Analyzing Game-Based Collaborative Problem Solving with Computational Psychometrics

Stephen T. Polyak, PhD  
ACTNext  
500 ACT Dr.  
Iowa City, Iowa 52240  
steve.polyak@act.org

Alina A. von Davier, PhD  
ACTNext  
500 ACT Dr.  
Iowa City, Iowa 52240  
Alina.vonDavier@act.org

Kurt Peterschmidt  
ACTNext  
500 ACT Dr.  
Iowa City, Iowa 52240  
Kurt.Peterschmidt@act.org

ABSTRACT

This paper describes a psychometrically-based approach to mining and classifying behavioral data both in real-time and in post-game analyses. The data were collected from a sample of middle school children who interacted with a game-like, online simulation of collaborative problem solving tasks. In this simulation, a user is required to collaborate with a virtual agent to solve a series of tasks within a first-person maze environment. The tasks were developed following the psychometric principles of Evidence Centered Design (ECD) and are aligned with the Holistic Framework developed by ACT. The analyses presented in this paper are an application of an emerging discipline called computational psychometrics which is growing out of traditional psychometrics and incorporates techniques from educational data mining, machine learning and other computer/cognitive science fields. In the real-time analysis, our aim was to start with limited knowledge of skill mastery and then demonstrate a form of continuous Bayesian evidence tracing that updates sub-skill level probabilities as new conversation flow event evidence is presented. This is performed using Bayes’ rule and conversation item conditional probability tables. The items are polytomous and each response option has been tagged with a skill at a performance level. In our post-game analysis, our goal was to discover unique gameplay profiles by performing a cluster analysis of user’s sub-skill performance scores based on their patterns of selected dialog responses.

KEYWORDS

psychometrics, problem-solving, collaboration, clustering, simulation, game, evidence-centered

1 INTRODUCTION

Collaborative problem solving (CPS) is widely considered as one of the critical skills for academic and career success in the 21st century [14]. The literature on this topic highlights changing trends that are leading to more employment opportunities that demand collaboration and interaction between people in problem-solving contexts [16, 30]. This has increased the need in the education industry to address ways to teach and assess these skills [49]. In this paper we consider the cognitive and social perspectives of the collaborative problem solving process and examine the circumstances under which collaborative problem solving might best take place to evaluate a participant’s level of competency. We outline a structure through which the contributing processes can be monitored and assessed in an electronic environment. In doing so, we reference an emerging discipline called computational psychometrics that is growing out of traditional psychometrics and incorporating techniques from educational data mining, machine learning and other computer/cognitive science fields. We also introduce our initial work on a collaborative problem solving simulation in which a user is required to collaborate with a virtual agent in order to solve a series of tasks/problems within a first-person maze environment. We demonstrate two techniques based on our knowledge of computational psychometrics:

- real-time Bayesian evidence tracing that updates sub-skill level probabilities as new evidence is presented
- a post-game clustering analysis of a user’s sub-skill performance scores aimed at defining different profiles of simulation results

2 MATERIAL AND METHODS

In this section we share our study approach, starting with the identification and selection of the specific CPS sub-skills we monitored. We then describe our simulation/game design, task development and conversation tree construction. Given these constructs, we detail our methods for computational psychometric evidence tagging and continuous tracing. We overview the steps in study execution and data collection. Finally, we define our postgame analysis process that utilizes a set of machine-learning based clustering techniques.

2.1 CPS Sub-skills

For this study, our methodology was to first select a set of collaborative problem solving sub-skills that have been researched and published as part of ACT’s investigations into helping people achieve education and workplace success. In “Beyond Academics: A Holistic Framework for Enhancing Education and Workplace
Success* Camara et al.[7] identified facets beyond the well known core academic skills which include the domain-specific knowledge and skills necessary to perform essential tasks in the core content areas of English language arts, mathematics, and science. These additional areas include:

- Cross-cutting capabilities: General knowledge and skills necessary to perform essential tasks across academic content areas. This includes technology and information literacy, collaborative problem solving, thinking and metacognition, and studying and learning.
- Behavioral skills: The interpersonal, self-regulatory, and task-related behaviors important for adaptation to, and successful performance in, education and workplace settings.
- Education and career navigation skills: The personal characteristics, processes, and knowledge that influence individuals as they navigate their educational and career paths (e.g., make informed, personally relevant decisions; develop actionable, achievable plans).

As seen above, the cross-cutting capabilities section of the Holistic Framework includes collaborative problem solving as part of a broad, four category enumeration:

1. Technology and Information Literacy
2. Collaborative Problem Solving
3. Thinking and Metacognition
4. Studying and Learning

Within the framework, CPS skills are further decomposed into various sub-skills and sub-skill areas. For example, sub-skill areas within CPS include:

- Behavior
- Collaborative Communication
- Problem Analysis
- Solution Planning
- Extended Collaboration (Teamwork)

For this study, we selected 5 sub-skills to gather and analyze for CPS evidence:

- Feature Identification (FI): Identifies the key features of the problem space
- Maintaining a Shared Understanding (MU): Identifying and reconciling gaps in understanding
- Engagement/Interaction (EN): Engagement in the group process and the degree to which that engagement is self-initiated
- Strategy (S): Evidence of establishing a plan of action or policy designed to achieve a major or overall aim
- Evaluate (EV): Recognizing own strengths and weaknesses in relation to others

2.1.1 CPS Assessments. Educators are requesting CPS assessments that reflect the way people actually teach, learn and work. There is a renewed interest in performance assessments and efforts are being made to develop these complex assessments in virtual settings. There are several examples within the educational assessment industry that have established a particular approach towards measuring CPS skills. These include:

- The Programme for International Student Assessment (PISA) 2015 administered a test of collaborative skills [34]
- The National Assessment of Educational Progress (NAEP) hosted a symposium on Collaborative Problem Solving (CPS) in September, 2014, and as a follow-up, the National Center for Educational Statistics (NCES) commissioned a white paper on the considerations for introduction of the CPS in NAEP [31]
- The College Board’s Advanced Placement (AP) Computer Science Assessment is being redesigned to include collaborative tasks
- Educational Testing Service (ETS) and the Army Research Institute co-hosted a working meeting, Innovative Assessment of Collaboration, November 3-4, 2014, and an edited interdisciplinary volume based on that meeting is being published with Springer Verlag [49]
- The Smarter Balance Consortium developed an assessment system where performance tasks, including collaborative tasks, are considered for being administered to students as a preparatory experience and then are followed with individual assessment [10]

These selected sub-skills and similar capabilities are important for success but are rarely given the attention they deserve, particularly when it comes to assessment and grading. Because CPS is largely enacted as an interactive set of tasks, we need a means to provide a multi-agent setting in which the subject under assessment can express their abilities. This means providing support for discussion, negotiation, decision making, etc. with another participant, be they a human or simulated agent. In either case, all of these interactive data are referred to as "process data" that offer insight into the interactional dynamics of team members; they are relevant for defining collaborative tasks and for evaluating the results of the collaboration. Traditionally these data were not available to researchers at a scale that would allow for meaningful inferences. With the advances in technology, these complex data can be captured in computerized or in virtual settings.

The data from collaborative tasks can consist of time-stamped sequences of events registered in a log file. From a statistical perspective, these activity logs or log files are detailed time series describing the actions and interactions of the users. Hao et al.[15] discuss and describe the log files for the collaborative assessments. In addition to the process data, if the collaboration is set up in a cognitive (say, science) task it will also result in outcome data. These types of data are more similar to the outcome data from the traditional tests and indicate if a particular question was answered correctly, whether the problem was solved (or to what degree it was solved). Another feature of the data from collaborative tasks is that they can be at different units of measurement: they can be characterized as individual and team data both as outcome data and process data.

Attempting to measure collaboration using a game or other virtual environment is not novel. Neither are the ideas of stealth assessment [39] or evidence centered assessment design [28, 39]. However, it is still common to see measurement of collaboration provided by post hoc survey data collection [37, 43]. Measuring through in game data collection techniques holds value, in that more
computational psychometrics can be made and some of the disadvantages of self-reports [33] can be avoided, such as self-presentation [35].

2.2 Simulation/Game Design

In order to collect data and test hypotheses for this study, ACT developed a CPS game called “Circuit Runner” which allows subjects to play online, in a web browser, with the mission to solve a series of challenges in order to “win” the game. The catch is that the player needs to collaborate with an automated, virtual agent that has information required to complete the challenges.

Figure 1: Circuit Runner: A CPS Dialog Panel Game Screen-shot

In total there are five distinct challenges that range from an agent/player feature discussion around a coded door lock panel to a more sophisticated challenge that involves collaborative discovery of a sequence of power transfer steps in order to succeed. The player navigates from challenge to challenge via a 3-D maze in a first person perspective and is also given continuous access to the agent via a dialog panel which can present prompts and dialog responses from various dialog/conversation trees the player may select. A view of the conversation panel within the game is provided in Figure 1. All of the dialog response selections made by the player are recorded in a game “conversation flow” log data file. We can think of the presentation of conversation prompts via the agent to be analogous to the presentation of item prompts in a more conventional assessment. The selection of conversation choices by the participant result in item responses captured during the game. Additional telemetry data is gathered including clicks, keystrokes, distance travelled, challenge duration, and dialog selection timing.

2.3 Computational Psychometrics

Given these constructs for assessing CPS skills, we consider our methodological basis applying computational psychometrics [47].

Computational psychometrics is defined as a blend of data-driven computer science methods (machine learning and data mining, in particular), stochastic theory, and theory-driven psychometrics in order to measure latent abilities in real-time [25].

This blend can be instantiated as iterative and adaptive hierarchical inference data algorithms embedded in a theoretical psychometric framework. This hierarchical approach to multimodal data has been discussed in Khan et al.[20, 21]. The idea is that the approach to test development and data analysis is rooted in theory and starts with the implementation of the Evidence Centered Design (ECD) [29] principles, then the test is piloted and the multimodal metadata (fine grain data points) are collected along with the data from traditional information. This approach is also known as a top-down approach because it relies on the input and theories devised by human experts. The next step involves a bottom-up approach, in which the data are analyzed with unsupervised and supervised algorithms from data mining and machine learning. If new relevant patterns are identified (for example, performance pattern, behavior patterns), these will be considered for being incorporated in the revised psychometric models. The data mining and machine learning approaches applied to metadata will result in midlevel representations of the constructs that can be further incorporated in the original psychometric models along with the data from the traditional items. Next, the psychometric models are updated and the process is repeated with a second large-sample pilot data set. At this stage one may consider models for stochastic processes if the data allow. Once the psychometric model is stable, only then the assessment is administered to the population of interest. On the operational data, only supervised machine learning algorithms and already defined and validated psychometric models are further used in order to achieve a stable measurement and classification rules.

This computational psychometrics (CP) model we described can be instantiated using ECD in the CP context. This framework involves designing the system (learning and/or assessment) based on theory, identifying constructs associated with the competency of interest, and finding evidence for these constructs from low-level, multiple sensor data [5]. At this stage, one defines the constructs to be measured and develops the tasks that are collaborative so that they provide the “right” data to support the intended claims and choice of psychometric models. The CPS task may also be embedded into traditional assessments to increase the measurement accuracy for the cognitive construct and to enhance the data around the CPS task [25].

The types of psychometrics models associated with complex data with dependencies have been IRT-like models and Bayesian Belief Networks (BBN) [27] [23]. BBNs model the probability that a student has mastered a specific knowledge component conditional on the sequence of responses given to previous elements of a task, and eventually, to other tasks whether they are associated with that knowledge component or not (as long as they are part of the network and share at least an indirect connection. BBNs have long been applied in simulations and games to represent student knowledge and, thereby, guide the activities of the tutoring system [9] [12] [46]. BBNs have also played a central role in the design of complex assessments [38] therefore, they are an obvious
methodological bridge between measuring CPS skills and traditional psychometric theory. However, the practical implementation of BBNs often requires highly simplified assumptions and, as with traditional models, they have not been adapted to represent the knowledge of multiple individuals simultaneously.

Fortunately, there are other models such as stochastic point processes that have been used extensively in economics that can aid the modeling of interdependencies based on the temporal structure of the collaborative interactions [11], hidden Markov models [41], and models rooted in the cognitive or social theories such as Agent-based modeling, ACT-R [6] and Markov Decision Processes, which is a cognitive model with separable components (goals/motivation, beliefs about the world, ability to optimize behavior) and which defines behavior as an optimization of expected rewards based on current beliefs about the world [22].

The application of data mining (DM) and machine learning (ML) tools is also present in the CP context. Data mining does not have a long history in education or psychology because, until recently, educational and psychological data were not often of high enough dimensionality to require such techniques. The purpose of data mining techniques is to reduce the dimensionality of the dataset by extracting interpretable patterns to allow research questions to be addressed that would otherwise be feasible [36]. Different clusters (patterns of responses) may be assigned different scores. The tools known as visualization, clustering, classification, feature extraction, sequence clustering, and editing distance methods for scoring are examples of DM tools that may be applied. There are already promising results regarding the identification of new evidence to feedback into the ECD/test development cycle and generation of testable hypotheses [50] [17] [18].

Machine learning algorithms may be used for in-vivo adaptive “learning” and assessment, by using the results/features from a DM classifier (often in conjunction with a psychometric model in a Bayesian framework) that “learns” from the data to predict the success on a task. ML algorithms rely on the availability of large and representative training data sets. These algorithms have been used in education for the development of automated scoring of essays; now we are using similar algorithms for the automatic scoring of speech and chat in collaborative interactions and for the automated detection of affective states during the collaboration [20] [48]. In specific practical applications, this hierarchical iterative framework may be implemented in simplified or less explicit forms; for example, some of the steps may be there but may not be explicitly described and some of the steps may not be needed.

2.3.1 Skill Evidence Tagging. For the cCircuit Runner game, ACT holistic framework researchers designed the tasks, and the potential conversation flows, so that they would require participants to collaborate with the virtual agent in a way that would provide evidence of their latent skill ability associated with our selected CPS sub-skills. Most of the dialog tree responses were tagged with one or more sub-skills that were expert judged to provide skill evidence. Furthermore, this evidence was also refined into a level tag using a 3 level enumeration of High, Med, and Low. In Figure 2 we illustrate this tagging for one item/dialog tree prompt:

![Figure 2: CPS Response Coding](image)

"I am in front of a computer monitor. I have access to the teacher, a map of the maze, and something called an ASCII lookup table. The teacher is talking to me."

and a selected dialog response of:

"What is the teacher saying?"

This participant event/action presents evidence of CPS skills:

- Monitoring Understanding (MU) at the Med (2) level (MU.2)
- Engagement (EN) at a High (3.x) level (EN.3.4)

These items are polytomous and can effectively be scored for a participant based on their sub-skill association and level identification.

2.3.2 Bayesian Evidence Tracing. We can see that conversation flow between the participant and agent provides us with a continuous stream of evidence of a participant’s CPS sub-skill, our research question was:

"Given the real-time, sequential evidence presented via the data of dialog response selections in this game, can we intelligently predict the performance level at each sub-skill?"

The methodology we chose to follow to answer this question used a Bayesian approach related to those typically found in intelligent tutoring systems, such as Bayesian Knowledge Tracing (BKT) [9]. The steps to demonstrate this were as follows:

- Extract raw conversation flow game log from a set of played games
- Transform the conversation flow into a flattened file that combines prompt and response and filter out any potential test data
- Generate a 1-Hot encoding of evidence (discussed below)
- Compute Bayesian predictions for all five sub-skills, across each performance level
- Plot the evidence tracing for insight/analysis

Extract. The log data file that is extracted from the game is outlined in Table 1. Each user can have 1 or more sessions and each session can have 1 or more games. In practice though we are typically only interested in 1 game for a single user. As we can see, the log collects the presentation of a dialog tree prompt to the user in a game as row type ‘P’. The prompt presented is recorded in the column ‘prompt_id’. Row type ‘R’ records the response selected by
the user in the game for the prompt row immediately preceding it in the log. This raw game log file contained the game session log for several game instances.

**Transform.** Our next step was to flatten this representation so that the prompt and the response rows were combined into a single record as shown in Table 2. Additionally, we also filtered out data rows that were known to be developer gameplay 'user_ids' so that we were only looking at data from actual subject participants. There were also prompt rows followed by some in game action. So instead of a response to that prompt, the user had done something that subsequently caused another prompt to appear. Since there was no response to that initial prompt, it, along with the following action, were also filtered out. Ultimately, \( N = 159 \) unique games for this analysis.

**1-Hot.** Taking the flattened prompt/response data, we encoded each game as a single row in a 159x286 matrix outlined in Table 3. The number of rows is the \( N \) count and the number of columns are the three identifiers (session, user, game) plus the 283 potential, selectable dialog responses (D=283). We encoded a '1' if the user selected the identified response at any time during the game. It should be noted that several of the dialog sub-trees can allow a user to loop back through the tree within a single game. If the user selected a particular response more than once in a game we still recorded the selection with a single '1'. Otherwise, if the user never selected a particular response during the game the encoding for that column is '0'.

**Compute.** Before we introduce our computation of probabilities for the performance levels of a game’s CPS sub-skills, let’s first review Bayes’ theorem and how its application will allow us to trace the evidence over time.

**Bayes’ Theorem.** One way to think of Bayes’ theorem [4] is that it gives us a way to update the probability of a hypothesis, \( H \), in light of some body of evidence, \( E \). This way of thinking about Bayes’ theorem is called the diachronic interpretation. More precisely, the probability of the hypotheses changes over time as we see new evidence. Rewriting Bayes’ theorem with \( H \) and \( E \) yields

\[
p(H|E) = \frac{p(E|H)p(H)}{p(E)}
\]

In this interpretation, each term has a name:

- \( p(H) \) is the probability of the hypothesis before we see the evidence, called the prior probability, or just ‘prior’.
- \( p(H|E) \) is what we want to compute, the probability of the hypothesis after we see the evidence, called the ‘posterior.’
- \( p(E|H) \) is the probability of the evidence under the hypothesis, called the likelihood.
- \( p(E) \) is the probability of the evidence under any hypothesis, called the normalizing constant.

As an example, let’s consider an application of the Bayes’ Theorem to a simple selection task using two bins to select from. On the performance of this task, we will consider the evidence (E) from a selection event and attempt to compute the probability of two competing hypotheses \( (H_1) \) and \( (H_2) \). Hypothesis 1 will consider that the selection event happened using bin 1 and hypothesis 2 will consider that the event used bin 2. In Figure 3 we depict the two bins, bin #1 and bin #2. Bin #1 contains 10 blue widgets (B) and 30 red widgets (R). Bin #2 contains 20 blue widgets (B) and 20 red widgets (R). Let’s say that a selection event occurs and the evidence is that of a red widget (R). We will now apply the Bayes’ theorem to consider the probability associated with each hypothesis:

\( (1) \) \( H_1 \): The red widget came from bin #1

\( (2) \) \( H_2 \): The red widget came from bin #2

The prior for both \( p(H_1) \) and \( p(H_2) \) are the same, \( \frac{1}{2} \), because we are assuming that red and blue widgets appear equally in each bin. The likelihoods are different though, as we can see based on the composition of the bins. Specifically we have,

\[
p(E|H_1) = \frac{3}{4}
\]

\[
p(E|H_2) = \frac{1}{2}
\]

Putting this all together we can compute the posterior for both hypotheses as:

\[
p(H_1|E) = \frac{\frac{1}{2} \times \frac{3}{4}}{\left(\frac{1}{2} \times \frac{3}{4}\right) + \left(\frac{1}{2} \times \frac{1}{2}\right)} = 0.6
\]
We can then state that given the evidence of a red widget we believe there is a 60% chance this was associated with bin #1 and a 40% chance this was associated with bin #2.

Response to Skill. Given this computation, we can apply it to the evidence and hypotheses we have for the CPS game. In our selection example, the evidence was straight-forward: was the widget blue or red? In the CPS game we need a lookup table for our response to determine which CPS sub-skill and at which performance level the response selection evidence is associated with. In Table 4 we list what our lookup table contains. As row 1 indicates, given a prompt '0.1' and a response '1' we can see that this has been tagged by an ACT content expert as providing evidence of:

- Engagement (EN) at a high level (3) (and specifically explanation #4 in that high level)
- Finding Information (FI) at a med level (2) (and specifically explanation #2 in that med level)
- Monitoring Understanding (MU) at a med level (2)

As Mislevy et al. [27] describe in their application of ECD to interpreting game log data, we can refer to these sub-skills as latent variables, student model variables (SMVs) or competencies/proficiencies and will denote them using \( \theta \). “[the authors] posit that students’ performances, characterized by features \( x_j \), arise from some underlying dimensions of knowledge, skill, familiarity, preferences, strategy availabilities, or whatever way we want to characterize them for the purposes at hand. These are called latent variables in the psychometric literature, and student model variables (SMVs), or sometimes competencies or proficiencies, in ECD terminology. We will denote them by \( \theta \)”.

Figure 4 presents a directed graph representation of a multivariate model with parameters that specify conditional distributions of \( x_j \) (an instance of a selected CPS dialog response) given \( \theta \). The \( \beta \) parameters can represent the "nature and strengths of the relationship" between an \( x_j \) and the associated latent variable \( \theta \). In this way we can express the relationship between latent variables in our model and the dialog selection evidence using conditional probability tables (CPT) [27].

### Conditional Probability Tables

In our Bayesian example, the \( p(H|E) \), or likelihood, was a function of the composition of the bins. In our application of Bayes’ rule to the game prediction we will use a conditional probability table for our likelihood term instead. An example of a CPT is shown in Table 5. This table was built to provide a modest weighting that indicates a slightly higher likelihood that users will pick responses aligned with their latent variable. Using this table we can explicitly model the type of evidence (high/medium/low performance level, designated by research tagging) which is along the row and the hypothesized performance level of the latent variable (low/medium/high) along the column. Said another way, this table illustrates that if a participant’s latent variable is low (row 1) then there is a slightly higher likelihood (0.4) that they will select a low tagged response option instead of a medium/med or high level (0.3). In practice, there could be a unique CPT created for each item/conversation prompt instance. These

\[
p(H_2|E) = \frac{\frac{1}{2} \times \frac{1}{2}}{\left(\frac{1}{2} \times \frac{3}{4}\right) + \left(\frac{1}{2} \times \frac{1}{2}\right)} = 0.4
\]  

(5)

Figure 3: Bayesian Selection Example

Figure 4: Conditional distribution of \( x_s \) given \( \theta \)

<table>
<thead>
<tr>
<th>( \theta )</th>
<th>( X_1 )</th>
<th>low</th>
<th>med</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta_{low} )</td>
<td>0.4</td>
<td>0.3</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>( \theta_{med} )</td>
<td>0.3</td>
<td>0.4</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>( \theta_{high} )</td>
<td>0.3</td>
<td>0.3</td>
<td>0.4</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Conditional Probability Table (CPT)
unique CPTs might be derived empirically through statistical analysis or could be built using expert judgement. This would allow researchers to fine tune the likelihoods based on the particular item content/difficulty.

Evidence Tracing. In our Bayesian widget selection example, we presented two possible hypotheses: either the widget came from bin 1 or 2. For the CPS game, we are presented with a response that indicates sub-skill ($ss_i$) evidence at a particular performance level. As we trace a student’s selections we are maintaining three possible hypotheses about the participants latent variable per each sub-skill, viz.

1. Hypothesis: $\theta_{ss_i}^{high}$, Given the evidence to date, the player has a high level for this sub-skill
2. Hypothesis: $\theta_{ss_i}^{med}$, Given the evidence to date, the player has a medium level for this sub-skill
3. Hypothesis: $\theta_{ss_i}^{low}$, Given the evidence to date, the player has a low level for this sub-skill

For each game ($G$=game_id) then, our algorithm for computing probabilities for the performance levels of a particular sub-skill $ss_i$ is presented in Figure 5.

In the initialize step, we set the prior for all hypotheses about a student’s sub-skill level at $\frac{1}{3}$, since we have no other evidence. For each dialog response, if it was tagged for the sub-skill then we will recompute the posterior for each hypothesis by incorporating the new evidence. The $\beta$ value used for the likelihood will be based on a CPT lookup that considers which table is being used for which dialog/response pairing and also what level the skill was tagged with. In our initial application we used the same CPT for all evidence (Table 5) but in our future work we intend to work with the dialog content authors to fine tune the application of CPTs based on a more refined judgement of distributions. We demonstrate the results of our tracing in section 3.1.

2.4 Study Execution

We recruited a total of 159 middle school children to play the game. The game was accompanied with a research survey containing personality and background questions. The survey data included age, gender, grades, technology use, and personality facets. On average, the participants spent around 30 minutes playing the game. We are currently performing a second run of the study that recruits up to 1000 participants using Amazon Mechanical Turk. In that run we are also including a few more instruments in addition to the game play:

- a pre-survey and post-survey (demographics, background questions)
- a collaborative problem solving questionnaire
- a situational judgement task assessment involving workplace behaviors relating to collaboration and problem-solving
- a HEXACO personality assessment. HEXACO is a six-factor structure of personality-descriptive adjectives. [2]

2.5 Postgame Analysis

In the postgame analysis, we extracted the raw conversation flow logs from the game and transformed the data to align with the skill/level tagging data provided by the ACT holistic framework researchers. We then used these data to address the following research question:

“Given the raw data of selected dialog responses across various games played, can we intelligently group patterns of selections into clusters that may represent different classifications of CPS skill evidence?”

Mislevy et al. [27] demonstrated how traditional assessment approaches relate to emerging techniques for synthesizing the evidence we outlined in our research question. In particular they demonstrate how the models/methods of psychometrics can be leveraged in game-based assessments to collect evidence about aspects of a game player’s activities and capabilities.

“Exploratory data analysis (particularly visualization and hypothesis generation tools) and educational data mining techniques (including recent methods such as unsupervised neural network modeling and ... cluster analysis, latent class analysis, and multidimensional scaling) can identify associations among observable features of performance that suggest new student-model variables ... Educational data mining is the process of extracting patterns from large data sets to provide insights into instructional practices and student learning. It can often be employed for exactly the tasks of evidence identification: feature extraction based on patterns in data ...

Bauckhage et. al. also discussed the concept and challenges stemming from our research question with respect to clustering game behavior data. [3]

“the proliferation of behavioral data poses the problem of how to derive insights therefrom. Behavioral data sets can be large, time-dependent and high-dimensional. Clustering offers a way to explore such data and to discover patterns that can reduce the overall complexity...
of the data. Clustering and other techniques for player profiling and play style analysis have, therefore, become popular in the nascent field of game analytics. However, the proper use of clustering techniques requires expertise and an understanding of games is essential to evaluate results.

Based on this and other related research [8, 19, 24, 32, 40], it was evident that a machine learning-based, clustering methodology would be useful to explore patterns within our game dialog selection data. In particular we demonstrate an application of game-related, k-means clustering (as reported in other related research [45]) versus other types reported such as Linear Discriminant Analysis (LDA) [13] or Mixture Model clustering [44].

2.5.1 Extract. The log data file that is extracted from the game is outlined in Table 1. As we can see, the log collects the presentation of a dialog tree prompt to the user in a game as row type 'P'. The prompt presented is recorded in the column 'prompt_id'. Row type 'R' records the response selected by the user in the game for the prompt row immediately preceding it in the log. This raw game log file contained the game session log for several game instances.

2.5.2 Transform. As we mentioned in our Bayesian workflow, our next step was to flatten this representation so that the prompt and the response rows were combined into a single record as shown in Table 2. Additionally, we also filtered out data rows that were from known test 'user_ids' so that we were only looking at data from actual subjects. There were also prompt rows followed by some in game action. So instead of a response to that prompt, the user had done something that subsequently caused another prompt to appear. Since there was no response to that initial prompt, it, along with the following action, were also filtered out. The N count for this analysis was 159 unique games.

2.5.3 k-means Methodology. The methodology we followed involved these steps:
- Extract raw conversation flow game log from a set of played games
- Transform the conversation flow into a flattened file that combines prompts and responses, and filter out any potential developer gameplay data
- Encode each game as a single row in a 1-Hot encoding of selected dialog responses
- Translate the 1-Hot encoding into 5 datasets corresponding to evidence acquired on all 5 CPS domains
- Perform basic scoring of each game on the 5 CPS domains
- Perform k-means clustering [42] of game domain scores
- Present summary and results of clustering

2.5.4 Encode/Translate. Taking the flattened prompt/response data we encoded each game as a single row in a 159x286 matrix outlined in Table 3. The number of rows is the N count and the number of columns are the 3 identifiers (session, user, game) plus the 283 potential, selectable dialog responses (D=283). We encoded a '1' if the user selected the identified response at any time during the game. It should be noted that several of the dialog sub-trees can allow a user to loop back through the tree within a single game. If the user selected a particular response more than once in a game we still recorded the selection with a single '1'. Otherwise, if the user never selected a particular response during the game the encoding for that column was '0'. Each of the unique dialog prompt/response combinations were coded based on the 5 domains as defined in the CPS game data section.

Given this mapping, we were able to create 5 domain evidence matrix variations on the 1-Hot matrix where we substituted the k-means++ assignment of the K=8 clusters we noticed similar patterns between several of the clusters. In particular, there appeared to evidence acquired on all 5 CPS domains as defined in the CPS game data section.

2.5.5 Score. Given the 5 domain evidence matrices (as a variation from the 1-Hot encoding) we could then score a game on each of the 5 domains by a simple summing of evidence across each response feature.

\[
\text{score}^{FI} = \sum_{d=1}^{D} x_d^{FI} \\
\text{score}^{MU} = \sum_{d=1}^{D} x_d^{MU} \\
\text{score}^{EN} = \sum_{d=1}^{D} x_d^{EN} \\
\text{score}^{EV} = \sum_{d=1}^{D} x_d^{EV} \\
\text{score}^{S} = \sum_{d=1}^{D} x_d^{S}
\]

We then reformatted the scores into a domain score matrix 159x8 where the rows=N and the columns were the 3 identifiers (session, user, game) plus the 5 summed evidence score for each domain as show in Table 6.

2.5.6 Cluster. Using this derived score matrix we then performed an unsupervised learning k-means clustering of the data using the Graphlab-Create library\(^3\). We selected the K value based on the following heuristic: \( K = \sqrt{N}/2.0 = 8 \) clusters.

2.5.7 K Exploration. Starting with K=8 based on the heuristic value, we continued to evaluate additional potential K value assignments. The k-means implementation of Graphlab-Create uses the k-means++ algorithm for initial choice of cluster centers. This results in some randomization and variance of cluster assignment with each building of the model. As we visualized the data points with the assignment of the K=8 clusters we noticed similar patterns between several of the clusters. In particular, there appeared to be overlap between 4 sets of 2. This indicated that a 4 cluster assignment may be more appropriate.

We decided to build the model numerous times with a K value of 8 and compare cluster assignments between these model building runs. We saw that row assignment from the initial cluster assignment didn’t always result in classification to the same cluster as on a subsequent build of the model. Sorting the data on the first model build and looking at the cluster classification across the next two builds of the model, we saw some of the same assignments. We subsequently chose K = 6 and performed the same multiple run build of the model. Drift was somewhat less, but not significantly

\(^3\)https://turi.com/products/create/
so. Setting $K = 4$ and building the model several times showed much less variance in cluster assignment. There was still some drift, but it was significantly less than what we saw with a $K = 8$ and in general cluster assignments persisted across multiple builds of the model even with randomly chosen initial centers.

2.5.8 K-NN Query by Game Id. In addition to the k-means model, we also built a K-Nearest Neighbor (K-NN) model [1] using Graphlab-Create which allows us to go back and query the data for games that were similar to a selected game id using a cosine similarity distance metric.

2.5.9 Mixture Model Methodology. There are drawbacks to using the k-means clustering algorithm:

- assumes a specific shape of cluster distributions (spherically symmetric)
- only provides hard assignments to one of the possible clusters

k-means can be understood as a specific instance of a more generic approach to clustering that is defined by analyzing a mixture of distributions that can be computed using an Expectation Maximization (EM) algorithm [26]. Following the same methodology we outlined above to derive our data frame of CPS dialog we are tracking each level (high/medium/low) as a separate, but restricted to only looking at the final probabilities (posterior values), ultimately finishing with a ‘medium’ level assessment. If we were looking at the total dialog response time we initially used matplotlib to render similar views as can be seen in Figure 6. This graph shows the increases and decreases of the probability estimates for a participant’s EN sub-skill over time. There are three lines because we are tracking each level (high/medium/low) as a separate, but linked variable. All three variables begin using a prior set at .333 and then diverge as the evidence is traced using Bayesian analysis. Additionally we used Tableau to render similar views as can be seen in Figure 7. This view allows an analyst to see the predictions of performance levels for each skill, over time, for a single game. The blue area represents a high level, the white area is medium level, and the orange area is the probability of a low level. This view uses an area of fill representation.

The blue area represents a high level, the white area is medium level, and the orange area is the probability of a low level. This view uses an area of fill representation.

3 RESULTS

In the results section, we present visualizations of real-time Bayesian evidence tracing based on a participant’s continuous log evidence. We also present the results from our clustering data along with views of cluster data indicators and distributions.

3.1 Bayesian Evidence Tracing Results

Our implementation of the Bayesian algorithm described in Figure 5 was done in Python using a Jupyter notebook web application. We also used the SFrame API from Graphlab-Create to manipulate the game log data. In order to visualize the sub-skill probabilities over time we initially used matplotlib. An example of the plot for a sample game_id=114 can be seen in Figure 6. This graph shows the increases and decreases of the probability estimates for a participant’s EN sub-skill over time. There are three lines because we are tracking each level (high/medium/low) as a separate, but linked variable. All three variables begin using a prior set at .333 and then diverge as the evidence is traced using Bayesian analysis. Additionally we used Tableau to render similar views as can be seen in Figure 7. This view allows an analyst to see the predictions of performance levels for each skill, over time, for a single game. The blue area represents a high level, the white area is medium level, and the orange area is the probability of a low level. This view uses an area of fill representation.

Looking at the evidence collected for the single game_id=114 (Fig. 6), we can see the sub-skills for monitoring understanding (MU) and feature identification (FI) quickly settled on a ‘medium’ level assessment during the first third of the total dialog response interactions. In contrast, the strategy (S) and evaluate (EV) sub-skills settled on a ‘low’ level assessment over the final two thirds of the interactions. The engagement (EN) scores showed fairly dramatic swings between all three performance levels over time, ultimately finishing with a ‘medium’ level assessment. If we were restricted to only looking at the final probabilities (posterior values),

![Table 6: Scores Matrix](http://www.tableau.com)

<table>
<thead>
<tr>
<th>session_id</th>
<th>user_id</th>
<th>game_id</th>
<th>FL_Score</th>
<th>MU_Score</th>
<th>EN_Score</th>
<th>EV_Score</th>
<th>S_Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td>33</td>
<td>211</td>
<td>47</td>
<td>10</td>
<td>26</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>57</td>
<td>38</td>
<td>310</td>
<td>39</td>
<td>21</td>
<td>31</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Scores Matrix

The mean and covariance of the distribution: $N(x_i|\hat{\mu}_k, \hat{\Sigma}_k)$ divided by the normalizing constant which considers the probability over all possible clusters $\sum_{j=1}^{K} \hat{\pi}_N(x_i|\hat{\mu}_j, \hat{\Sigma}_j)$. We implemented the code for both the E-step and M-step in Python and ran the implementation over 120 iterations using the MU, FI and EN scores. The S and EV domains were excluded based on their low information content. We also implemented a matplotlib function to plot the computed responsibilities after a specified number of iterations in order to show how the clustering evolved over time. We present those plots in the clustering results section.
we wouldn’t have been able to notice these real-time patterns in gameplay. Since the Bayesian Evidence Tracing algorithm is an ‘anytime algorithm’, we are able to directly interrogate this model at any point to determine the current estimate of a user’s sub-skill probability.

3.2 Clustering Results

As we described in our methods section, we implemented two clustering approaches, a hard clustering assignment with k-means and a soft clustering assignment using a Gaussian mixture model approach. Additionally, we implemented a K-nearest neighbor (K-NN) mechanism to lookup related games based on the clustering data.

3.2.1 k-means/K-NN Results. The clustering model using the k-means approach yielded the game counts per cluster as shown in Figure 8. We also report the sum of the squared distances of the cluster members from their final centroid in Table 7.

3.2.2 Cluster Characteristics. Now that we have created a clustering model of the game evidence scores, we can inspect the model to see what each cluster might represent about the player/game play evidence of CPS. To that end, we can look at the mean score for each of the 5 domain areas for the members of each cluster. The score scales of the 5 domains scores vary considerably, viz. the

---

### Table 7: CPS Data Cluster Counts

<table>
<thead>
<tr>
<th>cluster_id</th>
<th>size</th>
<th>sum_squared_distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>24</td>
<td>2995.88</td>
</tr>
<tr>
<td>1</td>
<td>24</td>
<td>736.42</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>978.73</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td>2332.40</td>
</tr>
<tr>
<td>4</td>
<td>29</td>
<td>489.45</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>1085.40</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>673.50</td>
</tr>
<tr>
<td>7</td>
<td>18</td>
<td>1623.56</td>
</tr>
</tbody>
</table>

---

Figure 6: Engagement (EN) Sub-Skill Level Probability over Time for a Single Game

![Figure 6: Engagement (EN) Sub-Skill Level Probability over Time for a Single Game](image)

Figure 7: Probability (y-axis) over time (x-axis) for a single game (game id = 114) (Blue=High,White=Med,Orange=Low). Engagement (EN), Monitor Understanding (MU), Feature Identification (FI), Evaluation (EV), Strategy(S)
Table 8: CPS Max/Min Mean Scores for the 5 Domains

<table>
<thead>
<tr>
<th>Domain</th>
<th>Max Mean Score</th>
<th>Min Mean Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>FI</td>
<td>55.625</td>
<td>2.138</td>
</tr>
<tr>
<td>MU</td>
<td>36.545</td>
<td>2.379</td>
</tr>
<tr>
<td>EN</td>
<td>70.727</td>
<td>3.345</td>
</tr>
<tr>
<td>EV</td>
<td>5.273</td>
<td>0.793</td>
</tr>
<tr>
<td>S</td>
<td>6.545</td>
<td>0.069</td>
</tr>
</tbody>
</table>

\[ x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \]

For visualization purposes, we normalized the mean scores as follows:

In Figure 9 we present a graph of the normalized mean scores for each domain across all 8 clusters. We roughly sorted the clusters from left to right within each sub-skill column according to relatively increasing score means.

Cluster 2 (N=11) represents the games that exhibit the highest CPS scores across nearly all domains (except for FI), whereas cluster 4 (N=29) represents the games that exhibit the lowest CPS scores. Given that we didn’t filter out incomplete games, i.e. games where subjects did not make it all the way through the final challenge, it is likely that cluster 4 represents many of these incomplete games. Cluster 6 (N=8) game plays excelled at FI and presented very good scores across the board as well. Cluster 3 (N=25) games provided a balanced set of very good scores, especially in EN and EV. Cluster 5 (N=20) game plays excelled at EV and S. Cluster 1 game plays (N=24) provided fairly weak evidence of CPS skills overall, whereas clusters 7 (N=18) and 0 (N=24) presented low to average scores.

We also loaded the data into a Tableau workbook\(^8\) to analyze the cluster characteristics using various worksheets. In that analysis, we saw a vertical distribution of normalized scores grouped by score feature (EN, FI, MU, S, EV) for each of the 8 clusters that showed that while EN, FI and MU features appeared to have fairly tightly grouped cluster values the features values from S, EV appeared to be much more diffuse within a cluster. As EN, FI, and MU are the important feature drivers of the cluster characteristics we looked at a similar view. That allowed us to examine the cluster distributions across a range of score groupings over EN, FI and MU. In Figure 10 we re-arrange the data to illustrate the vertical cluster scores (the black line indicates the mean) with each column as a cluster.

3.2.3 K-NN Query by Game Id. In addition to the k-means model, we also built a K-Nearest Neighbor (K-NN) model \([1]\) using Graphlab-Create, which allows us to go back and query the data for games that were similar to the source game using a cosine similarity distance metric. A sample K-NN query results are shown in Table 9. The name column is simply a unique identifier based on the concatenation of user, session and game ids.

Table 9: K-NN query results for similar games

<table>
<thead>
<tr>
<th>Name</th>
<th>Distance</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>46:33:211</td>
<td>0.0</td>
<td>1</td>
</tr>
<tr>
<td>207:181:1220</td>
<td>0.28</td>
<td>2</td>
</tr>
<tr>
<td>99:70:711</td>
<td>0.29</td>
<td>3</td>
</tr>
<tr>
<td>611:578:1981</td>
<td>0.32</td>
<td>4</td>
</tr>
<tr>
<td>441:418:1640</td>
<td>0.33</td>
<td>5</td>
</tr>
</tbody>
</table>

3.2.4 Mixture Model Results. In Figure 11 we represent how our application of an EM algorithm learned the dialog score cluster responsibilities over a series of iterations. For 2-D visualization purposes we just show the MU/FI features. The color of each dot represents a blending of cluster probabilities.

As we can see the Mixture Model approach updates the cluster distribution shapes over each iteration, effectively learning the mean and covariance of each distribution. In Figure 12 we plot the final shape of the cluster distributions (k = 4), again limiting this to just the MU and FI score dimensions. As we can see, this method of clustering allowed the model to learn asymmetric elliptical cluster shapes.
shapes and also provided us with probabilistic assignments of each observation to any of the clusters. Thus we are able to represent more robust cluster characterizations beyond a simple in/out hard assignment.

Our interpretation of these data is that the observations in the upper right cluster represent players that were exhaustively exploring the dialog trees which resulted in maximizing their dialog scores. The next cluster to the left represents players who were focused on getting just the data they needed in their collaboration to complete the challenges. The two far left, bottom clusters represent players that were not engaged and probably didn’t play through to the final challenge.
4 DISCUSSION

In this paper we have demonstrated the application of computational psychometrics to gathering insights into a participant’s CPS sub-skills using evidence gathered from an online simulation/game. We showed how we can take the granular evidence gathered from the conversation flow and simulation/game activity data and map that onto our performance level estimates of latent variables. These higher level constructs are driven by CPS subject matter expert tagging and tunable conditional probability tables. This methodology creates a model that can be inspected at any time during the game to provide a probability-based estimate of participant ability. As we move forward with this work we can use this model to start to build more sophisticated simulation/game interactions that could change adaptively, based on our real-time estimate of ability. For example, if we see participants are showing evidence of low feature identification (FI) we can add cues/tips to help them in this facet of interaction.

While the real-time Bayesian evidence tracing has proven useful in generating actionable insights for an individual participant during a game, our clustering work reported here has addressed our need to also compare across games. Our application of k-means gave us the ability to quickly characterize all games in the study and to group similar gameplay with each other, thus yielding different game profiles. Using K-NN we are able to treat these clusters as queryable sets that allow us to find participants that had similar evidence patterns of CPS sub-skills. In applying our Gaussian mixture model we were able to generate a more flexible cluster characterization of each game that can allow for partial cluster membership in more than 1 game profile.

We are working on the next iteration of our Circuit Runner game using the methods and results we have reported here. In our future work we are considering the integration of Bayesian evidence tracing with an application of adaptive conversation flows. We are also incorporating new instruments that will provide more demographics/data on the participants, such as a HEXACO assessment of personality and the results of a CPS questionnaire. We are also considering human-human CPS interaction scenarios that could feature scripted or open-ended conversations.

REFERENCES


