ASEE's 123rd Annual • Conference & Exposition • New Orleans, LA • June 26-29, 2016

Paper ID #17067

WORK IN PROGRESS: Data Explorer – Assessment Data Integration, Analytics, and Visualization for STEM Education Research

Joshua Levi Weese, Kansas State University

Josh Weese is a PhD candidate in the department of Computer Science at Kansas State University. Focusing on education research, this experience comes from work as a graduate teaching assistant, various outreach programs, and time spent as a NSF GK-12 fellow. His downtime is spent in outreach programs aimed toward enriching local K-12 students' experience in STEM, especially in computer science and sensor technologies.

Dr. William H. Hsu, Kansas State University

William H. Hsu is an associate professor of Computing and Information Sciences at Kansas State University. He received a B.S. in Mathematical Sciences and Computer Science and an M.S.Eng. in Computer Science from Johns Hopkins University in 1993, and a Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 1998. His dissertation explored the optimization of inductive bias in supervised machine learning for predictive analytics. At the National Center for Supercomputing Applications (NCSA), he was a co-recipient of an Industrial Grand Challenge Award for visual analytics of text corpora. His research interests include machine learning, probabilistic reasoning, and information visualization, with applications to geoinformatics, cybersecurity, education, digital humanities, and biomedical informatics. Published applications of his research include structured information extraction; spatiotemporal event detection for veterinary epidemiology, crime mapping, and opinion mining; and analysis of heterogeneous information networks. Current work in his lab deals with: deep learning and spatiotemporal pattern recognition; data mining and visualization in education research; graphical models of probability and utility for data science; and developing domain-adaptive models of large natural language corpora and social media for text mining, network science, sentiment analysis, and recommender systems. Dr. Hsu has over 50 refereed publications in conferences, journals, and books, plus over 40 additional publications.

Work-in-Progress: DataExplorer - Assessment Data Integration, Analytics, and Visualization for STEM Education Research

Abstract

We describe a comprehensive system for comparative evaluation of uploaded and preprocessed data in physics education research with applicability to standardized assessments for discipline-based education research, especially in science, technology, mathematics, and engineering. Views are provided for inspection of aggregate statistics about student scores, comparison over time within one course, or comparison across multiple years. The design of this system includes a search facility for retrieving anonymized data from classes similar to the uploader's own. These visualizations include tracking of student performance on a range of standardized assessments. These assessments can be viewed as pre- and post-tests with comparative statistics (e.g., normalized gain), decomposed by answer in the case of multiple-choice questions, and manipulated using pre-specified data transformations such as aggregation and refinement (drill down and roll up). Furthermore, the system is designed to incorporate a scalable framework for machine learning-based analytics, including clustering and similarity-based retrieval, time series prediction, and probabilistic reasoning.

Keywords

discipline-based education research, data science, information visualization, information retrieval, analytics

Introduction

We describe two primary components of an analytics system for STEM education research, developed for the American Association for Physics Teachers (AAPT). The purpose of this data exploration system is to allow instructors to comparatively assess student performance in intraclass, longitudinal, and interinstitutional contexts. The interface allows instructors to upload course data including student demographics and exams to a secure site, then retrieve descriptive statistics and detailed visualizations of this data.

The first component consists of a rule-based system for pattern analysis that infers multiple common assessment formats with minimal metadata, and in some cases without headers. This paper describes the incremental development of a priority-based inference mechanism with matching heuristics, based on real and synthetic sample data, and further discusses the application of machine learning and data mining algorithms to the adaptation of probabilistic pattern analyzers. Early results indicate potential for user modeling and adaptive personalized recognition of document types and abstract type definitions.

The second component is an information retrieval and information visualization module for comparative evaluation of uploaded and preprocessed data. Views are provided for inspection of aggregate statistics about student scores, comparison over time within one course, or comparison across multiple years. These visualizations include tracking of student performance on a range of standardized assessments including the Force Concept Inventory (FCI).[1] the Force and Motion

Conceptual Evaluation (FMCE) of Thornton and Sokoloff (1998)[2], and the Brief Electricity and Magnetism Assessment (BEMA).[3] Assessments can be viewed as pre- and post-tests with comparative statistics (e.g., normalized gain), decomposed by answer in the case of multiple-choice questions, and manipulated using prespecified data transformations such as aggregation and refinement (drill down and roll up). The system is designed to support inclusion of a range of supervised inductive learning methods for schema inference, unsupervised learning algorithms for similarity-based retrieval, supervised learning for regression-based time series prediction, and Bayesian models for causal inference on the decision support end.

Both informal assessment of the system and intensive user testing on a pre-release version have yielded positive feedback. This feedback is instrumental in feature revision, both to improve system functionality and to plan the adaptation of the design of these two data exploration components to other STEM disciplines, such as computer science and mathematics. Lessons learned from visualization design and user experience feedback are reported in the context of usability criteria such as desired functionality of the pattern inference system.

The paper concludes with a discussion of the system as an emerging technology, the schedule for its deployment and continued augmentation, and the design rationale for user-centered intelligent systems components. The focal point of future work in this area is on facilitating meaningful interactive exploration of the data by multiple types of stakeholders who have been identified for this type of education research portal. This is achieved using a synthesis of data-driven approaches towards information extraction, retrieval, transformation, and visualization.

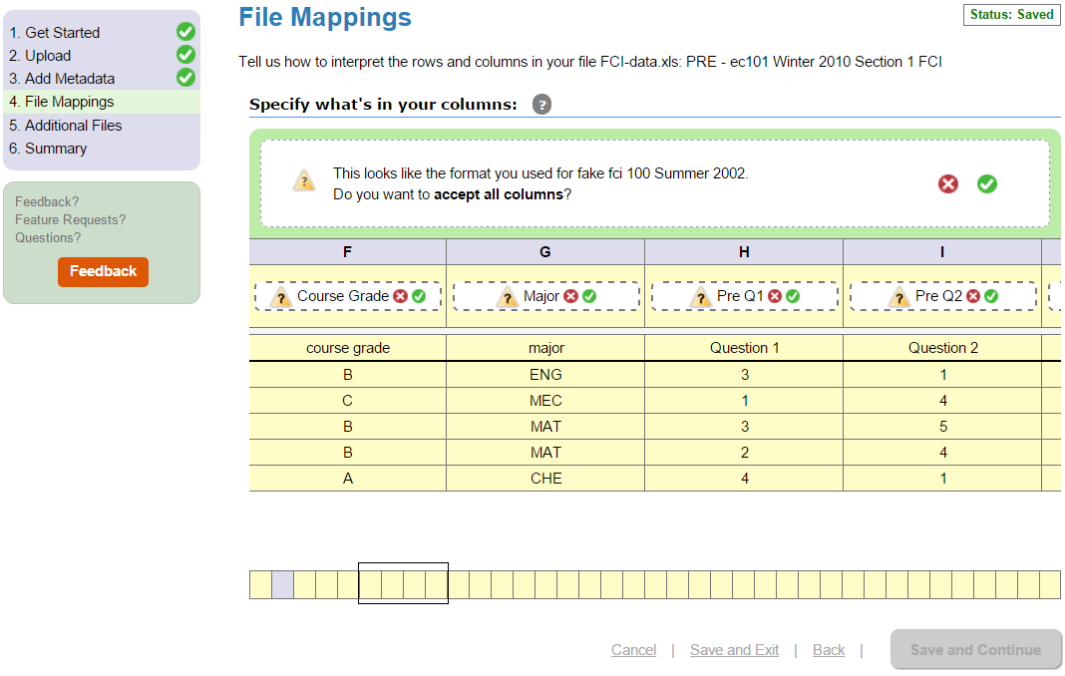

Figure 1. Data Explorer intake interface depicting workflow (left) and example of schema inference and interactive validation (right).

System Overview: Data Explorer

The system (referred to throughout the paper as the Data Explorer) consists of three primary functional modules:

1. Data uploading and preparation, including schema and header inference
2. Information visualization, including breakdown of assessments by question and tracking student performance in courses over time (within-course or longitudinally)
3. Information retrieval, comprising query interfaces and query synthesis

The Data Explorer is a data management system and federated display for educational data that provides data import, integration, interactive validation, and analytics functions. This section describes the first three components, which consist of a data intake front-end where instructors can import assessment data in a spreadsheet format. Next, they can annotate uploaded files by adding metadata for courses and assessment provenance. Then can then specify the organization of data, using a file mapping system that automatically infers the tabular schema of the data. This schema specifies the sequence of columns, similar to a relational database schema but without database normalization requirements. The system infers this schema from sequences of column headers that are scanned and parsed (the *parser* component) from patterns of data formats observed in tabular data (the *guesser* component). The user can then interactively check and edit the result, reviewing the tentative file mapping using the preview shown in Figure 1 and correcting any inference errors. Finally, the result is sent to the analytics and rendering components of the Data Explorer, which prepare descriptive statistics, comparative statistics, and visualizations of the imported data.

Emerging Technology: Data Import, Schema, and Header Inference

The first approach, typified by the work of Keininger (1998[4], 2001[5]) on block segmentation, focuses on matching cells using a neighborhood-based search. Because the intake process for the Data Explorer involves no optical character recognition (OCR) or handwritten character recognition (HCR), we omit layout recognition aspects of the document path and focus on schema inference from delimited files that are either already properly aligned or admit a proper alignment given a correctly inferred schema.

This is closer to the second approach, exemplified in the previous work of Doan, Domingos, and Halevy (2003)[6] on using machine learning to produce classifiers for schema matching. Cafarella, Halevy, Wang, Wu, and Zhang (2008)[7] extend this approach by targeting relational schema and using constraints on relational well-formednesss. More recently, Venetis, Halevy, Madhavan, Paşca, et al. (2011)[8] infer semantic properties of web data by using observed weak typing constraints (isA relations, also known as hyponymy) in online knowledge sources. In a variation on this general approach, we also use pattern matching heuristics and constraints, but restrict our matching to type constraints such as enumerative types on multiple-choice questions.

Finally, the third approach, holistic information extraction from tables, is characteristic of systems such as that of Nagy, Seth, Jin, Embley, et al., (2011)[9], which use syntactic elements of tables – header paths in particular – to extract relational tuples. This approach subsumes tabular data cleaning. For example, Fang, Mitra, Tang, and Giles (2012)[10] use supervised inductive learning to learn the concept of a genuine table (as opposed to spacers and decorative elements),

and also empirically validate heuristics for physical structure analysis (table segmentation, which is obviated in our task) and logical structure analysis. Suchanek and Weikum (2013)[11] examine how to capture such tables in the wild, e.g., as embedded in articles on the web or in print; some relevant ideas from this approach are how to use rule-based data transformations to segment uploaded data (remove headers, trim extraneous elements) and validate them against known good tuples. Adelfio and Samet (2014)[12] specifically address our chief problem of schema extraction for tabular data by using a conditional random field (CRF) classifier learned from data; this approach has achieved marked success in shallow parsing tasks such as named entity recognition in text. Finally, Zhang (2014)[13] re-examines the problem of capturing relations in tables using a combination of named entity recognition and the kinds of semantic constraints applied by the second approach.

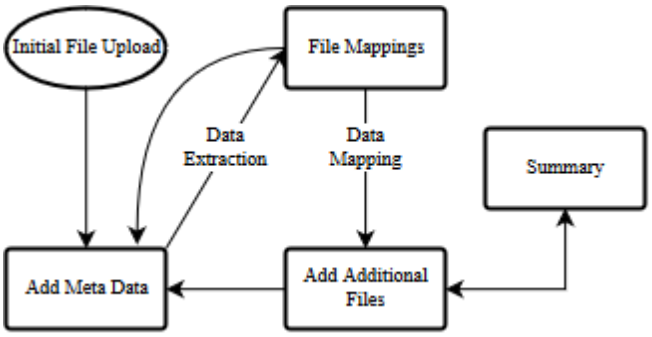

Figure 2. Data flow for importer of Data Explorer.

Our methodology is informed by the latter (schema inference and tuple extraction) approaches described above rather than the first (layout analysis) approach. The users of our system who are usually Physics educators upload their historical assessments through an iterative data upload interface depicted in Figure 2. The data upload interface accepts assessment files that are in a limited set of formats in the current system. The accepted file formats are xls, xlsx, and csv. Simplistic file requirements, which include having a header row and one student per data row, help assure extraction of the correct headers and student data while allowing users to upload a wide range of data formats.

Apart from accepting and verifying the integrity of the uploaded files the data upload interface prompts the user to specify meta information ("Add Meta Data" in Figure 2), such as approximate number of students that took the assessment and whether the file contains either pre-, post-, or pre- and post-test assessment data. Some of these assessment features are required, while others are optional. The assessment specific information, such as assessment name and assessment type (belief survey or standard multiple choice), provide a rough estimate of the number of questions (usually represented as columns) that are present within the uploaded, whereas the number of students gives an estimate of the number of rows with student scores. The data upload interface checks the integrity of the file and extracts all the data that is present within the various file types. The extracted data is saved as a *data frame*, a two-dimensional data structure, where the atomic data items present in the input file are stored in individual cells of the

data frame. The row-column relationships of the data items in the uploaded files are preserved in the data frame.

The objective of the file parser is to identify the boundaries of the assessment scores within the data frame, as well as identify the location of the headers. The presence of other extraneous legacy information within the data makes the task of extracting payload data from the data frame a complicated exercise. Some of the various kinds of information that is available within these files, apart from the payload, could be the rubric or the scoring criteria for the particular assessment; it could also have information dealing with aggregate student demographic information and other extraneous data. Considering all these variabilities, we create a heuristics based parser that takes the meta information that is provided during the file upload process to extract the valid assessment payload from the test data. The presence of both pre- and post-assessment scores within the same data frame is another degree of freedom that adds to the complexity of the parsing approach.

Heuristic ($\boldsymbol{\alpha}$)	Description	Condition to Count ($\boldsymbol{\sigma}$)	Contributed Value ($\boldsymbol{\gamma}$)
String cells	The number of cells in a row that are text.	$> \boldsymbol{thresh}$	1
Integer cells	The number of cells in a row that contain integers.	$> \boldsymbol{thresh}$	1
Float cells	The number of cells in a row that contain floating-point numbers	$< \mathbf{0}$	-1
Duplicate cells	The number of duplicate cells in a row	$> \boldsymbol{thresh}$	1
Unique cells	The number of unique cells in a row	$< \boldsymbol{numberOfQuestions}$	-1
Pre/Post	Detects whether or not the row contains “pre” or “post”	$> \mathbf{0}$	1
Long question number	Detects the number of large question numbers (helps when assessment data is outputted by online tools)	$> \boldsymbol{numberOfQuestions} - \mathbf{10}$ $\boldsymbol{numberOfQuesions} > \mathbf{0}$	1
Max consecutive number	Detects the largest consecutive number series in a row after stripped of alpha characters (Q1, Q2, Q3, etc.)	$> \boldsymbol{thresh}$	3
Unique markers	The number of unique known headers (Student ID, Gender, etc.)	$> \mathbf{1}$	2
Repeated markers	The number of repeated known headers (question, ques, q, pre, post)	$> (\boldsymbol{thresh} - \mathbf{3})$	2

Table 1. Heuristics for identifying the header row.

In order to identify the boundaries of the payload within the data, we first start by identifying the header row of the payload. The header row consists of column names of the various columns available in the assessment scores. These could be student particulars such as name, identifier, or gender, or the particular assessment information, such as grade, question number, or aggregate score. Our model consists a series of heuristics that score rows and columns for identifying which row contains column headers, and which rows contain the student data. This helps eliminate user added calculations and miscellaneous data, and extracts relevant student information. Table 1 shows the heuristics for determining the header row**,** where $\boldsymbol{numberOfQuestions}$ is equal to the number of questions in the assessment (collected in the add metadata phase) and $\boldsymbol{thresh} = \lfloor \boldsymbol{numberOfQuestions} - (\boldsymbol{numberOfQuestions} * \boldsymbol{.2}) \rfloor$.

This threshold gauges an approximate number of columns to expect for questions; the buffer adds tolerance for poorly formatted files. From Table 1, we define the header row to be $\forall \boldsymbol{r} \in \boldsymbol{rows}\ \mathbf{max}(\sum_{\alpha_{i,r}}^{n} \boldsymbol{\gamma_i}\ \boldsymbol{if}\ \boldsymbol{\sigma_i}$ where $\boldsymbol{\alpha_{i,r}^{m}}$ is the heuristics for row $\boldsymbol{r}$. The header row is then used to determine the table boundaries for relevant student data by comparing each row to row markers from known templates; otherwise, in the case a row is absent of markers, the length of the row (number of non-empty cells) is compared to $\boldsymbol{thresh}$, as defined for Table 1. If a row is blank, we use a combination of 80% of the class size (given by the user as metadata) and a two-row margin in order to allow small gaps in student data. If this margin is exceeded, and the number rows in the current block of data parsed is less than 80% of the class size, the start of the student data is moved after the blank rows and parsing continues. This allows the parser to skip over blocks of precomputed statistics and other user specific information; however, if the user gives a greatly over or under estimate on class size, files with more than two row gaps in the data underneath header will be unsuccessfully parsed.

The schema inference model is able to successfully parse 77/80 testing files (a mixture of sanitized real data submitted to the project and synthetic data). A file is parsed successfully if it identified the header row and included all rows of student data. If the parser includes miscellaneous columns of data, the test is allowed to pass as these columns can be excluded in post processing; 23 tests were passed in this manner. The last three tests failed due to the assessment answer keys being included as part of the block of student data. This problem can be solved for templated files; however, for semi-structured files, we are unable to differentiate answer keys from real student data. Accuracy of the schema inference during beta testing and future production deployment is partly dependent on user feedback (missing student rows or columns), as well as the headers that are verified by the user (columns thought to be student data but was not).

The guesser module (interface seen in Figure 1 and position in system as “File Mappings” in Figure 2), uses a hybrid similarity measure to detect approximate matches between candidate header strings and template strings. This consists of a convex combination of two edit distance functions (Levenshtein and Jaro-Winkler), both computed by dynamic programming. The weights are calculated using a generalized logistic function:

$$\boldsymbol{w} = \boldsymbol{Y(t)} = \boldsymbol{A} + \frac{\boldsymbol{K} - \boldsymbol{A}}{(\boldsymbol{C} + \boldsymbol{Q}\boldsymbol{e}^{-\boldsymbol{B}(\boldsymbol{t}-\boldsymbol{M})})^{\frac{\boldsymbol{1}}{\boldsymbol{\nu}}}}$$

where $\boldsymbol{K = C = 1}$, $\boldsymbol{A = 0.3}$, $\boldsymbol{Q = v = M = 5}$, $\boldsymbol{B = 2.7}$, and $\boldsymbol{t}$ is the Levenshtein distance. $\boldsymbol{A}$ is the lower asymptote, $\boldsymbol{K}$ is the upper asymptote, $\boldsymbol{B}$ is the growth rate, $\boldsymbol{M}$ is the baseline distance (input), $\boldsymbol{v}$ is a skew parameter (for controlling the inflection point), and $\boldsymbol{Q}$ is the baseline weight (output). The final distance measure for strings $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$can then be defined as:

$$\boldsymbol{dist(s_1, s_2) = wd_1 + (1 - w)d_2}$$

where $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ are normalized Jaro-Winkler and thresholded Levenshtein edit distances, respectively, $\boldsymbol{d_{JW}}$ is the raw Jaro-Winkler distance and:

$$\boldsymbol{d_1 = \left(1 - d_{JW}\right)^{\frac{t(M-B)}{M}}}$$

The confidence of a column header labeled as a given class is then given by:

$$\boldsymbol{conf = 1 - dist(header, label)}$$

If the header and the class label both contain numeric parts (i.e. "Question 24"), then we compare the distance of the numeric and alpha parts separately and then combining with weights .75 and .25 respectively. This increases the likelihood of labeling alphanumeric question columns with the correct question number. If the confidence of the best candidate label for a column header is less than .45, the inferred header in the File Mappings is presented to the user as "Unknown, otherwise the inferred header is shown.

From initial beta testing, inference of column headers shows strong positive results. While being able to match columns in our synthetic data, we judge the performance of our model on the data which users have uploaded and completed the file mappings process. In order measure performance, we first frame true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) in our problem. If we infer a column header and the user verifies it as correct, it is counted as a TP. However, if the inferred header was verified as something different (inferred header is overridden), it is counted as a FP. This incorrect guess would normally be counted as a TN; however, while our task is to infer column headers, we also are tasked with excluding columns of extraneous data mingled in with student data. For this reason, if the inferred column header is "Unknown," and the user verifies the header as "Do Not Import," we count it as a TN since this column is confirmed to be unnecessary for analysis and visualization. If a column header is "Unknown," and the user verifies the column as actual student data, we count this as a FN. The results from the initial user testing are found in Table 2. Users 16,17, 19, and 20 have been highly active compared to other testers, but still show strong positive results. User 18 has shown to have a bad experience using our system to upload their files. Upon inspection of the raw headers stored, it seems that either the system picked the wrong row as the header or the file does not contain headers. Due to privacy concerns, we do not store the files in their original state. After they are mapped to student records, the original file cannot be reverse engineered making ground truth, verification, and debugging difficult for both the schema and column tag inference models without user input.

User	Number of Files	Number of Columns	TN	FN	TP	FP	Accuracy	Precision	Recall	F1
1	1	32	0	0	31	1	0.9688	0.9688	1.0000	0.9841
2	1	40	5	7	27	1	0.8000	0.9643	0.7941	0.8710
3	1	44	1	1	42	0	0.9773	1.0000	0.9767	0.9882
4	4	50	3	6	40	1	0.8600	0.9756	0.8696	0.9195
5	2	62	0	0	62	0	1.0000	1.0000	1.0000	1.0000
6	2	62	0	0	62	0	1.0000	1.0000	1.0000	1.0000
7	2	68	0	0	68	0	1.0000	1.0000	1.0000	1.0000
8	2	70	5	2	62	1	0.9571	0.9841	0.9688	0.9764
9	2	88	0	0	87	1	0.9886	0.9886	1.0000	0.9943
10	3	96	0	0	96	0	1.0000	1.0000	1.0000	1.0000
11	4	124	0	1	123	0	0.9919	1.0000	0.9919	0.9960
12	6	186	0	1	185	0	0.9946	1.0000	0.9946	0.9973
13	4	192	0	2	187	3	0.9740	0.9842	0.9894	0.9868
14	6	198	0	0	198	0	1.0000	1.0000	1.0000	1.0000
15	6	218	1	14	201	2	0.9266	0.9901	0.9349	0.9617
16	12	471	0	4	404	63	0.8577	0.8651	0.9902	0.9234
17	15	785	2	20	710	53	0.9070	0.9305	0.9726	0.9511
18	17	905	2	80	487	336	0.5403	0.5917	0.8589	0.7007
19	20	909	9	31	830	39	0.9230	0.9551	0.9640	0.9595
20	34	1632	0	0	1632	0	1.0000	1.0000	1.0000	1.0000

Table 2. Results from the column tagger for initial beta users.

Work in Progress: Information Visualization

The information visualization facility of the Data Explorer contains a variety of functions implemented using the D3.js JavaScript library by Bostock, Ogievetsky, and Heer (2011)[14]. Figure 3 shows how normalized (Hake) gain is plotted, with order statistics (mean and median) and standard deviation, for a class's performance on an assessment. Figure 4 shows how the visualization services also allow drill-down ("breakdown") by question, an important type of analytical query that results in the display of a distribution of answers for each question and facilitates comparative analytics for pre- and post-instructional assessments. The objective of these visualizations is to provide instructors with actionable insight concerning: topics covered; the impact of instruction and classwork on student learning as assessed formally using tests such as FCI, FMCE, and BEMA; and longitudinal trends of concern. In continuing work, we are exploring additional ways to drill down into multidimensional assessment data, such as using the *TableLens* visualization of Rao and Card (1994)[15].

Figure 3. Data visualizer component of the Data Explorer, displaying a histogram of normalized gain for a hypothetical class on the Force Concept Inventory (FCI) assessment.

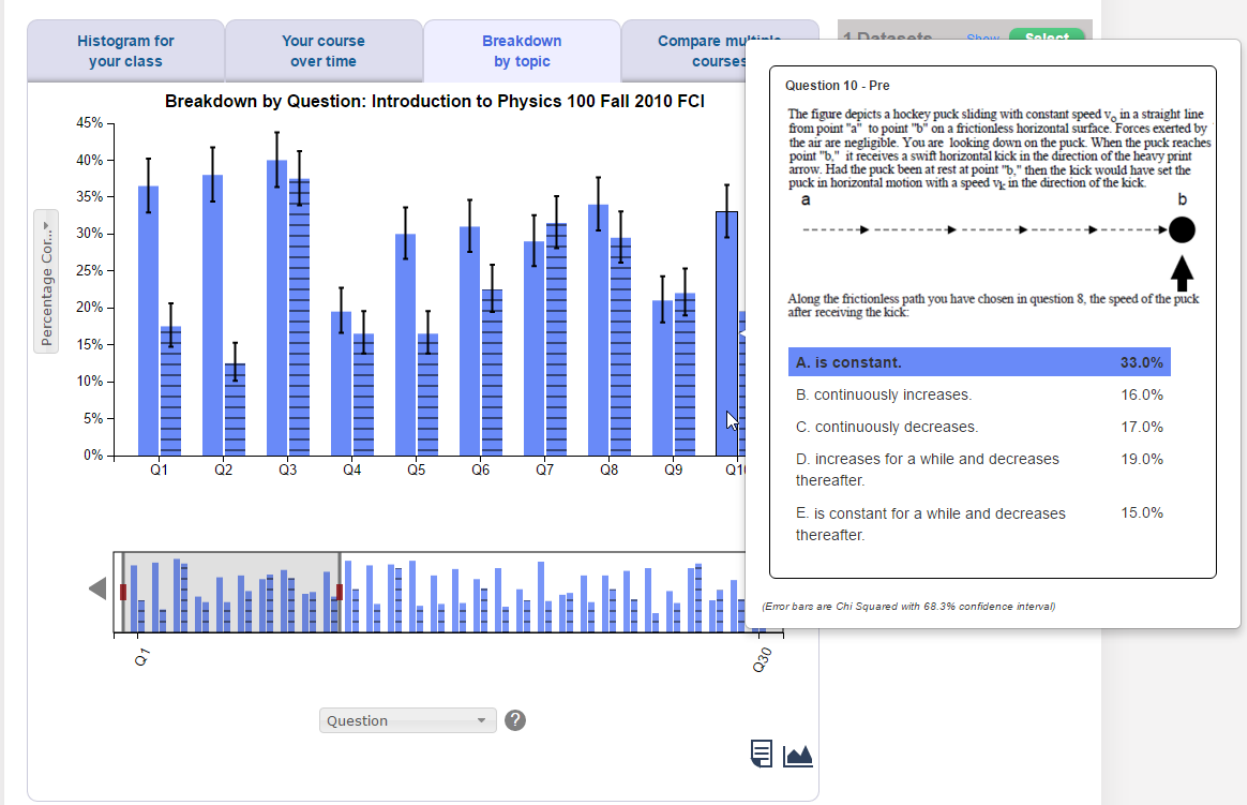

Figure 4. A "Breakdown by Question" view, showing drill-down for a single question and multiple-choice responses, together with the distribution of student responses, on a post-instructional assessment question (also for the FCI).

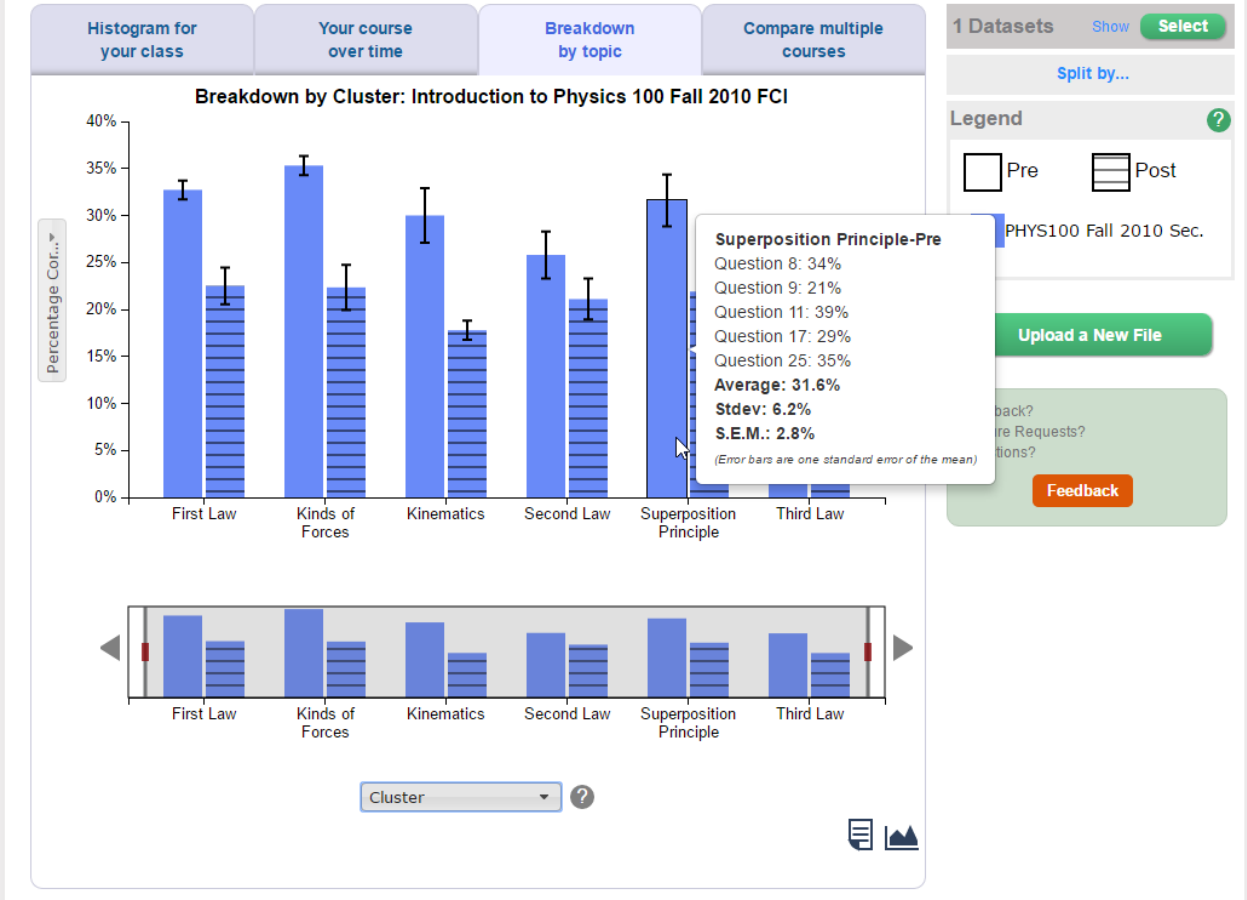

Figure 5. Visualization of student performance on pre- and post- assessment, organized by classification of question. Class labels are assigned by subject matter experts (physics education researchers).

Continuing Work: Information Retrieval and Data Mining

A further capability, designed to facilitate instructor exploration of assessment data, is that of grouping questions by known or discovered category. Figure 5 shows the results of visualizing hand-labeled categories (which are known as *classes* in machine learning, *clusters* in statistics, and *segments* in business analytics). Work in progress aims at using unsupervised learning to perform clustering of assessment questions (by topic modeling or by other similarity-based learning). The key capability that this future work aims at is that of retrieving *classes like mine* relative to longitudinal data (short time series) and similarity measures adapted to such time series. Meanwhile, clustering can also enable similarity-based queries for time series data as introduced by Rafiei and Mendelzon (1997)[16]. Our time series consist of student assessment scores and normalized gain measures, and thus admit the same kind of dimensionality reduction and indexing as developed by Keogh, Chakrabarti, Pazzani, and Mehrotra (2000)[17]. Ultimately, our goal is to develop a data-driven approach towards concept similarity in assessment data in STEM education, as Madhyastha and Hunt (2009)[18] were able to do to some degree for diagnostic assessments.

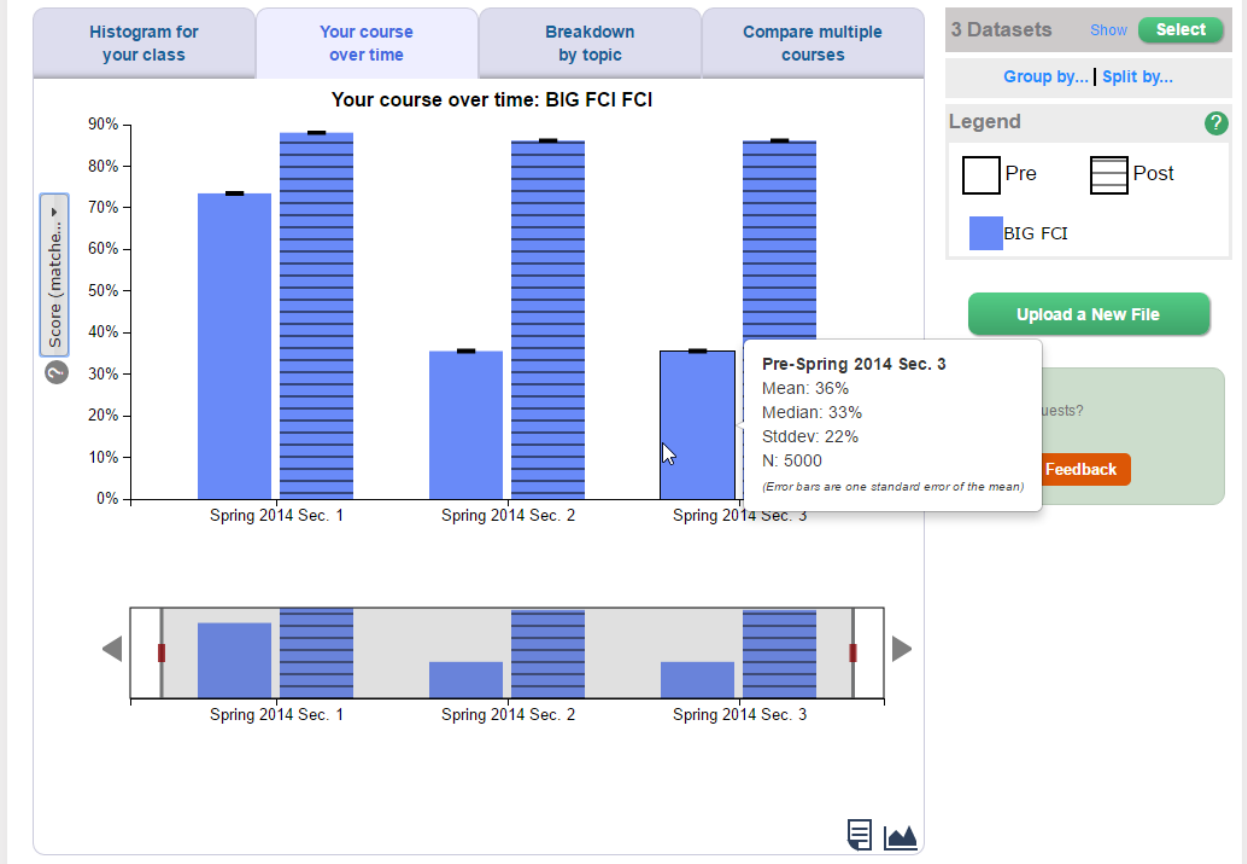

Figure 6. Visualization of courses over time: tracking performance across classes in multiple offerings (semesters and sections) in a longitudinal study.

Future Work: Instructional Decision Support and Adaptive Recommendation

Figure 6 includes a visualization of assessments across multiple courses taught at a single institution, typically by a single instructor under whose login the data are grouped for multiple semester combinations. The visualization subsystem also provides a facility for drilling down by section. This provides the analytical setting for one of our long-term objectives: to progress from interactive visualization within this federated display to adaptive decision support systems and tutoring systems.[19]

Conclusion

In this paper we have presented a data integration and information management system for STEM education research. The functionality outlined in the example screen captures is focused around our continuing research regarding schema inference and educational data mining from student assessments. The key novel contributions with respect to data integration are intelligent systems components for schema inference where columns and other elements are unlabeled, nonstandard, and may include missing data. The novel contribution with respect to analytics are the interactive information visualization components that both provide insights into assessment data and generate requirements for similarity-based retrieval and comparative evaluation of student performance.

Acknowledgments

This paper would not be possible without the work of our team, including Surya Teja Kallumadi, Jacob Ehrlich, Pavel Kuropatkin, and Josh Manning. This project was supported under NSF grant DUE-1347821, Collaborative Research, Community Implementation, WIDER: Data Explorer and Assessment Resources for Faculty.

References

1 Hestenes, David, and Halloun, Ibrahim. "Interpreting the force concept inventory." The Physics Teacher 33.8, 1995, pp 502-506.a

2 Thornton, Ronald K., and Sokoloff, David R. "Assessing student learning of Newton's laws: The force and motion conceptual evaluation and the evaluation of active learning laboratory and lecture curricula." American Journal of Physics 66.4, 1998, pp 338-352.

3 Ding, Lin, et al. "Evaluating an electricity and magnetism assessment tool: Brief electricity and magnetism assessment." Physical review special Topics-Physics education research 2.1, 2006.

4 Keininger, Thomas G., "Table structure recognition based on robust block segmentation." Proc. SPIE 3305, Document Recognition V, 22, 1998. doi:10.1117/12.304642.

5 Keininger, Thomas G., and Andreas Dengel, "Applying the T-Recs table recognition system to the business letter domain", Proceedings of the 6th International Conference on Document Analysis and Recognition (ICDAR 2001), 2001, pp. 518-522.

6 Doan, Anhai, Pedro Domingos, and Alon Halevy, "Learning to Match the Schemas of Data Sources: A Multistrategy Approach", Machine Learning, Springer, Berlin, 2003, Vol. 50 No. 3, pp. 279-301.

7 Cafarella, Michael J., Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang, "Uncovering the Relational Web", Proceedings of the 11th International Workshop on Web and Databases (WebDB 2008), 2008.

8 Venetis, Petros, Alon Halevy, Jayant Madhavan, Marius Paşca, et al., "Recovering Semantics of Tables on the Web", Proceedings of the 37th International Conference on Very Large Data Bases (VLDB 2011), 2011, pp. 528-538.

9 Nagy, George, Sharad Seth, Dongpu Jin, David W. Embley, et al., "Data Extraction from Web Tables: the Devil is in the Details", Proceedings of the 2011 International Conference on Document Analysis and Recognition (ICDAR), 2011, pp 242-246.

10 Fang, Jing, Prasenjit Mitra, Zhi Tang, and C. Lee Giles, "Table Header Detection and Classification", Proceedings of the National Conference on Artificial Intelligence (AAAI 2012), 2012, pp. 599-605.

11 Suchanek, Fabian and Gerhard Weikum, "Knowledge Harvesting in the Big-Data Era", Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2013), 2013, pp. 933-938.

12 Adelfio, Marco D. and Hanan Samet, "Schema Extraction for Tabular Data on the Web", 39th International Conference on Very Large Data Bases (VLDB 2013), 2013, pp. 421-432.

13 Zhang, Ziqi, "Towards Efficient and Effective Semantic Table Interpretation", The Semantic Web - ISWC 2014, 2014, pp. 487-502

14 Bostock, Michael, Vadim Ogievetsky, and Jeffrey Heer. "D^3: Data-Driven Documents". IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis), 2011. Vol. 17 No. 12, pp. 2301-2309.

15 Rao, Ramana & Stuart K. Card, "The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1994), 1994, pp. 318-322.

16 Rafiei, Davood, and Alberto Mendelzon. "Similarity-Based Queries for Time Series Data", In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 13-24.

17 Keogh, Eamonn, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra, “Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases”, Knowledge and Information Systems, 2001, Vol. 3, No. 3. DOI: 10.1007/PL00011669
18 Madhyastha Tara & Earl Hunt. "Mining Diagnostic Assessment Data for Concept Similarity", Journal of Educational Data Mining, 2009, Vol. 1, No. 1, pp. 72-91.
19 Brusilovsky, Peter, and Eva Millán, "User models for adaptive hypermedia and adaptive educational systems", The Adaptive Web: Methods and Strategies of Web Personalization, Springer, Berlin, 2007, pp 3-53.