Transferring an existing gaming detection model to different system using semi-supervised approach

Vedant Bahel
G H Raisoni College of Engineering,
Nagpur 442001, India
vbahel@ieee.org

Seth A. Adjei
Northern Kentucky University,
KY 41099, USA
adjeis1@nku.edu

Ryan S. Baker
University of Pennsylvania,
PA 19104, USA
rybaker@upenn.edu

ABSTRACT

Many researchers in Educational Data Mining and Learning Analytics have worked on models for the detection of students who "game the system", a behavior in which students misuse intelligent tutors or other online learning environments to complete problems or otherwise advance without learning. Such detectors are mostly specific to a learning system that they are based on. Researchers popularly use knowledge engineering or machine learning approach in designing the gaming detection models. In this paper, we try to transfer knowledge from an existing detector made for a specific learning system to another, using an unsupervised clustering-based machine learning approach. The goal is to check if the existing detector can be generalized across multiple learning systems with. Specifically, we evaluate how well a gaming detector previously created for Cognitive Tutor Algebra functions adapts to a new learning system, ASSISTments. The results obtained were not very satisfactory and have been discussed thoroughly in this paper.

Keywords

Gaming the system, Transfer Learning, Clustering, Semi-supervised learning, ASSISTments, Cognitive Tutor.

1. INTRODUCTION

In recent years, there has been considerable progress towards designing methods to detect "gaming the system". Gaming has been defined as a behaviour where students try to succeed by exploiting the functionalities of a learning environment instead of learning the material [1,8]. Research in multiple learning environments [9] has linked gaming to poor learning outcomes [10], increased boredom [14] and lower long-term levels of academic attainment [10]. Many researchers have worked on gaming detection methods for specific systems. Both Machine Learning [1,5,14] and knowledge engineering [2,3,5,13] approaches have been used for this purpose. Using knowledge engineering, researchers develop models that are designed to reproduce the knowledge we have about a specific learning behaviour. This is often achieved by designing a set of rules that matches a general common-sense definition of the behaviour [3] or by explicitly eliciting knowledge from an expert about how they determine whether a student is exhibiting a specific behaviour. Most knowledge engineering models of gaming try to identify two main gaming types: help abuse [12] and systematic guessing [11]. Help abuse has mainly been modelled using behaviours that include copying the answer from a hint and repeated help requests. Systematic guessing has been defined operationally as the behaviour of quickly answering questions after the error [2,4,13,15] and making successive errors [5]. A primary advantage of knowledge engineering is that, unlike machine learning, it does not require a large amount of coded data providing examples of students' behaviours since the knowledge is acquired directly from experts. However, often KE models focus only on 1-2 patterns of gaming [3,5], and it is reasonable to question whether such a complex and ill-defined construct can be fully described by 2-3 simple rules [19]. Paquette et. al. worked to develop a knowledge engineered model by identifying certain pattern features of student action that relate directly to gaming behaviour as observed by human experts.

On the other hand, machine learning approaches attempt to resolve the challenge of implicit expertise by leveraging data driven algorithms to discover models from positive and negative examples of a student's behaviour. Using this approach, a large amount of data is automatically inspected to find relationships between the students' fine-grained actions and higher-level behaviours, avoiding the need to explicitly elicit knowledge about the behaviour [16]. In [20], Baker et al discusses machine learning approaches to detect gaming the system. Specifically, the research discusses two primary methods for detecting gaming in Cognitive Tutor: Latent response model and J48 decision tree. Baker et al in [21] also uses step regression for detecting gaming in SQL-Tutor system.

Several researchers have attempted to apply transfer learning to the problem of gaming detection across systems. In this context, Torrey and Shavlik define transfer learning "as the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned" [17]. Transfer learning has been shown to improve the performance of machine learning models where there is limited data [18]. The approach aims to recognize knowledge in the source model and transfer it to the target model. In this research, the source model used for gaming detection is Paquette et. al [4] knowledge engineered gaming detector model built on Cognitive Tutor Algebra (CTA) learning system and the target model is built for the ASSISTments system using a clustering-based semi-supervised approach.

In [8], Paquette et. al successfully attempted to generalize the gaming detector cognitive model into a learning system (Cognitive Tutor Middle School and ASSISTments) with a KE approach. Generalization is important because the cost of building detectors is high and there are hundreds of systems that could benefit from including detectors of this type. Generalization of detectors would make them widely useful across systems. In this paper we attempt to answer the following question: How well does Paquette's transfer learning apply to a new dataset? Could the labelling be recovered if we applied an unsupervised learning technique like clustering? Answer these questions will imply that:

1. Paquette's gaming detection algorithm is truly transferable across systems (ASSISTments & Scatter Plot lesson of Cognitive Tutor for Middle School Math), and
2. The characteristics of student gaming actions can be detected, even with unsupervised techniques, and are truly system agnostic.

2. DATASET & BACKGROUND

For this research, we used data collected from two systems: Cognitive Tutor and ASSISTments. In this section, we describe each of the systems and provide a description of the datasets that were used.

2.1 Cognitive Tutor Algebra

The source model used in this paper for knowledge transfer is Paquette's knowledge engineered model for gaming detection [4]. This model is based on data from the Cognitive Tutor Algebra (CTA) system [7]. The CTA system examines students on advanced mathematical problems and records multiple parameters of the student's learning and question-answer process. Cognitive Tutors are a type of interactive learning environment which uses cognitive modelling and artificial intelligence to adapt to individual differences in student knowledge and learning. The Cognitive Tutor environment breaks down each mathematics problem into the steps of the process used to solve the problem, making the student's thinking visible. If a student is struggling, he or she can also request a hint. When the student requests a hint, the system first gives a conceptual hint. The student can request further hints, which become more and more specific until the student is given the answer (Refer Figure 1). Paquette's model is knowledge engineered on the data obtained from 59 students who used CTA as a part of their regular mathematical curriculum. Data from 12 tutor lessons was obtained and segmented in sequences of 5 actions, called clips, illustrating the student's behaviour. A total of 10,397 clips from this dataset were randomly selected; the chance of a clip being selected was weighted for each lesson according to the total number of clips in that lesson. Those clips were previously coded by an expert to develop machine-learned gaming models and contains 708 examples of gaming the system and 9,689 examples of behaviours that were not coded as gaming.

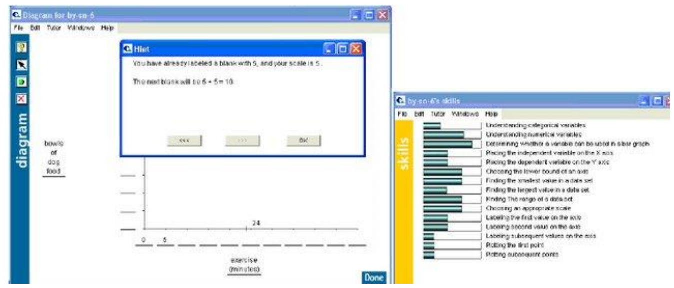

Figure 1. A student requested multiple hints in the Cognitive Tutor Algebra system finally has been prompted with the correct answer.

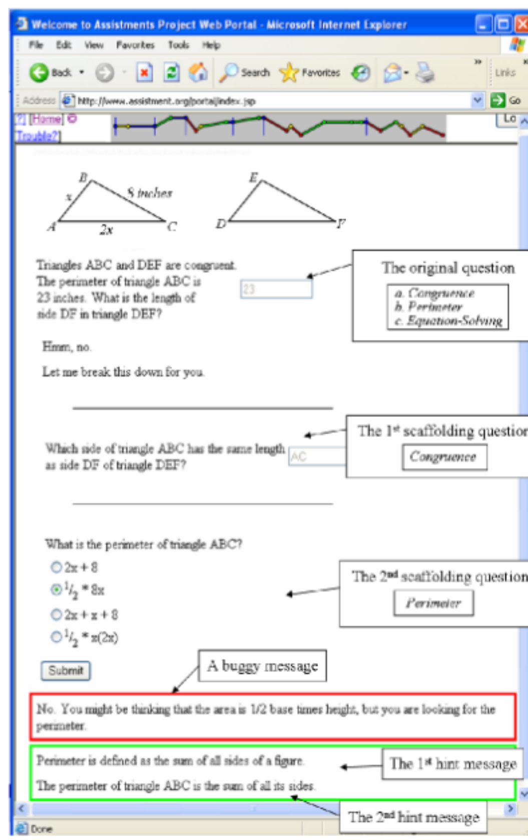

Figure 2. A screen showing a student getting "tutoring" to help the student figure out how to solve a question in the ASSISTments system.

2.2 ASSISTments

The second dataset that we used was collected from the ASSISTments learning system [6], an online system for teachers to assign math homework to students and review student performance as they complete the assignments. This system is similar in many ways to CTA. The ASSISTments dataset contains data collected from 1,367 students' interactions with the system. This dataset was used to test the generalizability of the gaming model created from the CTA system. This data includes a total of 822,233 problem solving actions, which were segmented into 240,450 clips (series of action). But unlike CTA, in ASSISTments, when students are presented with an "original" problem, they only need to provide its final answer. Individual steps are not required of students who solve the problem on the first attempt. However, students who do not provide the correct answer may be required to correctly answer scaffolding questions to successfully complete the problem. Thus, ASSISTments provide an option of scaffolding and hints to students. Thus, ASSISTments problems can be solved in one step if the student's first attempt is correct. As such, a specific clip in this system could have an arbitrarily large number of actions. All the clips with more than 25 actions were removed, since those constituted 0.7% of the data and could have caused serious bias towards a different gaming pattern that was being identified by the expert. Thus, the resulting dataset consisted of 1060 clips labelled by the human expert which constituted 64 gaming clips (6.02%) and 996 non-gaming clips (93.70%) [1, 6, 8].

2.3 Paquette's cognitive model (IBKE)

The cognitive gaming detection model by Paquette et. al [4] is a knowledge engineered model based on how a human expert evaluates gaming behaviours exhibited by a student in a clip. The model implemented was developed using data collected from Cognitive Tutor Algebra (described earlier in this paper) and interview to analyse how an expert observes gaming behaviour. Results indicated that the expert's coding method could be classified into two cognitive processes: interpreting the student's individual actions and identifying patterns of gaming across those actions. Although the expert executes these in parallel, the resulting cognitive model executes these as consecutive steps without changing the fundamental reasoning process. As a result, 13 patterns of action were found to be associated with gaming behaviour, each matching a predefined set of gaming constituents identified in [1]. Finally, the model labelled any clip containing actions that match any of those 13 patterns as gaming. This model is referred to as "Interview-Based Knowledge Engineering" (IBKE) through this paper. It must be noted that we labelled Paquette's model as such,

3. METHOD

We implemented a clustering-based semi-supervised approach to extract patterns identified by the IBKE in CTA and transfer it to the ASSISTments dataset. In this approach, the gaming construct was first transferred between systems, as-is. Then clustering was used to refine the gaming construct, to re-center it after bringing it between data sets. We consider k-means clustering algorithm. k-means is a popularly used clustering algorithm where 'n' clusters are created with random centroids. This algorithm is based on the nearest distance method. All the data points in the dataset get allocated to the cluster with the least distance to the centroid. Once all the points are associated with different clusters. The mean value of features is re-calculated for each cluster and this mean is allocated as the new centroid. This is done until no cluster changes its value after re-calculation. Thus, each centroid creates segments in the data space like cells in a Voronoi diagram.

3.1 Seeding clusters

Though clustering is an unsupervised machine learning method, we seeded one of the clusters. making it a semi-supervised approach. In traditional k-means clustering , a random set of centroids is chosen and further refined after several iterations of the k-means algorithm. In this paper we assign initial centroids based on our prior knowledge of the gaming labels in the dataset, a process we call cluster seeding. The seeding of calculated parameters adds latent knowledge to the un-supervised approach and thereby making it semi-supervised.

3.2 Implementation

For the overall goal of transfer learning, we first ran IBKE (originally developed for the Cognitive Tutor) on the ASSISTments dataset and got the IBKE label for that dataset. The next goal was to use the clustering with IBKE labels as seeds for the ASSISTments dataset. For the same, the average values of the features were calculated for data points with IBKE labelled as gaming and non-gaming, respectively. K-means clustering was used to determine the naturally occurring groupings in the dataset, using IBKE's labels to seed the cluster generation algorithm. In doing so, we experimented with values of k ranging from 2 through 9. This range of values was chosen due to the small size of the dataset. In each case, one cluster was seeded as a gaming cluster and the other clusters were seeded as non-gaming. In other words, for each value of k, all the student actions which IBKE labelled as gaming were initially assigned to a single cluster, and the k-1 non-gaming clusters were created by randomly dividing the IBKE non-gaming data points into k-1 groups. We then run the k-means algorithm with the aim of detecting whether the gaming actions will end up within the same cluster after k-means converges.

Each clustering was evaluated using recall and precision, based on the cluster a point was assigned to and the actual gaming label from the coder. These metrics were chosen based on the fact that k-means clustering naturally generates a categorical classification rather than a probability.

The code repository can be found https://github.com/vedantbahel/clustering-gaming-detection-edm.

4. RESULT & DISCUSSION

The results were inferred by comparing the labels obtained by clustering in ASSISTments with the original (ground truth) labels by a human expert, as in [1]. The results of the k-means clustering scheme is shown in the table below encoded as K#, where # represents the number of clusters.

Table 1. Performance of the various models across the clustering scheme

Clustering Scheme	Recall	Precision
IBKE ASSISTment	0.484	0.234
K2	0.406	0.0704
K3	0.343	0.0721
K4	0.343	0.0698

K5	0.328	0.0766
K6	0.281	0.0810
K7	0.312	0.0738
K8	0.140	0.0638
K9	0.156	0.3703

For comparison, we also display the result of IBKE by comparing IBKE labels to the ground-truth labels [4].

As it can be seen in Table 1, both the performance metrics decreased with increasing numbers of clusters, except for K9. The model generally performed substantially better before using clustering to shift the concept, suggesting that our approach was unsuccessful.

5. CONCLUSION & FUTURE SCOPE

In this paper, we discussed our semi-supervised clustering-based approach to evaluate how well an existing gaming detector designed for Cognitive Tutor Algebra (CTA) system adapts to ASSISTments. We have considered Paquette et al's gaming detector [4] (initially designed for CTA) as the source model for our transfer. Our approach was to consider knowledge from the previous system as a seed for clustering models.

In conclusion, none of the clustering schemes was able to truly outperform IBKE, thus seeding did not truly help with transferring the knowledge. Some of the possible reasons for poor performance might be:

(i) imbalanced data points in each category i.e., 64 gaming and 996 non-gaming data points.
(ii) the nature of the clustering algorithm and how well it fits with the data.

The current findings have not been very conclusive. This suggests that further work needs to be carried out to comprehensively answer the research questions we posed. For next steps, we plan to follow up on other parametric and nonparametric clustering algorithms. Although we did try Expectation-Maximization (EM) based gaussian mixture clustering, it was unsuccessful and showed poorer results. We plan to try other parametric (like DENCLUE, DBSCAN, etc) and nonparametric techniques (like hierarchical, density-based clustering techniques) and look more into the k-means clustering method to understand how cluster shifts in k-means and why it is failing in the current approach. We plan to study the data points which are now being identified as gaming to see what characterizes the false positives. Another reason for the poor results could be class imbalance, as discussed earlier. Some data pre-processing could potentially give a solution to that problem.

6. ACKNOWLEDGMENTS

We would like to thank Luc Paquette for his support during this research.

7. REFERENCES

[1] Baker, R. S., Corbett, A. T., Roll, I., & Koedinger, K. R. (2008). Developing a generalizable detector of when students game the system. *User Modeling and User-Adapted Interaction*, *18*(3), 287-314.

[2] Beal, C. R., Qu, L., & Lee, H. (2006, June). Classifying learner engagement through integration of multiple data sources. In *AAAI* (pp. 151-156).

[3] Muldner, K., Burleson, W., Van de Sande, B., VanLehn, K. (2011). An Analysis of Students' Gaming Behaviors in an Intelligent Tutoring System: Predictors and Impact. User Modeling and User Adapted Interactions, 21, pp. 99–135.

[4] Paquette, L., de Carvalho, A. M., & Baker, R. S. (2014, July). Towards Understanding Expert Coding of Student Disengagement in Online Learning. In *CogSci*..

[5] Walonoski, J. A., & Heffernan, N. T. (2006, June). Detection and analysis of off-task gaming behavior in intelligent tutoring systems. In *International Conference on Intelligent Tutoring Systems* (pp. 382-391). Springer, Berlin, Heidelberg.

[6] Heffernan, N. T., & Heffernan, C. L. (2014). The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, *24*(4), 470-497.

[7] Ritter, S., Anderson, J. R., Koedinger, K. R., & Corbett, A. (2007). Cognitive Tutor: Applied research in mathematics education. *Psychonomic bulletin & review*, *14*(2), 249-255.

[8] Paquette, L., Baker, R. S., de Carvalho, A., & Ocumpaugh, J. (2015, June). Cross-system transfer of machine learned and knowledge engineered models of gaming the system. In *International Conference on User Modeling, Adaptation, and Personalization* (pp. 183-194). Springer, Cham.

[9] Cocea, M., Hershkovitz, A., Baker, R. S. J. d. (2009). The Impact of Off-Task and Gaming Behaviors on Learning: Immediate or Aggregate? Proc of AIED 2009, 507-514.

[10] San Pedro, M. O. Z., Baker, R. S. J. d., Bowers, A., J., Heffernan, N. T. (2013). Predicting College Enrolment from Student Interaction with an Intelligent Tutoring System in Middle School. Proc of EDM 2013, 177-184

[11] Baker, R. S. J. d., de Carvalho, A. M. J. A. (2008). Labeling Student Behavior Faster and More Precisely with Text Replays. Proc of EDM 2008, 38-47.

[12] Aleven, V., McLaren, B. M., Roll, I., Koedinger, K. R. (2006). Toward Meta-Cognitive Tutoring: A Model of Help Seeking with a Cognitive Tutor. Int'l Journal of Artificial Intelligence in Education, 16, 101-130.

[13] Johns, J., & Woolf, B. (2006, January). A dynamic mixture model to detect student motivation and proficiency. In *AAAI* (pp. 163-168).

[14] Baker, R. S., D'Mello, S. K., Rodrigo, M. M. T., & Graesser, A. C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, *68*(4), 223-241.

[15] Gong, Y., Beck, J., Heffernan, N. T., Forbes-Summers, E. (2010). The Fine-Grained Impact of Gaming (?) on Learning. Proc of ITS 2010, 194-203.

[16] Paquette, L., & Baker, R. S. (2019). Comparing machine learning to knowledge engineering for student behavior modeling: a case study in gaming the system. *Interactive Learning Environments*, *27*(5-6), 585-597.

[17] Torrey, L., & Shavlik, J. (2010). Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques (pp. 242-264). IGI global.

[18] Tsiakmaki, M., Kostopoulos, G., Kotsiantis, S., & Ragos, O. (2020). Transfer learning from deep neural networks for predicting student performance. *Applied Sciences*, *10*(6), 2145.

[19] Shih, B., Koedinger, K. R., & Scheines, R. (2011). A response time model for bottom-out hints as worked examples. *Handbook of educational data mining*, 201-212.

[20] Baker, R.S.J.d., Corbett, A.T., Roll, I., Koedinger, K.R., Aleven, V., Cocea, M., Hershkovitz, A., de Carvalho, A.M.J.B., Mitrovic, A., Mathews, M. (2013) Modeling and Studying Gaming the System with Educational Data Mining. In Azevedo, R., & Aleven, V. (Eds.) *International Handbook of Metacognition and Learning Technologies*. pp. 97-116. New York, NY: Springer.

[21] Baker, R. S., Mitrović, A., & Mathews, M. (2010, June). Detecting gaming the system in constraint-based tutors. In *International Conference on User Modeling, Adaptation, and Personalization* (pp. 267-278). Springer, Berlin, Heidelberg.