文章编号：2096-1472(2018)-09-37-03
DOI:10.19644/j.cnki.issn2096-1472.2018.09.011

基于主动学习的数据清洗系统

郭开彦，王洪亚，程炜东
(东华大学计算机科学与技术学院，上海 201620)

摘　要：ADC(Active learning based data cleaning system)运用主动学习的方法，在高效的清洗过程中，部分利用用户交互，提升模型清洗能力，提高数据质量。ADC包含学习模块和选择模块。在学习模块中，模块维护一个概率分类器，计算确定度(模型对修复结果的确定程度)，利用确定度为数据修复做决策。在选择模块中，模块运行数据选择算法，选择最不确定、最有利于数据质量提升的数据交给用户清洗，再选择高分类贡献度的干净数据补充到训练集中，逐步提升模型的修复能力。系统演示表明，ADC系统只需要很少的用户参与，就可以极大地提高数据质量，从而提升了数据清洗的效率。

关键词：数据清洗；主动学习；确定度
中图分类号：TP319　**文献标识码**：A

The Active Learning Based Data Cleaning System

GUO Kaiyan,WANG Hongya,CHENG Weidong
(*School of Computer Science and Technology*,*Donghua University*,*Shanghai* 201620,*China*)

Abstract: In order to enhance cleaning capability of the model and improve data quality,ADC (Active-learning-based Data Cleaning system) uses active learning methods to partially utilize user interactions during the process of efficient cleaning.ADC contains two modules:the learning module and the selection module.The learning module maintains a probability classifier,calculates certainty (how the model determines the repair result),and uses certainty to make decisions for data repair.The selection module runs a data selection algorithm,which selects the data that is most uncertain and most conducive to the improvement of data quality,and then sends the results to the user for cleaning.Following this the selection module selects the clean data with high classification contribution to supplement the training set,and then gradually enhances the repair ability of the model.

Keywords:data cleaning;active learning;certainty

1 引言(Introduction)

检测和修复脏数据是数据分析中的挑战之一，低质量的数据将导致分析不准确和决策不可靠。数据质量专家估计，错误的数据可能会使企业花费其系统实施预算总额的10%至20%[1]。他们一致认为，纠正数据错误是一个耗时、耗力且十分乏味的过程，而一个项目预算的40%至50%可能都用在数据修复中。更多的数据来源和更多的数据量意味着数据质量问题的多样性和复杂性更大，以及以成本效益的方式来保持数据质量的复杂性更高。因此，各种数据清理方法相继被提出，以便自动或半自动地识别错误，并在可能的情况下纠正它们。

在过去几年里，出现了大量基于完整性约束[2−5]、统计[6]或机器学习[7]的数据清理方法。尽管它们具有适用性和通用性，但它们无法确保修复数据的正确性。为了提高这些方法的准确性，常用的方法有引入表格主数据和领域专家[8−11]等。然而这些方法需要的资源是稀缺的，通常也很昂贵。为了解决这些问题，结合知识库和众包的方法被提出[12]，而知识库的构建、存储、维护，以及众包的使用仍需要一定的成本。本系统在机器学习的数据清洗方法基础上，引入领域专家，将机器判定不确定的数据交与人清洗，在高效的清洗过程中，仅需要很少的人工资源就能使数据质量进一步的提升，且修复数据的正确性有一定的保证。

本文设计并实现了ADC系统(Active learning based Data Cleaning system)，运用主动学习的方法，在高效的清洗过程中，部分利用用户交互，提升模型清洗能力，提高数据质量。本系统包含了几个特性：(1)使用BvSB准则[13−15]，赋予数据确定度，用以表示学习模型对其预测的修复结果的确定程度。(2)构建概率分类器学习数据分布，输出待清洗数据的确定度，用于判断修复是否应用。(3)将学习模型最不能确定的数据交与用户处理，保证了所有清洗数据的正确性。(4)选择模型从用户清洗的数据中，选出高分类贡献度的数据补充到训练集中，不断的提升模型的修复能力。(5)系统的交互

基金项目：本文工作受国家自然科学基金(编号61370205)，上海市自然科学基金(编号13ZR1400800)资助.

界面简洁便利。

2 系统实现(System implementation)

2.1 系统结构

系统结构主要分为三个部分：数据存储、学习模块、选择模块。系统结构图如图1所示。

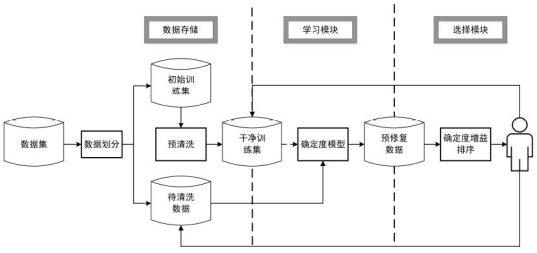

图1 系统结构图

Fig.1 System structure diagram

(1)数据存储部分。这一部分负责数据的准备。主要功能包括：a.初始训练集的划分；b.干净训练集的准备和存储；c.待清洗数据的存储。数据经过一系列的初始处理，然后传递给学习模块，为确定度模型提供数据支持。

(2)学习模块部分。这一部分负责确定度模型的维护和运行。学习模块是ADC系统的核心模块，它训练并存储一个基于概率分类器的确定度模型，并为选择模块生成一系列其建议修复的数据。这些修复数据会带上一个确定度值，以标识模型对此修复的确定程度。

(3)选择模块部分。这一部分主要是修复选择和人机交互过程。选择模块运行筛选规则，选出确定度模型建议的容易出错的修复交予用户查看。然后，根据用户查看与否将数据划分为干净数据和待清洗数据，反馈给数据存储部分，扩充干净训练集，以此迭代操作增强学习模块的确定度模型，提升系统的清洗能力。

2.2 系统流程

ADC系统是一个迭代清洗系统，数据会在数据存储部分、学习模块、选择模块之间循环处理并发挥作用直至系统对数据清洗完成。

所有数据由数据存储部分存储，ADC系统首先划分出一块数据量很小的数据集(200—300条记录)，将其交予用户查看清洗后传输给学习模块用于构建初始模型。同时根据这部分数据的干净程度，计算脏数据(待清洗数据)的原始确定度$certainty_{\text{initial}}$。剩下的大量待清洗数据将被划分为多块，分批输入到模型进行处理。

学习模块接收数据存储部分传输的小部分干净数据集，通过分析数据的概率分布训练出初始的概率分类器(ADC系统使用贝叶斯分类器)，计算每一个分类结果的确定度。确定度的计算遵循BvSB准则。学习模块最后根据每个结果的确定度是否大于原始确定度把数据分为建议修改数据和不做修改数据两部分，使用确定度的分类器能有效的检测脏数据并给出正确修复。因为自动化修复模型始终存在一定的错误率，尤其是前期，所以建议修复数据将传输到选择模块进行人工检查。与修复数据一起传输的还有每一条修复的确定度增益$certainty_{\text{gain}}$，用于选择模块筛选数据。同理，不做修改的数据可能存在为检测出的脏数据，所以将其返回待清洗数据集中，直到最后一轮处理。

选择模块接收学习模块建议的修复数据，以及对应的确定度增益，运行筛选规则，按照确定度增益排序，设定一个阈值将修复数据分成两部分并做相应处理：一部分修复数据是确定度模型认为存在错误的可能性较大，模型对这部分数据正确性确定程度不高，因此将其交予用户查看；另一部分修复是模型认为存在错误可能性较低，这一部分数据则不提交用户查看。阈值的设定可根据对数据质量的要求作相应变化。阈值越大，用户查看的数据量越多，用户参与度越大，数据质量越好；阈值越小，用户参与度越小，数据质量在确定度模型修复结果上提升效果不明显。用户查看清洗过的数据并将其返回到干净数据集，补充确定度模型的基础数据量，完善数据分布，通过迭代学习的方式提升模型清洗能力；用户未查看的数据将返回到待清洗数据集中，等待最后一轮清洗。

在待清洗数据集中所有数据块都执行了一次清洗后(原划分的数据块清洗完成)，模型清洗能力有了极大提升，最后将所有待清洗数据集输入模型进行二次清洗，经过学习模块、选择模块得到最后的数据，完成整个清洗任务。

2.3 关键技术

根据系统结构图1，关键技术点划分如下：

(1)数据划分：这是数据的准备工作。系统需要将数据划分成用于模型训练的数据和给予模型迭代修复的数据。其中，用于模型训练的数据很少，因此人查看该部分数据的工作量少，同时训练集不会包含太多对提升模型清洗能力作用较小的数据，使模型迭代增强的速度更快。ADC系统从原始数据中随机抽取用于训练的数据集，以保留原始数据的概率分布。给予模型修复的数据会被划分为很小的(100行数据)数据块，ADC每次选择一个数据块清洗，形成迭代的清洗过程。

(2)预清洗：人清洗初始训练集的过程。在人清洗初始训练集过程中，通过统计正确数据的数量，可预估整个数据集的数据质量。预估的数据质量将被用于计算原数据值的确定度$certainty_{\text{initial}}$，$certainty_{\text{initial}}$反映数据保留原值的确定程度。计算公式如下：

$$certainty_{\text{initial}} = p(right) - \frac{1-p(right)}{nunique(y)-1} \tag{1}$$

被减数$p(right)$表示保持样例类别的概率，减数表示样例类别修改为其他某一个值的平均概率。$certainty_{\text{initial}}$将辅助确定度模型提供更可靠的修复建议。

(3)确定度模型：使用确定度指标筛选最终修复的模型。确定度反映了模型对给出的修复建议的确定程度，BvSB方法使用模型最好的猜测和次好的猜测来决定模型对样例的确定度，计算公式如下：

$$certainty = p(y_{\text{Best}} \mid x) - p(y_{\text{Second-Best}} \mid x) \tag{2}$$

$p(y_{\text{Best}} \mid x)$和$p(y_{\text{Second-Best}} \mid x)$分别表示样例$x$属于最优类标和次优类标的概率。

当$certainty > certainty_{\text{initial}}$时，对这条数据执行更新，生成建议修复值，否则保留原值不变。

(4)确定度增益排序：筛选模型最不确定建议修复的数据。ADC系统计算确定度增益，用于区分建议修复值与原值的差距，以此选出模型相对不确定的值。确定度增益计算公式如下：

$$certainty_{\text{gain}} = certainty - certainty_{\text{initial}} \tag{3}$$

$certainty_{\text{gain}}$越小，表示模型给出的建议修复值与原值的确定程度越一致，即分歧越大，模型自身划分错误的可能性就越大，将这部分数据交与人查看可有效避免模型的错误决策

对数据清洗带来的负面影响。

(5)交互结果展示：在用户查看建议修复数据时，统计每m个检查过的数据中系统判断正确和错误值的比例并展示，用于辅助用户判断系统修复能力，以便提早进入下一轮的修复。m过大，会使得用户查看更多的数据才能得到统计反馈；m过小使得统计值波动变大，影响用户判断。系统默认$m=\#columns_{error}\times 2$($\#columns_{error}$为错误数据所在列的列数)，即用户至少需查看两行数据。

3　系统概述(System overview)

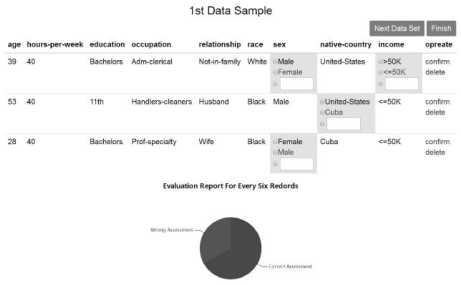

图2 部分系统交互界面

Fig.2 Part of system interaction interface

本文使用UCI(http://archive.ics.uci.edu/ml/)上的adult数据集，在sex、native-country、income属性列上插入错误并对它们进行清洗。图2展示了系统与用户交互的部分界面。系统将模型给出的修复用淡绿色高亮展示，并设置三种用户交互方式：(1)选择原数据值，即黑色记录；(2)选择建议修复数据值，即红色记录；(3)用户手动给出正确值，即文本框输入。对一行修复建议完成判断后，用户可确认以上操作或者删除这条数据。然后系统根据用户反馈，展示每六条数据模型给出正确建议的比例，用饼状图表示，以评估数据清洗的程度，使用户可以根据对模型清洗能力的满意度来提前结束当前数据块的修复。当用户对整个模型清洗能力满意时，点击“Finish”按钮可结束迭代过程，进入最终的清洗过程。

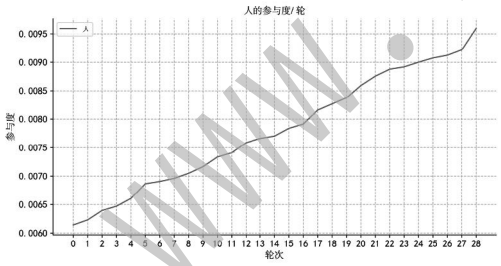

图3 实际用户参与度

Fig.3 Actual user engagement

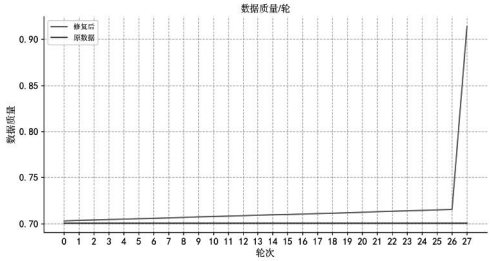

图4 数据质量随迭代次数增加的变化

Fig.4 Change in data quality with increasing number of iterations

图3和图4展示了交互清洗过程中用户参与度和数据质量的变化。人参与度定义为：人查看的记录数占所有需要清洗的记录的比例。数据质量的定义为：清洗后的数据中，正确记录占所有需要清洗的记录的比例。计算公式如下：

$$quality=\frac{\#recodes_{\text{right}}}{\#rows\times\#columns_{error}} \tag{4}$$

其中$\#recodes_{\text{right}}$是清洗后正确的数据数，$\#rows$为数据行数，$\#columns_{error}$是错误数据所在列的个数。由图3可看出，ADC系统只需要很少的用户参与，就可以极大地提高数据质量，从而提升了数据清洗的效率。

实验在adult数据集sex、native-country、income属性上插入30%的脏数据，用于系统清洗。干净数据用于模拟用户，对系统给出的建议修复，给出正确选择。用户满意度设置为：用户每检查6个值，模型错误率低于10%，则用户对这一批数据满意；若连续5轮，用户查看的值小于15个，则对模型的清洗能力满意。

由图3可看出，ADC系统只需要很少的用户参与，就可以极大地提高数据质量，从而提升了数据清洗的效率。第0轮清洗，用户检查300条数据用于初始模型的构建，所以初始参与度不为0。随着用户不断清洗每一批数据(100条数据)，用户的参与度不断提升，但相对上万条的数据总量，参与度提升并不快。最后一轮清洗，系统将剩余未查看的数据处理后交与用户查看，由于这部分数据量较大，所以用户参与度明显升高。然而经过多轮的清洗，模型的清洗能力不断增强，建议修复数据的错误率降低，使得用户最终的参与度很低，不到数据总量的1%。

由图4可以看出，ADC系统经过多轮迭代清洗后，数据质量从70%提升到90%以上。前27轮清洗，因为只有用户查看的数据被清洗，所以数据质量提升不高。最后一轮清洗，用户查看一部分系统建议修复的数据，而剩下的大量未查看的修复被应用。因为修复模型经过多轮迭代增强，使得最终修复正确率很高，系统再将模型不确定的修复交与用户查看后，数据质量显著提升。

4　结论(Conclusion)

本文研究的基于主动学习的清洗系统(ADC)，结合了机器学习的高效性和人为参与的准确性。在基概率分类器上，加入确定度指标，将模型建议修复值与原值相比较，提高了模型清洗的准确度。另外，使用确定度指标发现模型可能错误的修复建议，将这少部分的错误交与人查看，弥补了自动清洗准确性不能保证的不足，使得数据质量进一步的提升。进一步的研究方向包括：(1)发现和使用准确度更高的清洗模型；(2)发现更有效地识别出模型的错误建议的方法。

参考文献(References)

[1] English L P.Information Quality Applied: Best Practices for Improving Business Information,Processes and Systems[M]. Wiley Publishing,2009.

[2] Fei C,Miller R J.A unified model for data and constraint repair[C].IEEE, International Conference on Data Engineering. IEEE Computer Society,2011:446-457.

[3] Chu X,Ilyas I F,Papotti P.Holistic data cleaning:Putting violations into context[C].IEEE International Conference on Data Engineering. IEEE Computer So-ciety,2013:458-469.

[4] Geerts F,Mecca G,Papotti P,et al.The LLUNATIC datacleaning

framework[C].VLDB,2013:625-636.
[5] Song S,Cheng H,Yu J X,et al.Repairing vertex labels under neighborhood constraints[J].Proceedings of the Vldb Endowment,2014,7(11):987-998.
[6] Mayfield C,Neville J,Prabhakar S.ERACER:a database approach for statistical inference and data cleaning[C].ACM SIGMOD International Conference on Management of Data. ACM,2010:75-86.
[7] Yakout M,Elmagarmid A K.Don't be SCAREd:use SCalable Automatic REpairing with maximal likelihood and bounded changes[C].Acm Conference on Management of Data,2013:553-564.
[8] Fan W,Li J,Ma S,et al.Towards certain fixes with editing rules and master data[J].Vldb Journal,2012,21(2):213-238.
[9] Raman V,Hellerstein J M.Potter's Wheel:An Interactive Data Cleaning System[C].International Conference on Very Large Data Bases.Morgan Kaufmann Publishers Inc,2001:381-390.
[10] Volkovs M,Fei C,Szlichta J,et al.Continuous data cleaning[C]. IEEE,International Conference on Data Engineering. IEEE,2014:244-255.
[11] Yakout M,Elmagarmid A K,Neville J,et al.Guided Data Repair[J].Proceedings of the Vldb Endowment,2011,4(5): 1223-1226.
[12] Chu X,Morcos J,Ilyas I F,et al.KATARA:reliable data cleaning with knowledge bases and crowdsourcing[J].Proceedings of the Vldb Endowment,2015,8(12):1952-1955.
[13] Joshi A J,Porikli F,Papanikolopoulos N.Multiclass active learning for image classification[C].Computer Vision and Pattern Recognition, 2009.CVPR 2009.IEEE Conference on.IEEE,2009:2372-2379.
[14] Yakout,Mohamed,Elmagarmid,et al.GDR:a system for guided data repair[J].Sigmod,2010,4(5):1223-1226.
[15] Chu X,Morcos J,Ilyas I F,et al.KATARA:A Data Cleaning System Powered by Knowledge Bases and Crowdsourcing[J]. Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data,2015:1247-1261.

作者简介：
郭开彦(1992-)，男，硕士生.研究领域：数据清洗.
王洪亚(1976-)，男，博士，教授.研究领域：数据库理论与系统，实时计算，移动计算.
程炜东(1994-)，男，硕士生.研究领域：数据清洗.

(上接第53页)

开发一个实用的简单程序。作为补充，开设程序设计实训，以程序设计课程内容为基础，引入软件工程的理念，引入事件响应机制，引入事件响应函数。

第一阶段，给学生提供一个具有完整程序框架的简单示例，包括结构定义、主程序(空)、主程序实现说明、事件响应函数(空)、响应函数的实现说明。首先教师介绍完整程序框架的各部分，学生根据响应函数实现说明实现响应函数、根据主程序实现说明实现主程序。

第二阶段，给学生提供具有完整程序框架的复杂示例，其结构和内容与第一个示例类似，但不在提供实现说明，要求学生实现。

第三阶段，根据要求，学生参考前两个阶段的练习内容，自主设计内容，实现一个完整的软件，包括程序和设计、实现文档。

以C语言程序设计实训为例，采用了Funcode平台，编写一个小游戏，比如传统的挖金矿、坦克大战等，平台本身提供了程序框架和空函数，学生只要实现具体的函数内容。程序实现不再是枯燥的字符，而是声音、图像，图形界面和可视化编程方式极大地激发了学生的热情。在课程提供的基本游戏示例基础上，学生衍生了更多的创意。

利用Python语言生态圈，可以实现丰富的功能，如数据处理分析、图形表示、图像处理、专业应用、三维可视化等。功能系统、工具创意等使Python实训更加多彩。

程序设计实训不仅练习程序设计课程中介绍的基本内容，如分支循环等程序结构、数组结构体等构造类型，还向学生展现了窗口程序、应用系统开发的方法，弥补了程序设计基本知识与常用软件形式之间的鸿沟，从软件开发的高度向学生潜移默化程序设计的理论和方法。

7 结论(Conclusion)

对于大部分非计算机专业，其程序设计课程目标是掌握基本程序设计方法，而解决实际问题的算法往往比较复杂。Python生态圈解决了这一矛盾，只要学生找到合适的函数库，调用即可，因此学习Python语言可以解决大部分问题。但是对于特殊问题，没有现成的函数库使用，或者现有库不能满足要求，需要自行开发，则需要C/C++语言。

程序设计课程重点在设计而非语言，在强化算法能力、计算思维的同时还需兼顾实用软件开发。非计算机专业的程序设计课程，囿于语言选择、课程学时与学生基础，往往脱离了本来目标，偏向语言传授。本文以学生为中心，结合自学与在线练习，重点提高程序设计能力，并以实训的方式跨越基础程序设计与应用软件开发的鸿沟，取得了良好效果。

参考文献(References)

[1] 周世平,童向荣,卢云宏.程序设计基础课程改革方案探讨[J].计算机教育,2015(3):84-86.
[2] 郭银章,王丽芳.基于项目任务驱动的C语言程序设计课程教学改革与实践[J].计算机教育,2017(2):41-44.
[3] 宛西原,汪霞.非计算机本科专业计算机程序设计课程的改革思考[J].计算机工程与科学,2014(4):56-59.
[4] 车万翔,苏小红,袁永峰,等.计算机专业高级语言程序设计课程改革探索[J].计算机教育,2014(13):56-63.
[5] 嵩天,黄天羽,礼欣.Python语言:程序设计课程教学改革的理想选择[J].中国大学教学,2016(2):42-47.

作者简介：
刘培刚(1979-)，男，博士，讲师.研究领域：三维地质建模与可视化.本文通讯作者.
杨劭辉(1967-)，男，本科，讲师.研究领域：计算机应用.
张学辉(1977-)，男，硕士，副教授.研究领域：计算机应用.