Data Debugging (Full Presentation)

Daniel W. Barowy Dimitar Gochev Emery D. Berger
University of Massachusetts, Amherst

Abstract

Testing and static analysis can help root out bugs in programs, but not in data. We introduce *data debugging*, an approach that combines program analysis and statistical analysis to find potential data errors. Since it is impossible to know *a priori* whether data are erroneous or not, data debugging locates data that has an unusual impact on the computation. Such data is either very important, or wrong. Data debugging is especially useful in the context of data-intensive programming environments that intertwine data with programs in the form of queries or formulas. We present the first data debugging tool, CHECKCELL, an add-in for Microsoft Excel. CHECKCELL colors cells red when they have an unusually high impact on the spreadsheet's computations. CHECKCELL is both analytically and empirically fast and effective; in a case study, it automatically identifies a key flaw in the infamous Reinhart and Rogoff spreadsheet.

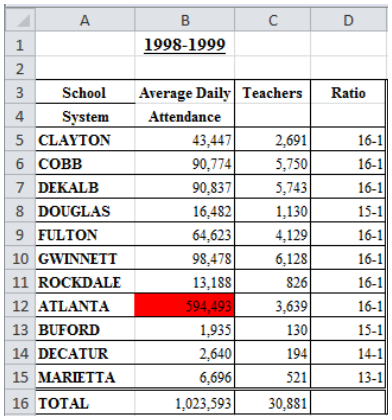
Problem: Locating Input Errors

Most work in the programming language community has focused on ways to discover whether the program performing the computation is correct. However, a program is just one part of a computation. Existing tools ignore the correctness of program *inputs*. Unlike programs, data cannot be easily tested or analyzed for correctness.

Data errors are especially problematic in data-intensive programming environments like databases, spreadsheets, and certain scientific computations. The results produced by the computations—queries, formulas, charts, and other analyses—may be rendered invalid by data errors. These errors can be costly: errors in spreadsheet data alone have led to losses of millions of dollars [5, 6]. As the importance of these applications grow, the scarcity of tools for finding input errors is an increasingly urgent problem.

Our Approach

We present **data debugging**, an approach for locating potential input errors. Since it is impossible to know *a priori* whether data are erroneous or not, data debugging does the next best thing: *locating data that has an unusual impact on the computation*. Intuitively, data that has an inordinate impact on the final result is either very important, or it is wrong. By contrast, wrong data whose presence has no particularly unusual effect on the final result does not merit special attention. Data debugging combines data dependence analysis and statistical analysis to find and highlight data proportional to the severity of its impact on the results of a computation.



	A	B	C	D
1		1998-1999		
2				
3	School	Average Daily	Teachers	Ratio
4	System	Attendance		
5	CLAYTON	43,447	2,691	16-1
6	COBB	90,774	5,750	16-1
7	DEKALB	90,837	5,743	16-1
8	DOUGLAS	16,482	1,130	15-1
9	FULTON	64,623	4,129	16-1
10	GWINNETT	98,478	6,128	16-1
11	ROCKDALE	13,188	826	16-1
12	ATLANTA	594,493	3,639	16-1
13	BUFORD	1,935	130	15-1
14	DECATUR	2,640	194	14-1
15	MARIETTA	6,696	521	13-1
16	TOTAL	1,023,593	30,881	

Figure 1. A sample spreadsheet from the EUSES corpus with a real typographical error, which CHECKCELL highlights in red.

After building a data dependence graph of the computations, data debugging measures data impact by replacing data items with data chosen from the same input group (e.g., an input vector) and observing the resulting changes in computations that depend on that data. This nonparametric approach lets data debugging find errors in both numeric and non-numeric data, without any requirement that data follow any particular statistical distribution. Data debugging pinpoints problems within the data itself.

CheckCell: Data Debugging for Spreadsheets

We built a prototype data debugging tool for Excel called CHECKCELL. CHECKCELL interactively guides a user to fix unusual inputs, highlighted in bright red (Figure 1).

Data debugging reduces effort and error. Since CHECKCELL cannot fix errors without user feedback, we evaluated it with a simulated user who knows the correct value to input once the error has been identified by the tool. We employed human workers via Amazon's Mechanical Turk crowdsourcing platform to generate errors, which we then inserted into a random selection of spreadsheets from the EUSES cor-

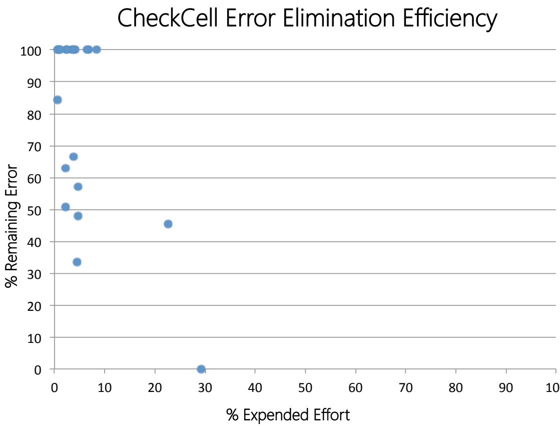


Figure 2. CheckCell can efficiently guide users toward finding the most important errors. Lower and further to the left of the chart is better, indicating that important errors are removed with little effort.

pus [2]. We randomly inserted errors at a rate of 5%, which is consistent with our observations of the typo rate from our Mechanical Turk users. We measure CheckCell's efficacy against two metrics: (1) reduction in the normalized total error of the spreadsheet, and (2) reduction in user effort in terms of the percentage of inputs that needed to be manually inspected (Figure 2).

CheckCell is effective in automatically finding injected errors in 42.8% of the cases (9 of 21 cases). In these cases, CheckCell was able to guide simulated users to reduce overall error by roughly 50%, while examining an average of just 4.5 cells (8.3%). Without CheckCell, a user would need to examine 100% of the cells in the spreadsheet–an average of 54 cells (see Figure 2). On average, CheckCell had one true positive, 3.5 false positives, and 4.8 false negatives. The fact that CheckCell is able to reduce error significantly after only a single correction strongly supports our claim that CheckCell finds the most important errors in a spreadsheet.

Data debugging works with real errors. In 2010, the economists Carmen Reinhart and Kenneth Rogoff presented results of an extensive study of the correlation between indebtedness (debt/GDP) and economic growth [4]. The results of this study were widely used to justify the controversial austerity measures taken to reduce debt loads in many nations affected by the global financial crisis.

The instrument used to perform the actual analysis was an Excel spreadsheet. Economists at UMass Amherst analyzed this spreadsheet and uncovered a number of critical flaws in Reinhart and Rogoff's calculations, which undermine their conclusions [3].

We applied CheckCell to the Reinhart-Rogoff spreadsheet. CheckCell singled out one cell in bright red, identifying it as a value with an unusual impact on the final result. We reported this to Michael Ash, one of the UMass economists invoked in debunking the Reinhart-Rogoff study, who confirmed that this value indicated a key methodological problem in the spreadsheet that they identified through extensive manual auditing [1].

Data debugging is fast. To measure the runtime of CheckCell, we ran it on a selection of 30 spreadsheets drawn randomly from the EUSES corpus. For 18 of the 26 benchmarks, CheckCell takes 30 seconds or less to complete. Its runtime is less than two minutes for all but three of the spreadsheets. The average runtime over all spreadsheets is 49 seconds; without the three outliers, it is 21 seconds. The time cost of CheckCell is largely dominated to the cost of the impact analysis, which depends on the number of inputs.

Potential Applications

We plan to explore data debugging for other data-intensive domains, including scientific computing environments like R and database management systems.

All of these domains will require tailoring of the existing algorithms to their particular context. For databases, we plan to treat as computations both stored procedures and cached queries. While it is straightforward to apply data debugging to databases when queries have no side effects, handling queries that do modify the database will take some care in order to avoid an excessive performance penalty. We are exploring piggybacking our analysis on database transactions, which allow efficient use of concurrent resources and rollback to avoid mutating the database.

References

[1] M. Ash and R. Pollin. Supplemental Technical Critique of Reinhart and Rogoff, "Growth in a Time of Debt". Research brief, Political Economy Research Institute, University of Massachusetts Amherst, Apr. 2013.

[2] M. Fisher and G. Rothermel. The EUSES spreadsheet corpus: a shared resource for supporting experimentation with spreadsheet dependability mechanisms. *SIGSOFT Softw. Eng. Notes*, July 2005.

[3] T. Herndon, M. Ash, and R. Pollin. Does High Public Debt Consistently Stifle Economic Growth? A Critique of Reinhart and Rogoff. Working Paper Series 322, Political Economy Research Institute, University of Massachusetts Amherst, Apr. 2013.

[4] C. M. Reinhart and K. S. Rogoff. Growth in a time of debt. Working Paper 15639, National Bureau of Economic Research, January 2010.

[5] M. Sakal and L. Raković. Errors in building and using electronic tables: Financial consequences and minimisation techiques. *International Journal of Strategic Management and Decision Support Systems in Strategic Management*, 17(3):29–35, 2012.

[6] V. Samar and S. Patni. Controlling the information flow in spreadsheets. *CoRR*, abs/0803.2527, 2008.