Identifying Struggling Students by Comparing Online Tutor Clickstreams

By

Ethan Prihar

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Data Science

By

_____

April 2021

APPROVED:

_____

Professor Neil Heffernan, Major Thesis Advisor

Identifying Struggling Students by Comparing Online Tutor Clickstreams*

Ethan Prihar[1], Alexander Moore[1], and Neil Heffernan[1]

Worcester Polytechnic Institute

Abstract. New ways to identify students in need of assistance are imperative to the evolution of online tutoring platforms. Currently implemented models to identify struggling students use costly and tedious classroom observation paired with student's platform usage, and are often suitable for only a subset of students. With the recent influx of new students to online tutoring platforms due to COVID-19, a simple method to quickly identify struggling students could help facilitate effective remote learning. To this end, we created an anomaly detection algorithm that models the normal behavior of students during remote learning and recognizes when students deviate from this behavior. We demonstrated how anomalous behavior not only revealed which students needed additional assistance, but also helped predict student learning outcomes and reduced the confidence intervals in research experiments performed within the online tutoring platform.

Keywords: Online Learning · Tutoring · Unsupervised Learning · Anomaly Detection · Outlier Detection

1 Introduction

Finding patterns in student behavior that correlate negatively with learning is often costly, requiring professional observers to watch students as they complete assignments [22, 3, 12, 15]. Algorithms created to identify these behaviors can be biased toward correctly identifying patterns in select populations [6] and can provide too specific or too great a quantity of information to be practically deployed by an instructor to help their students [12]. Furthermore, a model that requires expensive labeled data is unlikely to be updated often, which introduces model bias as populations and use cases change over time.

* I would like to thank Neil Heffernan, Lane Harrison, Alex Moore, and multiple NSF grants (e.g., 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, 1535428, 1440753, 1316736, 1252297, 1109483, DRL-1031398), as well as the US Department of Education for three different funding lines; a) the Institute for Education Sciences (e.g., IES R305A170137, R305A170243, R305A180401, R305A120125, R305A180401, R305C100024), b) the Graduate Assistance in Areas of National Need program (e.g., P200A180088 P200A150306), and c) the EIR. We also thank the Office of Naval Research (N00014-18-1-2768), Schmidt Futures, and anonymous philanthropy.

These common problems have been exacerbated by recent events. COVID-19 has lead to an unprecedented demand for remote learning [27] and within the online learning platform ASSISTments [11, 20] the number of users has grown tenfold since schools have switched to teaching remotely. Many students and teachers who have made the transition to remote learning have not previously used an online tutoring platform. This can cause inequity in students' quality of learning due to a lack of available resources and access to technology in lower income districts, exacerbating the achievement gap [17, 16, 9].

Unsupervised anomaly detection algorithms are a quickly trainable and deployable method to support instructors during this transition. Anomaly detection can identify unusual student clickstream patterns without needing a labelled dataset. This mitigates the time, expense, and subjectivity associated with manual classroom observation. Once trained, the model can be used to alert instructors when students are behaving abnormally and allow the instructor to assist the students as they see fit.

We define our objectives as follows:

1. Train a model capable of predicting student behavior using only students' clickstream data.
2. Use the student behavior model to identify abnormally behaving students.
3. Investigate the extent to which our measure of anomalous behavior correlates with learning outcomes and engagement.
4. Determine if our anomaly detection algorithm can improve researcher's confidence in experiments performed in ASSISTments.

2 Background

2.1 ASSISTments

ASSISTments is an online learning platform that enables teachers to assign content from their curriculum and assesses student progress in the classroom or remotely [11]. Within ASSISTments, as students complete assigned work, the clickstream data of each student is recorded, aggregated into statistics, and then provided to teachers in reports. These reports inform teachers of the common wrong answers and low performing students in their class. ASSISTments also supports randomized controlled experimentation using its content libraries, allowing independent researchers to test experimental pedagogies. Researchers can create assignments in which students are randomly assigned to different experimental conditions. Each condition contains either no additional tutoring (control) or a new tutoring strategy (treatment). As students complete the experimental assignment, ASSISTments collects data on their performance, which is used to evaluate the effectiveness of the new tutoring strategy [11]. For our anomaly detection algorithm, we used the raw clickstream data collected from students using the ASSISTments tutor to model student behavior, and the data collected during two experiments performed within the platform to determine whether we could increase experimental confidence.

2.2 Related Work

Evaluating Students' Latent Qualities For more than 40 years, knowledge tracing has used data on students' problem responses to estimate subject mastery, which can be used to identify students in need of instructor intervention [8]. Knowledge tracing and its variants stem from mastery learning, an assumption that students can achieve expertise if the domain knowledge is shaped into a hierarchy of component skills, and learning experiences are structured such that prerequisite skills to mastery are taught before subsequent ones [8, 23]. The knowledge tracing process estimates the probability that the student has learned each of the requisite skills necessary to master a task as the student solves exercises. While knowledge tracing can be used to identify struggling students, it does so only by estimating students' mastery of skills. Our anomaly detection algorithm has the potential to recognize struggling students by recognizing atypical behavior, which can include behavior indicative of a lack of skill mastery among other behaviors counterproductive to learning.

Students' clickstream data has also been used to predict their emotional state. Affect detection identifies the emotional state of students and relates that state to their learning gains. Past work has shown that emotions like boredom correlate negatively with learning, while emotions like frustration correlate positively with learning [22, 15]. Initially, affect models were created by observing students' emotional state in class and correlating it with their test scores. Since then, student clickstream data correlated with classroom observation have been used to train affect models, but this method has fallen short at generalizing to different types of students. For example, affect models were less accurate for students from rural areas when the model was trained on data gathered from urban and suburban areas [6]. These models require labeled datasets that are difficult to update without further human observation of students. Generalization of our algorithm to new groups of students comes naturally as new students use the platform, which facilitates custom models for specific groups of students if necessary.

Predicting Students' Behavior In previous studies related to online student behavior, experts created features indicative of cheating based on students' behavior within a massive open online course and trained a classification model to identify labeled cases of cheating [1]. Furthermore, the similar nature of cheating behaviors was used to generalize this model to recognize when other types of cheating occurred [2]. Although this process identified cheating students, it required the creation of informative features and relied upon manually labeled cheating examples. If another type of cheating arises, in which students behave differently than in the initial type of cheating, this method would require new labeled data and potentially new features which would pose a significant ongoing overhead cost. The unlabelled data used to train our anomaly detection algorithm is readily generated as students interact with the online learning platform. No human observation is necessary. If circumstances change, the algorithm can be quickly retrained and implemented.

Another student behavior that has been of interest to the learning science community is gaming. Gaming is an attempt by the student to exploit properties of the tutoring platform to progress, rather than learn the material [4]. In past research, gaming behaviors were identified by experts, and were either algorithmically or manually derived into indicative features [14, 18, 28, 19, 5, 21]. These features were used to create models that could identify students within the tutoring platform who were trying to game the system. These methods relied on experts to confirm which patterns were indicative of gaming, and as new gaming patterns arose, these algorithms fell short. Our anomaly detection algorithm can perform ongoing learning of current and emerging undesired student behavior without the need for expert analysis.

3 Methodology

In order to identify anomalous students, we first trained a model to predict typical student behavior and then used the error in the model's predictions to identify students behaving anomalously. In the following sections we provide details on the data available for model training and evaluation, the structure of the models, and the model's training and validation process.

3.1 Data Processing

Within ASSISTments every action a student takes is recorded. The action records consist of action-timestamp pairs grouped by student and assignment. Working with this clickstream data is an extremely low-level interpretation of students' interactions with ASSISTments; it does not contain additional information such as features of the student, classroom, learning material, or past performance. The types of student actions contained in this data are described in Table 1.

Table 1: Student Actions Recorded in ASSISTments

Student Action	Description
Assignment Started	Student began an assignment
Assignment Resumed	Student returned to an incomplete assignment
Assignment Finished	Student completed an assignment
Problem Started	Student began a problem
Problem Finished	Student completed all parts of a problem
Tutoring Requested	Student viewed tutoring material
Correct Response	Student submitted a correct answer
Wrong Response	Student submitted a wrong answer
Open Response	Student submitted an open response question
Answer Requested	Student was shown the correct answer
Continue Selected	Student moved on to the next problem

Only actions from Skill Builder assignments were used to train the model. Skill Builders are assignments in ASSISTments in which students answer a sequence of problems addressing a single math skill until they answer three problems in a row correctly. Skill Builders were used for training because they have a consistent format and are unlikely to cause divergences in typical student behavior. The distribution of the number of actions taken in Skill Builders is a highly-skewed exponential distribution: almost all students took less than 50 actions to complete each of their assignments, but outlying observations show some students taking 100 to 400 actions.

3.2 The Behavior Prediction Model

For our anomaly detection algorithm to be successful, the behavior prediction model had to be complex enough to capture trends in student behavior, but not so complex that it became capable of predicting the behavior of abnormally behaving students as well. To find a suitable model, we trained a logistic regression [13], neural network [26], decision tree [25], and Bernoulli naïve Bayes classifier [29] to predict a student's next action, given only their previous action and the time since taking an action.

To prepare the clickstream data for model training, we formatted the data into previous-action next-action pairs. To prepare the time data for model training, the time since taking an action was binned into 10 discrete ranges of increasing length. The ranges of the time bins grow to parallel the distribution of time between actions. The models therefore had 21 binary inputs (11 one-hot encoded actions and 10 time bins) and 11 binary outputs (11 one-hot encoded next actions).

To evaluate model quality, 985,000 actions from 7,300 students were used in 5-fold cross validation. The average accuracy, ROC AUC [10], and Cohen's Kappa [7] for each model was calculated and used to select the model used to identify anomalous students in the following evaluation.

3.3 Identification of Anomalous Students

The best model from the previous section, which was a logistic regression, was trained on all the data used in the 5-fold cross validation and was then used to predict the next action of 985,000 actions from 7,300 different students the model had never seen data from before. The average absolute error of the model's predictions across each student's actions became their "anomaly score". To determine if anomaly scores correlated with student performance, we calculated Spearman correlations [24] between the students' anomaly scores and their average correctness and time on task for all the problems the students completed in ASSISTments, excluding the assignments used to calculate their anomaly scores.

In addition to measuring the anomaly score's correlation with performance metrics, we investigated differences between students in the 95th percentile of anomaly scores, which we labeled "anomalous students", and the rest of the students, which we labeled "normal students". We investigated differences in the

frequency of actions taken and the time spent waiting before and after taking actions.

3.4 Improvements to Experimental Confidence

Lastly, it was investigated whether anomaly score could narrow the confidence interval in experimental results. To measure this, the data from two experiments performed in ASSISTments were re-evaluated. Both experiments are randomized controlled trials that each provided an additional piece of instruction during an assignment to students in the treatment group. The first experiment measured if the intervention reduced the number of problems required for students to master the material. The second experiment measured if the intervention reduced the time it took students to master the material.

For both experiments, We recomputed the 95% confidence interval of each experimental condition using a weighted standard deviation, where each student's weight was inversely proportional to their anomaly score; calculated across all their work aside from the work they did during the experiment. If weighting anomalous students less than their peers reduced the confidence intervals, that would support the claim that anomalous students have outlier behavior in experimental settings.

4 Results

4.1 Behavior Prediction Model Evaluation

The four models trained to predict students' next actions all performed relatively well. Each of the models scored highest in at least one of the three metrics calculated, and logistic regression scored highest in two of the metrics. For this reason, logistic regression was the model of choice to evaluate the relationship between anomaly score and student behavior, discussed in the following section. Table 2 shows the cross-validated performance metrics for all models.

Table 2: Performance Metrics for the Proposed Behavior Prediction Models

Model	Accuracy	ROC AUC	Cohen's Kappa
Logistic Regression	**0.71**	**0.96**	0.67
Neural Network	0.70	0.95	**0.68**
Naïve Bayes Classifier	**0.71**	0.94	0.66
Decision Tree	0.65	**0.96**	0.66

4.2 The Behavior of Anomalous Students

The students' anomaly scores, as defined in Section 3.2, correlated significantly with average correctness and time on task. The Spearman correlation coefficient

[24] and p-value of the correlations are shown in Table 3. Students with higher anomaly scores took only slightly less time than students with lower anomaly scores, but got significantly more problems wrong. These results could indicate that students with high anomaly scores have more difficulty learning the material, or exhibit more gaming behavior [5]. This is an encouraging implication as it indicates that anomaly score could be used to inform teachers of struggling students in their classes.

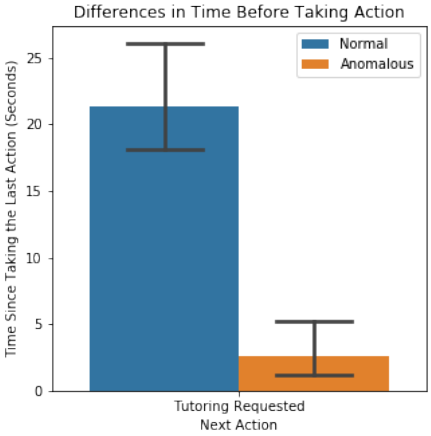
Table 3: Correlation Between Anomaly Score and Student Performance Metrics

Metric	Spearman's Rho	p-Value
Average Correctness	-0.21	<.001
Average Time-on-Task	-0.04	<.001

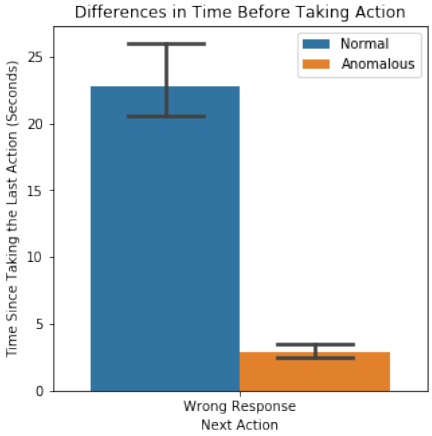
Additionally, when investigating the differences between normal and anomalous students, as defined in Section 3.2, wrong answers occurred 60% more frequently and correct responses occurred 32% less frequently in anomalous students' action sequences. The time a student waited before and after they submitted a wrong answer or received tutoring was also significantly different between normal and anomalous students. Figure 1 shows the average and 95% confidence intervals for the time before and after taking these actions. Figures 1a and 1b show that anomalous students spent about 20 seconds less looking at the problem before requesting tutoring or submitting a wrong answer. Figure 1c shows that anomalous students spent about 30 seconds less looking at tutoring and Figure 1d shows that anomalous students spent about 50 seconds less thinking about their wrong response before performing another action. These statistics paint the picture of a student that rushes to answer a problem, frequently submits wrong responses, and quickly requests tutoring. Then, without spending the time to process the new information, submits more wrong answers until they are eventually able to move on. This behavior is essentially the definition of gaming [5], and would certainly be of interest to teachers as it is counterproductive to learning and should be corrected. Students' anomaly scores could therefore be a useful tool for identifying students in need of instructional intervention without having to define, or even be aware of, the specific kinds of negative behaviors of the students.

4.3 The Effects of Anomalous Students on Experimental Confidence

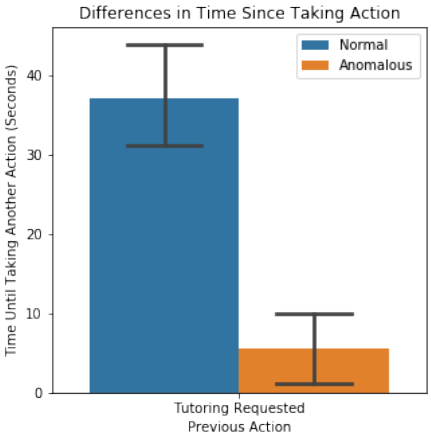
The unweighted and weighted confidence intervals for each experimental condition are shown in Table 4. In three of the four conditions, the size of the confidence interval decreased. If weighting each student inversely proportional to their anomaly score reduced the confidence intervals of the experimental conditions, this implies that anomalous students were often the outliers in these experiments. Using a weighted confidence interval could help reduce noise in experimental outcomes when the clickstream data of students are available.



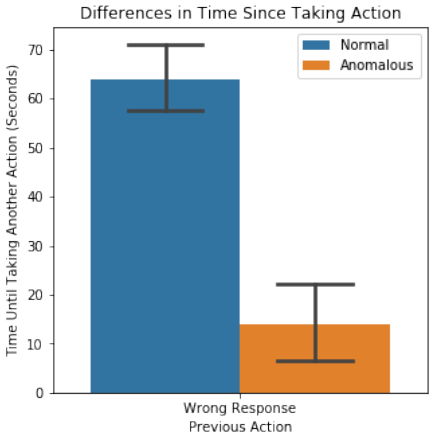
(a) Time Spent Before Requesting Tutoring



(b) Time Spent Before Submitting a Wrong Response



(c) Time Spent After Requesting Tutoring



(d) Time Spent After Submitting a Wrong Response

Fig. 1: The Average Time Spent by Normal and Anomalous Students Before and After Requesting Tutoring and Submitting Wrong Responses with 95% Confidence Bars

Table 4: Unweighted and Weighted 95% Confidence Interval for Each Experimental Condition

Condition	Regular CI	Weighted CI
Experiment 1 Control	**0.88** Problems	0.98 Problems
Experiment 1 Treatment	0.71 Problems	**0.70** Problems
Experiment 2 Control	122 Minutes	**117** Minutes
Experiment 2 Treatment	98 Minutes	**93** Minutes

5 Limitations and Future Work

While using students' clickstream data to identify anomalous students has the potential to improve educational practices, there are no guarantees that this algorithm will identify students with the same unproductive behaviors that we have found in ASSISTments clickstream data. By creating an unsupervised metric for student behavior we have removed the bias introduced by human labels but have also removed human values from our algorithm. This could pose an issue if a majority of students needed assistance. In such a scenario, the anomalous students would be the high achievers. Care should be taken when implementing this algorithm to manually examine the behavior of anomalous students to make sure that the algorithm's determination of anomalous behavior matches the expectation for the proposed use case. In the future, work could be done to modify this algorithm to accept an example of anomalous behavior, which it could generalize in a semi-supervised context. This could alleviate the need to manually examine the behavior of students, which while time consuming, is still preferable to creating a labelled dataset.

Using this anomaly detection algorithm to calculate a weighted confidence interval for experimental conditions also poses some limitations. The primary limitation is that there is no guarantee that the anomalous students are not important to the results. For example, a treatment condition could remediate anomalous behavior. If this is the case, giving lower weights to anomalous students could make the treatment appear ineffective when really it is particularly effective on anomalous students. Knowing what causes students to be labeled anomalous would help inform when to use this anomaly detection algorithm. Future work could develop an algorithm to explain the behavior of anomalous students.

6 Conclusion

Students' anomaly scores, calculated only by comparing their clickstreams, negatively correlated with their average correctness and time on task. Additionally, anomalous students spent significantly less time thinking about a problem before getting the answer wrong or requesting tutoring, and once they were told they got the answer wrong or shown tutoring, they spent significantly less time before attempting the problem again. Using ASSISTments data, the anomaly detection algorithm was able to identify a common mode in unusual student behavior: rushing to complete assignments without trying to learn, i.e., gaming [5]. While this algorithm has the potential to be used to inform teachers in real time if their students need assistance, the behaviors identified as anomalous must be examined before choosing how to address them, lest students receive irrelevant interventions because of an incorrect assumption of what it means to be anomalous.

References

1. Alexandron, G., Lee, S., Chen, Z., Pritchard, D.E.: Detecting cheaters in moocs using item response theory and learning analytics. In: UMAP (Extended Proceedings) (2016)
2. Alexandron, G., Ruipérez-Valiente, J.A., Pritchard, D.: Towards a general purpose anomaly detection method to identify cheaters in massive open online courses (06 2019)
3. Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z.: Off-task behavior in the cognitive tutor classroom: when students" game the system". In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 383–390 (2004)
4. d Baker, R.S., Corbett, A.T., Roll, I., Koedinger, K.R.: Developing a generalizable detector of when students game the system. User Modeling and User-Adapted Interaction **18**(3), 287–314 (2008)
5. d Baker, R.S., Mitrović, A., Mathews, M.: Detecting gaming the system in constraint-based tutors. In: International Conference on User Modeling, Adaptation, and Personalization. pp. 267–278. Springer (2010)
6. Botelho, A.F., Baker, R.S., Heffernan, N.T.: Improving sensor-free affect detection using deep learning. In: International Conference on Artificial Intelligence in Education. pp. 40–51. Springer (2017)
7. Cohen, J.: A coefficient of agreement for nominal scales. Educational and psychological measurement **20**(1), 37–46 (1960)
8. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction **4**(4), 253–278 (1994)
9. DeWitt, P.: Teachers work two hours less per day during covid-19: 8 key edweek survey findings. Education Week (2020)
10. Fawcett, T.: An introduction to roc analysis. Pattern recognition letters **27**(8), 861–874 (2006)
11. Heffernan, N.T., Heffernan, C.L.: 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 (2014)
12. Holstein, K., McLaren, B.M., Aleven, V.: Student learning benefits of a mixed-reality teacher awareness tool in ai-enhanced classrooms. In: International conference on artificial intelligence in education. pp. 154–168. Springer (2018)
13. Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X.: Applied logistic regression, vol. 398. John Wiley & Sons (2013)
14. Johns, J., Woolf, B.: A dynamic mixture model to detect student motivation and proficiency. In: Proceedings of the national conference on artificial intelligence. vol. 21, p. 163. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999 (2006)
15. Lehman, B., Matthews, M., D'Mello, S., Person, N.: What are you feeling? investigating student affective states during expert human tutoring sessions. In: International conference on intelligent tutoring systems. pp. 50–59. Springer (2008)
16. Levinson, M., Cevik, M., Lipsitch, M.: Reopening primary schools during the pandemic (2020)
17. Middleton, K.V.: The longer-term impact of covid-19 on k–12 student learning and assessment. Educational Measurement: Issues and Practice (2020)

18. Muldner, K., Burleson, W., Van de Sande, B., VanLehn, K.: An analysis of students' gaming behaviors in an intelligent tutoring system: Predictors and impacts. User modeling and user-adapted interaction **21**(1-2), 99–135 (2011)
19. Murray, R.C., VanLehn, K.: Effects of dissuading unnecessary help requests while providing proactive help. In: AIED. pp. 887–889. Citeseer (2005)
20. Ostrow, K.S., Heffernan, N.T.: Advancing the state of online learning: Stay integrated, stay accessible, stay curious. Learning science: Theory, research, & practice pp. 201–228 (2019)
21. Paquette, L., Baker, R.S.: 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 (2019)
22. Pardos, Z.A., Baker, R.S., San Pedro, M.O., Gowda, S.M., Gowda, S.M.: Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. In: Proceedings of the third international conference on learning analytics and knowledge. pp. 117–124 (2013)
23. Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L.J., Sohl-Dickstein, J.: Deep knowledge tracing. Advances in neural information processing systems **28**, 505–513 (2015)
24. Schober, P., Boer, C., Schwarte, L.A.: Correlation coefficients: appropriate use and interpretation. Anesthesia & Analgesia **126**(5), 1763–1768 (2018)
25. Steinberg, D., Colla, P.: Cart: classification and regression trees. The top ten algorithms in data mining **9**, 179 (2009)
26. Svozil, D., Kvasnicka, V., Pospichal, J.: Introduction to multi-layer feed-forward neural networks. Chemometrics and intelligent laboratory systems **39**(1), 43–62 (1997)
27. UNESCO: 290 million students out of school due to covid-19: Unesco releases first global numbers and mobilizes response. UNESCO (2020)
28. Walonoski, J.A., Heffernan, N.T.: Prevention of off-task gaming behavior in intelligent tutoring systems. In: International Conference on Intelligent Tutoring Systems. pp. 722–724. Springer (2006)
29. Zhang, H.: Exploring conditions for the optimality of naive bayes. International Journal of Pattern Recognition and Artificial Intelligence **19**(02), 183–198 (2005)