Visual Analytic Workflow to Understand Students' Performance in Computer Science Courses

Ravali Gampa
Department of Computer Science
California State University, Sacramento
Sacramento, CA
ravaligampa@csus.edu

Anna Baynes
Department of Computer Science
California State University, Sacramento
Sacramento, CA
abaynes@ecs.csus.edu

***Abstract*—This Work in Progress Research Paper presents a visual analytic workflow to assist instructors of introductory computer science courses to manage their students' learning and success. Sometimes introductory college classes are notoriously called "weed-out" courses, which students who fall behind, are discouraged from continuing the career path. Students with different educational backgrounds may be unnecessarily defeated in these courses. In this work, we identify what class data can be collected and supplied to data analytic tooling to generate insights into monitoring the students' progress. We first investigate a variety of machine learning tools and techniques on a class dataset. Then, we present work-in-progress designs of a visual analytic workflow. Through the interaction with the visual analytic tool, instructors of the introductory computer science course gather insights into the class, for example, "Which part of the programming assignment is causing students to have the most software bugs?," "Which exam questions best test the understanding of runtime analysis?," "What type of student activity results in fascinating over 50% of the class to participate and understand the material?"**

I. Introduction

Exploratory visual analytics provide a system or a pipeline toolset that allow the investigator to examine hypotheses through visual interaction [4]. For example, drilling up and down in the data to find aggregations or communities of common data points or identifying outliers which relate to surprising details in the dataset. Furthermore, educational datasets are becoming easier to collect and access through classroom management tools. Canvas [3] is a classroom management tool widely used in universities to manage course webpages. Using Canvas allows professors to easily build datasets representing their courses by maintaining gradebooks, analytics on students' activity and usage of message boards, file views, and other forms of participation. Instructors can also tag low level details such as specific topics on exams and homework assignments assigned through Canvas. This tagging can help understand student performance on course themes throughout the semester.

In this work we use Data Structures and Algorithms at Sacramento State University to be the hallmark example in exploring the design and usage of the visual analytic tool. Sacramento State University has a diverse student background which has seen high dropout rates in their lower division computer science courses. At Sacramento State University the lower division computer science course "Data Structures and Algorithms" is a meeting point for students from different educational backgrounds. For example, students from several community colleges and experience levels in programming and computer science come to class together to study the same material. The completion of this course determines whether the students can continue as a computer science undergraduate major [1], [20]. Although a bell curve is expected in class performance, there are situations where students lack the support to continue their studies [10], [5]. We wonder if exploratory visual analytics can help instructors utilize class data to gather insights, for example, "Which part of the programming assignment is causing students to have the most software bugs?" and "Which exam questions best test the understanding of runtime analysis?"

In this work-in-progress paper, we have compiled a list of machine learning and analytics tools. These tools include: Trifacta [17], Orange [14], and Knime [8]. We import classwork datasets into these tools and attempt a typical workflow to find insights. Next, we discuss our steps in aggregating and anonymizing datasets to study the introductory computer science course. Finally, we provide a discussion on designing the exploratory visual analytic workflow. The rest of the paper is organized as follows: related work, background investigation on existing analytic techniques, identifying class attributes and datasets, and presentation of designs for the visual analytic tool.

II. Related Work

The intersection of education and computer science offers several related works which drive our motivations [20], [9]. For example, [2] studies undergraduate computer science students and investigates the high rate of failure in programming courses through surveying the students various questions about how they feel about the campus, courses, and support they receive from the university. Based on their survey they recommend several changes to improve the way computer science is offered at their university. This work provides us with suggestions on information which should be included in the classroom datasets for exploration. [15] also examines data structure courses for computer science students. They survey students with open-ended questions to understand

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

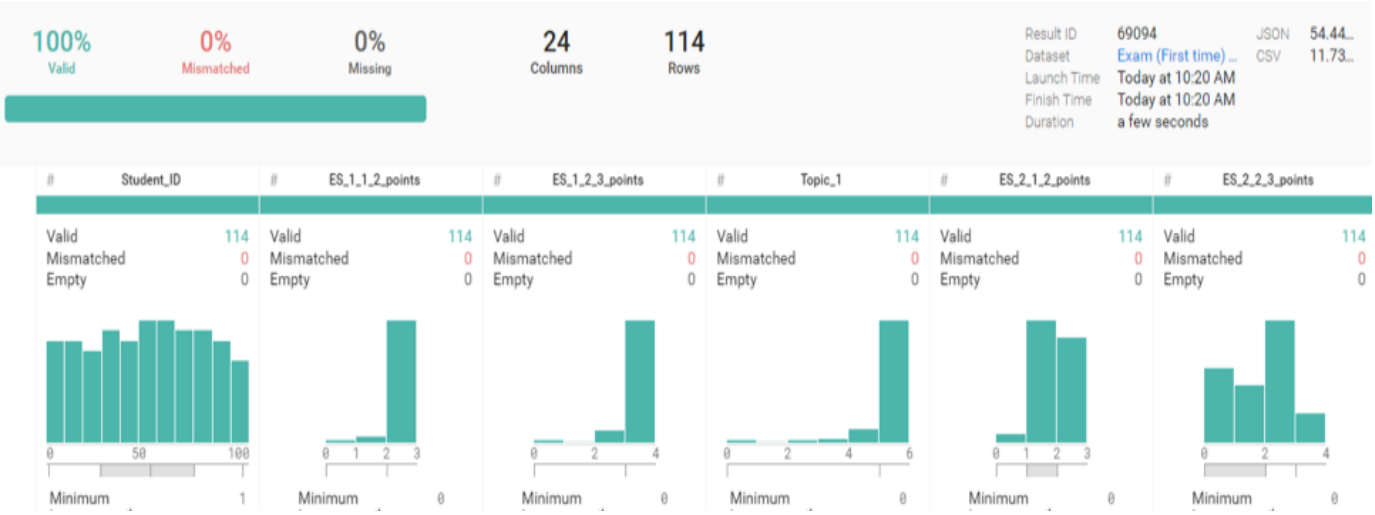

Fig. 1. Trifacta Wrangler has operations such as delete, sort, select. Above we see the data modified with these operations.

concepts which are difficult to understand and how the student overcame this difficulty. Based on their study they present recommendations on how their findings can be operationalized in the classroom. We have similar objectives in our work, but our viewpoint is to create visual exploration tools for the instructor to find insights in existing classwork data. [7] presents exploratory analysis on computer science classwork text data. Our work extends this research by exploring how to create a workflow of different analytic tools. In this work, we tackle these open issues to provide a targeted solution specifically for lower-division computer science courses.

In response to increasing computer science enrollment, [18] attempt to identify correlations of grades from student transcripts to determine what patterns of grades indicate future success in senior level computer science courses. Our work focuses on a particular class while [18] studies how the lower division courses' outcomes affect the later studies. In addition to studying student success in computer science courses, there has been work which researches applying different technologies into the classroom [12], [16]. [12] recognizes that crowd sourcing in teaching and learning have not been fully explored in higher education. They present a set of steps to incorporate crowd sourcing in computer science education.

III. Current Tools and Technology Investigated

In the following section, we describe the classroom dataset and our analysis method used on the various machine learning and analytic tools: Trifacta [17], Orange [14], and Knime [8].

A. Experimental Design and Dataset

In the first phase of the project, we explore current popular exploration tools on an academic dataset [19]. The dataset was created by collecting 115 first year undergraduate engineering students performance on exam topics with increasing difficulty. The dataset contains the students' time series of activities during six topics. We utilized the final grades of the students in the experiment for our analysis. The final grades show each student's grades for each individual exercise and the overall grade. Topics 1, 2, 4, and 6 have two problems each; topic 3 has four problems, topic 5 has three problems.

The first step towards analyzing the classroom data is cleaning and preprocessing the data. This step is one of the most important steps to obtain conclusions effectively about the performance of students. Sometimes this process can be tedious depending on the anomalies in dataset. In our dataset we see a mismatch between column names. For example, there are multiple names for the computer science topics (i.e. CS, CSc, Computer Science, Computer Sciences for a single computer science branch). These discrepancies need to be handled before analysis.

The next step towards analyzing the data is working with cleaned and wrangled data to identify insights. The term "data wrangling" is when data is transformed and mapped from the "raw" dataset into a usable dataset for analytics. The term is commonly used in information visualization and analytics. Data exploration tools include visualizations, such as, pie-charts, histograms, scatterplots etc. We will also apply data mining algorithms like decision trees, and regressions to understand the performance of the class.

B. Trifacta Wrangler

With Trifacta Wrangler [17], you can visually explore and understand datasets by cleaning messy data. Trifacta uses flows to organize datasets and recipes to perform operations on the dataset. Figure 1 shows Trifacta after importing the dataset and performing operations. Since we have to merge two datasets, we use the "Union" operation. Next we sort the attributes in order of student ID. The following results summary statistics are obtained: highest score is 98, lowest score is 7, average score is 55. Using the visualization in Figure 1, we see the following insights: 1) Topic 1 is the best understood by the students. 2) Topic 6 needs more practice because more than half the students are below the average. 3) The rest of the topics need more review to improve the performance.

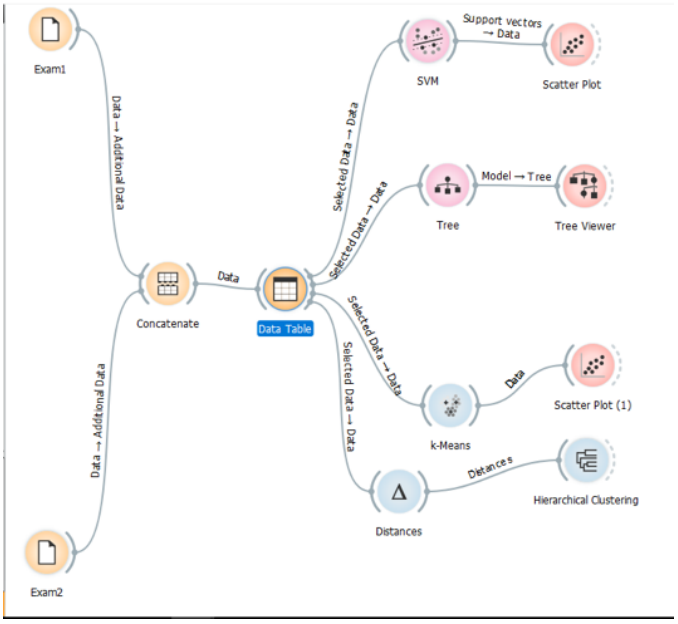

Fig. 2. Orange allows the user to create workflows with different datasets. The tool provides different algorithms like SVM, decision trees, and visualizations.

C. Orange

Next we utilize Orange [14] to explore the dataset. Orange provides supervised learning methods, including SVMs and decision trees and unsupervised methods such as K-means clustering. Orange also can visualize the dataset after obtaining results from the different algorithms. Figure 2 shows the workflow created to perform the analysis with Orange. The analysis shown concatenates the two exams, which contain the different topics. Next, a data table is created which allows different types of operations and visualizations. We found the same conclusions as Trifacta Wrangler using the visualizations produced by Orange.

D. Knime

Knime [8] has basic functionalities such as file read, file write, table manipulation operations, and analysis tools present in its Node repository. The user can drag and drop required functions to create a workflow. We import the classroom dataset into Knime and apply manipulation operations. However, we found it difficult to clean the dataset using Knime and used the wrangled data from Trifacta Wrangler. Different visualizations such as histograms, pie charts, scatter plots, or conditional box plots can be used to represent the results. For example, Figure 3 displays box plot to see how marks are distributed among students for each topic and total marks. Using these visualizations, we are able to obtain the same set of conclusions as we did for Trifacta Wrangler and Orange.

IV. Data Structures and Algorithms Dataset

In the previous section, the dataset focused on exam scores on six topics. We used a limited dataset to perform a preliminary investigation on the current tools. Next we study datasets

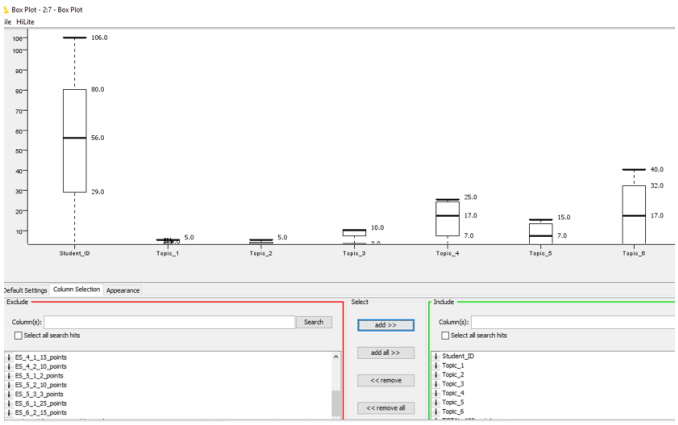

Fig. 3. Above Knime was used to create a boxplot to show how total marks are distributed among students for each topic and each exam.

with more attributes which we collect through Canvas [3]. The dataset includes the following attributes: student names, exam grades, exam responses (e.g. a student's solved exam), exams (e.g. an unsolved exam), exam solutions, homework grades, homework responses, homework assignments (e.g. unsolved homework assignments), homework solutions, activity on Canvas by date (e.g. a categorical data type whether a student participated or viewed a file on Canvas), communication with instructor by date (e.g. this is a tuple data type containing the communication text and date), and the class syllabus.

An issue sensitive to academic classroom datasets is anonymity. If an instructor is viewing her class only, student names and sensitive information do not need to be encrypted or anonymized. But, if the instructor would like to compare to other offerings of the course and share datasets, the data must be anonymized. The benefit of sharing datasets is increasing the sample size. A new instructor would have more examples with which to compare her students.

V. Proposals for Visual Analytic Workflows

In this section we discuss the limitations of the current tools for understanding students performance. Also, we present our current progress to fulfill these gaps. The tools we investigated provide data cleaning, summary and regression statistics, machine learning, and a variety of chart typologies. We used charts such as boxplots and dendrograms.

The current analytic tools are generalized for any application. The tools require the user to be familiar with their context, dataset, and starting direction for exploration. Consequently, instructors might find it difficult to begin their focused analysis on their class datasets. We propose bridging the gap between these tools and an instructor's analytic needs with visual analytic exploration operations specialized for educational datasets. This focused tooling includes scaffolding to explore educational datasets.

To explain this process, we utilize the flow chart in Figure 4 with a scenario from the Data Structures and Algorithms course. The flow chart shows a single iteration of an interactive visual exploration. The process begins with data collection,

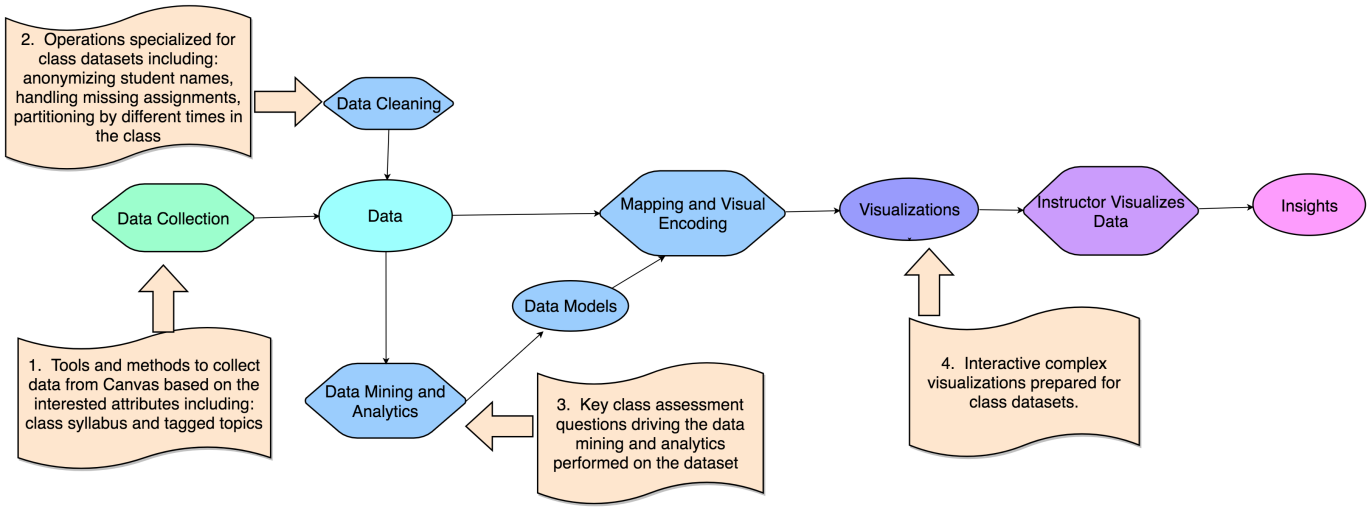

Fig. 4. The different phases of the visual analytic flow chart are supplemented with operations to support analytics on educational datasets.

which produces the data. Data cleaning operations can then be applied to the data. Next either the data can be mapped to visualizations or further data mining and machine learning operations can be performed before creating visualizations. The instructor is then able to explore the visualizations to identify insights.

To improve this process for the classroom setting, we propose improvements in the following ways: During the data collection phase, noted by 1 in Figure 4, our tool will provide operations to pull data from Canvas based on interested attributes. An interested attribute could be a topic outlined in the class syllabus. For example, the class will generally be introduced to algorithm runtime analysis techniques in the beginning of the semester. Then throughout the course, each time a new algorithm is introduced, the students will be required to do a runtime analysis. Therefore, "runtime analysis techniques" is considered a topic. The next improvement, shown in 2 of Figure 4, are additional operations in data cleaning. For example, an instructor may need to share her final visualizations, so the data would need to be anonymized.

In the Data Mining and Analytics phase, our tooling will provide targeted questions for the Data Structures and Algorithms course. Some of these questions include: "What level of student participation on Canvas correlates with passing the first exam?, " "Do students who correctly answered questions on binary trees also answer correctly questions on graphs?," and "Based on their answers to the first exam, which students need more help to pass the class?" We build a library of common themes and questions for the class. These questions drive which analytics and machine learning operations to perform on the dataset.

This example leads to our next scaffolding; our tool will provide interactive visualizations. For example, for the "runtime analysis techniques" topic, we can use an interactive timeline visualization to see how students' understanding of the topic changes throughout the semester. The next steps in our research are to design the interactive visualizations including timeline views, network, and enclosure diagrams. Our goal is to create a tool that after exploration helps build a narrative about the course. The narrative visual report would highlight questions centered around students' understanding.

An exploratory visualization will support the discovery of different classes of questions, for example "known-knowns", "known-unknowns," "unknown-knowns", and "unknown-unknowns." In the "known-unknown" case, the instructor may know of a subject (e.g. runtime analysis) she would like to investigate; but may not know the particular attributes (i.e. exams, homework assignments) involved. The "unknown-unknown" case occurs when a new instructor receives back the papers of the first exam she wrote. In this case, the instructor has no knowledge of how students will perform. She will need to grade and review the entire results to identify both usual and unusual cases of student performances. An exploratory tool can facilitate this browsing. Based on our related works section, several works in the past explored key identifying questions to assess classes and the needs of students. The tool we propose includes these expert curated questions and analysis targets. Finally, the "known-knowns" and "unknown-knowns" both explain cases where the instructor is already aware of an insight in the dataset, which may or may not be found yet.

VI. Conclusion

We present initial work on the datasets collected and the visual exploratory tool strategy. We propose designing a visual analytic workflow specialized for class datasets. We present a visual analytic flow chart supplemented with improvements which will support classroom datasets. We provide examples specific to a Data Structures and Algorithms course. We are currently designing complex interactive visualizations which are tied to key theme questions to assess students' performance.

REFERENCES

[1] J. Bennedsen and M. E. Caspersen, "Failure rates in introductory programming". ACM SIGCSE Bulletin, 2007.

[2] E. D. Canedo, G. A. Santos, and S. A. Andrade de Freitas. "Analysis of the Teaching-learning Methodology Adopted in the Introduction to Computer Science Classes." IEEE Frontiers in Education Conference, 2017.

[3] Canvas, https://www.canvaslms.com/research-education, Last Accessed: 04/23/2018.

[4] S. K. Card, J. D.Mackinlay, B. Shneiderman, "Readings in Information Visualization: Using Vision to Think". 1999. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[5] A. P. Carnevale, N. Smith, and M. Melton. "STEM: Science Technology Engineering Mathematics". Georgetown University Center on Education and the Workforce, 2011.

[6] Google Refine, http://openrefine.org/, Last Accessed: 03/29/2018.

[7] M. Jujare and A. Baynes. "Exploration of Text Analytic Tooling on Classwork to Support Students' Learning in Information Technology." ACM Proceedings of the 18th Annual Conference on Information Technology Education, 2017.

[8] Knime, https://www.knime.com/, Last Accessed: 04/13/2018.

[9] E. Lahtinen, K. Ala-Mutka, and H.-M. Jaarvinen, "A study of difficulties of novice programmers, " Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education, 2005.

[10] M. Meyer and S. Marx, "Engineering Dropouts: A Qualitative Examination of Why Undergraduates Leave Engineering". Journal of Engineering Education, 2014.

[11] P. Pitterson, Nicole, et al. "Investigating current approaches to assessing teaching evaluation in engineering departments. " IEEE Frontiers in Education Conference, 2016.

[12] W. Simao De Deus, H. M. Machado, R. M. Barros, J. A. Fabri, and A. L'erario. "Enhancing Collaboration among Undergraduates in Informatics: A Teaching and Learning Process Based on Crowdsourcing." IEEE Frontiers in Education Conference, 2017.

[13] Lipika Dey and Ishan Verma. 2013. "Text-Driven Multi-structured Data Analytics for Enterprise Intelligence". In Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 03 (WI-IAT '13). IEEE Computer Society, Washington, DC, USA, 213220.

[14] Orange, https://orange.biolab.si, Last Accessed: 03/29/2018.

[15] D. Simkins and A. Decker. "Examining the Intermediate Programmers Understanding of the Learning Process." IEEE Frontiers in Education Conference, 2017.

[16] K. Taghipour and H. T. Ng. 2016. "A Neural Approach to Automated Essay Scoring". In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (Association for Computational Linguistics). 1882 1891.

[17] Tifacta Wrangler, https://www.trifacta.com/start-wrangling/ , Last Accessed: 04/10/2018.

[18] D. Trytten, A. Mcgovern. "Moving from Managing Enrollment to Predicting Student Success." IEEE Frontiers in Education Conference, 2017.

[19] M. Vahdat, L. Oneto, D. Anguita, M. Funk, M. Rauterberg. "A learning analytics approach to correlate the academic achievements of students with interaction data from an educational simulator". In: G. Conole et al. (eds.): EC-TEL 2015, LNCS 9307, pp. 352-366. Springer (2015).

[20] B.C. Wilson and S. Shrock, "Contributing to success in an introductory computer science course: a study of twelve factors, " SIGCSE Bull., vol 33, no. 1, 2001.