分的近邻排序算法 (PSNM 算法) 设置滑动窗口最小值为 50, 最大值为 70, 窗口最小阈值为 0.75, 相似度最小阈值为 0.75. 而传统 SNM 算法也选取“Firstname”、“Lastname”、“家庭住址”、“所在城市”四个属性作为排序关键字, 对记录进行排序, 设置固定窗口大小为 50, 相似度最小阈值为 0.75. 各算法在选取不同数据量时的运行时间对比如表 3 所示.

表 3 各算法运行时间对比 (单位: s)

记录数 (条)	SNM	Cure	PSNM
5000	13.498	13.242	12.563
10 000	41.953	45.736	21.466
15 000	59.473	64.587	42.739
20 000	103.529	125.181	63.532
25 000	153.472	172.931	105.476

由表 3 可以看出, 在排序关键字选取相同的情况下, PSNM 算法的运行效率是高于其他算法的, 这是由于 PSNM 算法首先对数据进行了划分, 大大降低了数据量等级, 并且滑动窗口的使用和属性权值的设置, 也能减少记录间不必要的匹配, 节省了相似重复记录的检测时间, 提高了算法效率.

接着分两组进行实验, 分别为实验 1 和实验 2. 实验 1 中, 设置滑动窗口最小值 30, 最大值 50; 实验 2 中, 设置滑动窗口最小值 50, 最大值 70. 使用上述方案, 在相同的实验环境下, 分别利用 SNM 算法、Cure 算法和 PSNM 算法进行实验, 并对同一个数据集进行测试. 测试结果均与 Python Pandas 库测重结果进行对标, 实验 1 结果如图 2 至图 4 所示.

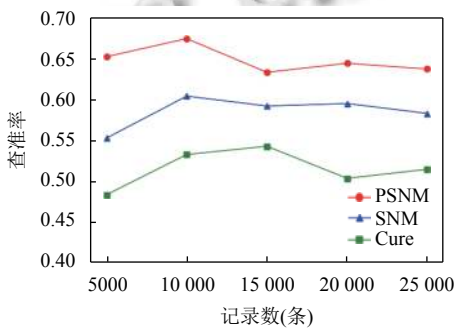

图 2 实验 1 中 SNM 与 PSNM 算法的查准率对比

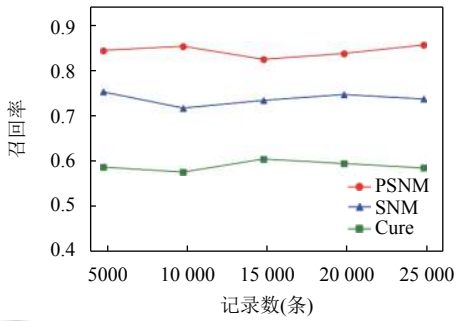

图 3 实验 1 中 SNM 与 PSNM 算法的召回率对比

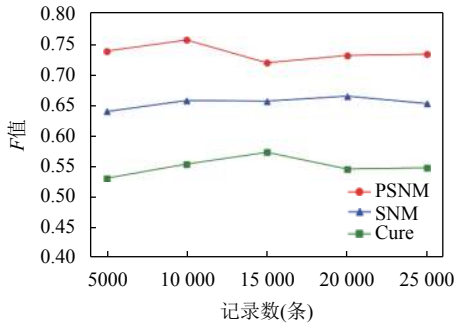

图 4 实验 1 中 SNM 与 PSNM 算法的 F 值比对

从图 2 可以看出, PSNM 算法的查准率始终高于 SNM 算法和 Cure 算法. 但是随着数据量的增大, 两个

算法的查准率之间的差距逐渐缩小, 这主要是因为随着数据量的增加, 滑动窗口最大值 50 已与记录不适配, 因此 PSNM 算法的查准率与 SNM 算法趋近.

从图 3 可以看出, PSNM 算法整体提高了相似重复记录的召回率, 提高了相似重复记录检测能力, 解决了因字段权重分配不合理以及窗口大小不合适导致的部分相似重复记录无法识别的问题.

从图 4 可以看出, PSNM 算法的综合性能指标 *F* 也优于 SNM 算法和 Cure 算法, 说明 PSNM 算法整体性能得到了提升.

实验 2 的结果如图 5、图 6、图 7 所示.

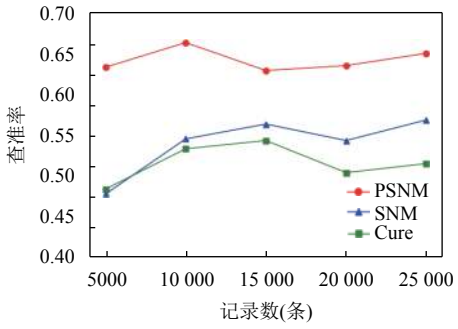

图 5 实验 2 中 SNM 与 PSNM 算法的查准率比对

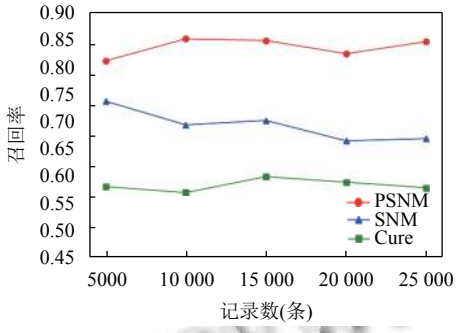

图 6 实验 2 中 SNM 与 PSNM 算法的召回率对比

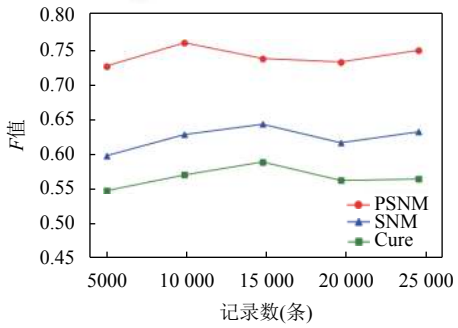

图 7 实验 2 中 SNM 与 PSNM 算法的 *F* 值对比

从图 5 可以看出, 窗口初始大小设置为 50 后, PSNM 算法的查准率仍优于 SNM 算法. 由实际数据可知, 窗口大小为 50 时, 很多记录之间的比较是没有必要的, 这时 PSNM 算法能动态调整窗口大小, 减少将非相似重复记录误认为相似重复记录的情况, 提高了算法的检测精度.

从图 6 可以看出, 此时窗口初始值 50 比较大, 基本能囊括所有的相似重复记录, 在所有数据集上, PSNM 算法召回率均高于 SNM 算法. 随着数据量的逐渐增大, PSNM 算法召回率波动并不大且基本维持在 0.75~0.85 之间, 而 SNM 算法的召回率却存在极大的波动且召回效果并不佳.

从图 7 可以看出, 当数据量增加时, PSNM 算法的 F 值始终是高于 SNM 算法和 Cure 算法的, 这说明了 PSNM 算法提高了整体性能.

5 结论

为提高被挖掘数据源的数据质量, 消除数据源中的相似重复记录是数据清洗研究中的一个热门课题. 本文在分析了传统 SNM 算法的基础上, 提出了三点改进: (1) 对大数据集进行划分; (2) 综合等级法选取排序关键字; (3) 滑动窗口大小可伸缩. 由于社工库数据量庞大, 因此先运用划分的思想, 将大数据集分割为小数据集, 在各个小数据集中进行相似重复记录检测. 最后通过实验验证, PSNM 算法不仅在时间效率方面有所提升, 并且在查准率、召回率、综合性能都优于原算法以及其他算法.

虽然数据划分提高了海量数据重复记录检测的时间效率, 但在划分过程中还会出现把相似记录划分到不同小数据集的情况, 从而造成相似重复记录的漏判, 因此数据划分技术还有待提高. 并且受到实验环境的制约, 本文仅处理了以二维表形式存储结构化的社工源数据, 并且在形成的海量数据集中仅选取部分数据进行检测. 下一步工作, 要继续处理非结构化数据以及将数据量继续扩大.

参考文献

1 Dhivyabharathi GV, Kumaresan S. A survey on duplicate record detection in real world data. Proceedings of the 3rd International Conference on Advanced Computing and Communication Systems. Coimbatore, India. 2016. 1–5.

2 杨巧巧, 郭振波, 王开西. 基于网格分组和属性权值的相似重复记录识别算法. 青岛大学学报 (自然科学版), 2017, 30(2): 69–73.
3 刘许刚, 黄海, 马宏. 一种基于分段匹配的字符串匹配算法. 计算机应用与软件, 2012, 29(3): 128–131. [doi: 10.3969/j.issn.1000-386X.2012.03.035]
4 Beskales G, Ilyas IF, Golab L, *et al*. On the relative trust between inconsistent data and inaccurate constraints. Proceedings of the IEEE 29th International Conference on Data Engineering. Brisbane, QLD, Australia. 2013. 541–552.
5 Monge AE, Elkan CP. An efficient domain-independent algorithm for detecting approximately duplicate database records. Proceedings of Workshop on Research Issues on Data Mining and Knowledge Discovery. Tucson, AZ, USA. 1997. 23–29.
6 Hernández MA, Stolfo SJ. Real-world data is dirty: Data cleansing and the merge/purge problem. Data Mining and Knowledge Discovery, 1998, 2(1): 9–37. [doi: 10.1023/A:1009761603038]
7 Hernández MA, Stolfo SJ. The merge/purge problem for large databases. Proceedings of 1995 ACM SIGMOD International Conference on Management of Data. San Jose, CA, USA. 1995. 127–138.
8 杨巧巧, 郭振波, 王开西. 基于聚类分组和属性综合权值的 SNM 改进算法. 工业控制计算机, 2017, 30(9): 27–28, 31. [doi: 10.3969/j.issn.1001-182X.2017.09.012]
9 张平. 海量数据相似重复记录检测的研究[硕士学位论文]. 桂林: 桂林电子科技大学, 2011.
10 时念云, 张金明, 褚希. 基于 CURE 算法的相似重复记录检测. 计算机工程, 2009, 35(5): 56–58. [doi: 10.3969/j.issn.1007-130X.2009.05.016]
11 刘雅思, 程力, 李晓. 基于长度过滤和动态容错的 SNM 改进算法. 计算机应用研究, 2017, 34(1): 147–150, 155. [doi: 10.3969/j.issn.1001-3695.2017.01.031]
12 Li M, Xie Q, Ding QL. An improved data cleaning algorithm based on SNM. Huang ZQ, Sun XM, Luo JZ, *et al*. Cloud Computing and Security. Cham: Springer, 2015. 259–269.
13 陈爽, 刁兴春, 宋金玉, 等. 基于伸缩窗口和等级调整的 SNM 改进方法. 计算机应用研究, 2013, 30(9): 2736–2739. [doi: 10.3969/j.issn.1001-3695.2013.09.044]
14 Low WL, Lee ML, Ling TW. A knowledge-based approach for duplicate elimination in data cleaning. Information Systems, 2001, 26(8): 585–606. [doi: 10.1016/S0306-4379(01)00041-2]
15 周典瑞. 基于可变滑动窗口的相似重复记录检测算法研究与设计[硕士学位论文]. 镇江: 江苏大学, 2013.