Expertise Estimation based on Simple Multimodal Features

Xavier Ochoa, Katherine Chiluiza, Gonzalo Méndez, Gonzalo Luzardo,
Bruno Guamán and Jaime Castells
Centro de Tecnologías de Información, Escuela Superior Politécnica del Litoral
Guayaquil, Ecuador
{xavier, kchilui, gmendez, gluzardo, bguaman, jcastells}@cti.espol.edu.ec

ABSTRACT

Multimodal Learning Analytics is a field that studies how to process learning data from dissimilar sources in order to automatically find useful information to give feedback to the learning process. This work processes video, audio and pen strokes information included in the Math Data Corpus, a set of multimodal resources provided to the participants of the Second International Workshop on Multimodal Learning Analytics. The result of this processing is a set of simple features that could discriminate between experts and non-experts in groups of students solving mathematical problems. The main finding is that several of those simple features, namely the percentage of time that the students use the calculator, the speed at which the student writes or draws and the percentage of time that the student mentions numbers or mathematical terms, are good discriminators between experts and non-experts students. Precision levels of 63% are obtained for individual problems and up to 80% when full sessions (aggregation of 16 problems) are analyzed. While the results are specific for the recorded settings, the methodology used to obtain and analyze the features could be used to create discriminations models for other contexts.

Categories and Subject Descriptors

K.3.1 [**Computing Milieux**]: Computers and Education-Computer Uses in Education

Keywords

Multimodal Learning Analytics, Math Data Corpus

1. INTRODUCTION

Learning Analytics is a new field that attempts to improve the learning process through the automatic measurement of the activities of participants in such process. In that respect is not different from the field of Business Analytics, with maximization of learning efficiency instead of maximization of profit at its goal. However, while in business the vast majority of relevant actions are by necessity kept on record, in learning, much of what happens during the process is not recorded and cannot be used to evaluate it.

The most readily available sources of learning data are the interactions of students and instructors in e-learning platforms. As most of these tools keep detailed logs of access and content consumption and production, it helps researchers to collect and process large amount of data that could provide insight in the usage and interactions within these tools. Yet, most of the traditional learning processes occurs in face-to-face settings with very little record keeping, apart from the memory of the participants and short and unstructured notes made by the instructors and students. To avoid the proverbial mistake of only searching where it is easy to search, new sources of data about the learning process should be recorded and analyzed. These sources are bound to be of varied nature: video and audio recordings, eye tracking information, biometric measurements, digital tools usage, among others. The capture and combined analysis of these diverse data is the focus of the sub-field of Multimodal Learning Analytics [1]. However, due to its novelty and perceived complexity, not much research is done in this sub-field. Apart from seminal works made by Blickstein [1], Worsley [2] and Scherer et al. [3], it is an unexplored field with great potential for discoveries.

The possibilities of Multimodal Learning Analytics are supported by the great advance in multimedia processing technology in recent years. The developed technologies and algorithms provide ways to tracks objects [4] and people[5] in videos , to produce good quality transcripts of audio or to identify letters and figures in sketches [6]. All these techniques can be used to obtain useful features from multimodal recordings of student activities in the real world.

This work used existing multimedia processing technologies to produce a set of simple features from a multimodal dataset of recordings of groups of students solving mathematical problems. These features were used to answer the following question: which factors of students' behavior, while solving a problem, are good predictors of an expert in a given group? The answer to this question provides ways to automatically identify the experts in groups and even provides feedback to instructors about the students' capabilities, one of the main purposes of Learning Analytics.

The structure of the paper is as follows: Section 2 presents a brief description of the multimodal dataset used. Section 3 presents the extracted features, with details of the rationale, algorithms and software used to obtain them. Section 4 describes statistical and classification approaches followed to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICMI MLA '13 Sidney, New South Wales Australia
ACM ACM 978-1-4503-2129-7/13/12 $15.00
http://dx.doi.org/10.1145/2522848.2533789.

determine the discrimination power of these features to predict the expert in a group. Section 5 discusses the findings of the previous section and provides light on their usefulness. Section 6 mentions related work, and finally Section 7 presents the conclusions of the work and ideas for further research.

2. DATASET

The data analyzed in this paper corresponds to the video, audio and digital pen information included in the Math Data Corpus (MDC) [7], a set of resources publicly available to the participants of the Second International Workshop on Multimodal Learning Analytics.

The MDC was composed by twelve high-fidelity time-synchronized multimodal recordings on collaborating groups of teenage students trying to solve several geometry and algebra problems. It also included several human-coded resources about: *a*) whether the problems were correctly solved by the participant students, *b*) temporal information associated to each problem, *c*) representational codification of the students' writing (not available for the complete set of problem solving sessions), and *d*) temporal offsets between the pen strokes and the media files of the recorded sessions (only available for six of the twelve sessions).

In total, the dataset contained multimodal information of 18 different students participating in 12 problem solving sessions. In each session, a group of 3 students worked together to solve a set of mathematics problems, each of which belonged to a different difficulty level: easy, moderate, hard and very hard. The students of each group met and worked twice, in two separate sessions, to solve two distinct sets of problems. In each of these sessions, one of the students was assigned as the leader of the group in order to interact, on behalf of the other members, with a computer system that displayed the students the problems to solve and received the answers submitted. The resources of the MDC also included details on the designated leader of each session and the system used to uniquely identify the students.

In a previous study described in [7], the problem solving sessions included in the MDC were manually assessed by several human evaluators to determine the expert student of each recorded session. To this end, a grading scale was established: a student received a positive or negative score according to whether he or she correctly answered a given problem or not. The assigned score depended on the difficulty level of the corresponding problem. For each session, the student's individual scores were summed up into an expertise score.

For a full description and additional details on the Math Data Corpus, the reader is referred to the work of Oviatt et al. [7].

3. FEATURE EXTRACTION

For each session recording, the audio, video and strokes files of each student were split into small pieces corresponding to the individual problems solved by each group. This segmentation was based on the time boundaries information detailed in the coding data related to the Math Data Corpus.This section describes the processing stages applied to each type of input data from the MDC along with the procedure used to extract the different features that were used for the expertise estimation. Due to the multimodal nature of the data, this section is divided by type of media.

All the software used for the feature extraction and its posterior analysis is freely available online [1] in order to provide means of verification and repeatability.

3.1 Video

3.1.1 Calculator Use

One of the hypothesis that lead our analysis was that the number of times a student uses the calculator ($NTUC$) while trying to solve a math problem should be a good indicator of whether he or she actually knows what inputs should be provided to the calculator in order to solve the given problem.

The first step to calculate this feature was to determine the position of the calculator and the direction in which it was pointing at. The top-down view video, that contains a close-up of the table where the students are working was used because it best captured the details of the calculator. An image of the calculator was captured manually from this video. An implementation of the Speeded Up Robust Features (SURF) technique [8] provided in the OpenCV library [9] was used to extract the feature points of the calculator images. The SURF algorithm was then applied to each frame of the video to obtain the feature points. The Fast Approximate Nearest Neighbor Search (FLANN) [10] library was used to match the feature points of the captured image of the calculator with the feature points of each frame. The best matched points were used to calculate the position of the calculator averaging their x and y coordinates and the direction in which it was pointing at using the rigid transformations capabilities provided by OpenCV. While there were some frames in which this matching was not possible due to object occlusions or changes in the illumination of the calculator, in general the described detection technique was robust and provided useful position and direction data.

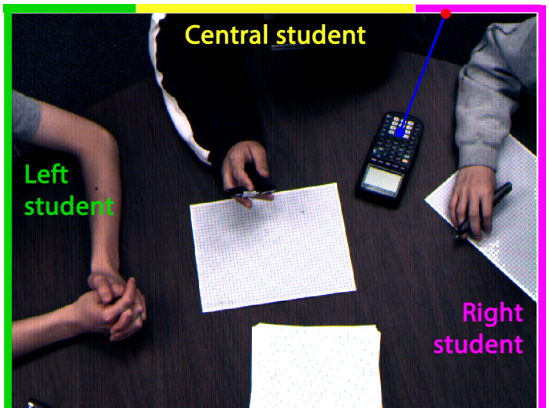


Figure 1: Determination of which student is using the calculator in the given frame. Colored edges indicate the working area of each student.

Using the calculator center point and the direction to which it was pointing at, a set of other points lying on the same 2D line were obtained. In MATLAB, these points were generated over a segment of the calculator direction line that

[1] http://ariadne.cti.espol.edu.ec/xavier/mla13

was traced up to touch either the left, top or right border of the frame. Specific intervals of these edges were used to define which parts of the video frame exclusively belonged to the working area of each student during the problem solving session. Figure 1 depicts the edge points that define the students' working areas. It also shows the results of our algorithm indicating that, in the shown scene, the calculator is being used by the student located at the right side of this view. This result is indicated by the intersection point of the calculator direction line with the part of the frame border corresponding to the right student. Since during each session the students changed their positions, a further student-position matching was needed to establish which student was located at the left, center and right areas of each frame. This mapping process was performed considering the time boundary information of each problem provided in the coding resources of the MDC. Finally, once the total number of times in which the calculator was used by each student was found, a proportion of its usage was computed in relation with the total number of frames where the tracking algorithm was able to successfully find the calculator. This feature is referred as Proportion of Calculator Usage (PCU).

3.1.2 *Total movement*

The total movement (TM) of a student represents the degree to which he or she moved during the solving problem session. It is hypothesized that the movement is related to the leadership and expertise. This measurement was calculated by processing the frontal videos of each student participating in a group contained in the MDC.

To determine the total movement of each student at a specific video frame, a movement model image was obtained as a result of the subtraction of the current frame and the previous one. This model was obtained by applying the CodeBook algorithm [11, 12], which determines all the significant changes between two consecutive frames and discard small variations caused by noise or changes in the lighting conditions. As a result of this algorithm, a binary image, where moving areas are represented by white regions, is obtained (see Figure 2).

The total movement of a student in a given frame is defined as the number of white pixels contained in the binary image output by the Codebook algorithm. This magnitude, when computed for the entire problem solving session, results from summing up its individual values obtained from each frame that compose a problem recording.

(a) Original frame (b) Difference frame

Figure 2: Results of the Codebook algorithm.

3.1.3 *Distance from the center of the table*

The distance of each participant to the center of the table (DHT) could be a measure of how concentrated the student is over the solution of the problem. It was calculated by first finding their position in the video and then calculating their distance to the center of the table at each frame. At the end, the averages of these distances were calculated for every problem resolution.

A head detection and tracking algorithm was used instead of following the whole body, because this part of the body was clearly visible in the videos. Participants moved considerably during each session and so a robust algorithm was needed not just for tracking their heads on a wide-angle top video, but also learning as their appearance changes. For this task, the Tracking-Learning-Detection (TLD) [13] algorithm was used.

OpenTld [13, 14], a C++ implementation of TLD, was used for tracking each participant's head. Three instances of Opentld were created, one for each student. First, the head of each participant is encircled in a bounding box at the first frame of the video. Then, at each subsequent frame, the algorithm tracks the head and learns any change on its appearance despite how much it moves. When detected, the object is bounded in a box and its centroid coordinates are saved for further processing. The Euclidean distance from each head centroid to the center of the table is calculated and then, the average of these distances is obtained by problem (see Figure 3). Additionally, the variance of the average distance head to table (SD-DHT), was calculated to determine if a participant remains mostly static or varies his or her distance to the table.

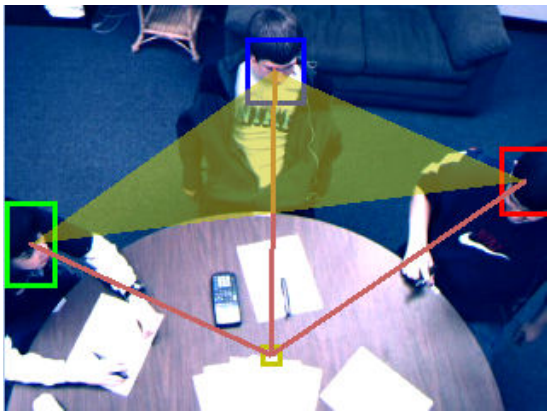


Figure 3: Calculation of the distance of the student's head to the center of the table.

3.2 Audio

An automatic transcription module generated the text representation of the words spoken by each student during each solving problem session. The Microsoft Speech Platform [2], the FFmpeg libraries [3], and the Google Speech to Text API [4] were used to this end.

After the problem-based segmentation stage, each problem was further segmented into several smaller recognizable

[2] microsoft.com/en-us/download/details.aspx?id=10121
[3] www.ffmpeg.org
[4] gist.github.com/alotaiba/1730160

audio units. The starting and ending timestamps of any audio portion that purely contained noise or non-recognizable speech fragments (i.e. loud breathing, excessive blow, and puffs) were automatically identified to rule the corresponding segments out of the subsequent speech recognition phase.

Two different recognition engines were individually applied to the recognizable units resulting from the previous segmentation: *a*) A Microsoft speech recognizer with a general, context free, dictation grammar, and *b*) a web-based recognizer that used the Google Speech Recognition online service. Both speech-to-text engines allowed us to recognize the audio pieces as free dictated text, without requiring any assumptions about specific contexts or word order to successfully identify and interpret the audio input. The results of the Google recognizer, whose accuracy has been reported between 17 and 20 percent for the Word Error Rate (WER) [15] and about 75% for the sentence-level semantic [16], were generally superior and more accurate than the Microsoft technology.

Using the outputs of the Microsoft and Google recognizers, a unique, unified, transcription file was generated for each problem solved by every student. The building process of this unified transcription did not require any time-based alignment since the input audio files processed by both recognizers resulted from the same segmentation process and, thus, corresponded to the same time intervals.

The recording of all students' voices was not considered in the audio processing stage since the speech recognition results of these files were too poor to be considered as significant. They were mostly recognized as noise.

In order to cope with the inherent language variations (inflected words, conjugation tenses, plurals), when looking for key words, the stats generator process included a stemming component to reduce the transcripts and any searched term to their written root form. The English stemmer used was based on the Lovins algorithm proposed in [17].

The unified transcription file was processed by a stats generator algorithm that characterized each student's behavior from their speech signal as follows: Two elementary stats were computed without considering the students' speech content and three others were based on the analysis of what each student actually said during the problem solving session. Following, a brief description of each of these features is presented:

Number of interventions (NOI): Indicates how many times a student took part or participated speaking while trying to solve a problem. This measurement only considered the recognizable audio pieces found.

Total speech duration (TSD): Resulted from summing up all the single time lengths (in seconds) of each student's intervention while solving a problem.

Times numbers were mentioned (TNM): Indicates how many interventions, a given student mentioned numbers. The algorithm looked for numbers that were detected by the speech recognizers either in its numeric representation (e.g. 2, 100) or as text (e.g. two, one hundred). It also considered several formats, including decimals and fractional numbers. The percentage of times when an intervention mentions numbers (PNM) was also calculated.

Times mathematical terms were mentioned (TMTM): Accounts for the number of interventions in which a student refers to a math term. To decide whether a given word can be considered as mathematical terminology, the stats generator consulted a list of 1,352 math terms obtained from the Mathwords [5] website. The percentage of times when an intervention mentions mathematical terms ($PMTM$) was also computed.

Times commands were pronounced (TCP): Indicates the number of times in which, within his/her intervention, a student pronounces one of the predefined commands that the computer assistant was able to understand (e.g. "ready to work").

3.3 Digital Pen

The information produced by the students' digital pen was processed using the stroke representations capabilities of PaleoSketch [18], a low-level sketch recognition framework promoted by the Sketch Recognition Lab of the Texas A & M University. For the sketch recognition process, the functionalities provided in the Strontium sketch recognition library [6] were used.

The goal of the processing stage applied to this data was detecting and recognizing all the significant geometrical primitives drawn by the students while trying to solve the problems. After the corresponding problem-based segmentation, a temporal offset was applied to each set of strokes in order to align them with the starting point of each experiment recording. This time-based alignment was performed using the information available in the linkage time files associated to six of the twelve experiments of the MDC. This information was not available for all the other sessions, because of which we manually calculate it by closely analyzing each recording against the digital pen data.

The used sketch recognition engine was able to identify five different types of low-level geometrical primitives: lines, rectangles, circles, ellipses and arrows. Unfortunately, it was not able to recognize any higher-level shape composed by other elementary sketches: It recognized complex structures only when they were drawn in one single continuous trace (a long stroke produced without lifting the pen).

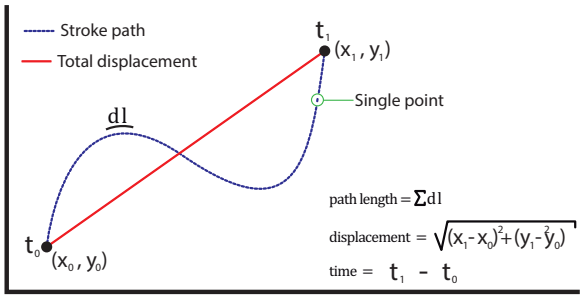


Figure 4: Scheme of magnitudes associated to a single pen stroke.

Two categories of features were obtained from the digital pen data. First, the following basic measurements were computed from the traces sets:

Total number of strokes (TNS): Indicates how many continuous traces were done by a given student during the problem solving session.

[5] www.mathwords.com
[6] github.com/eyce9000/strontium

Table 1: Coefficients of the Logistic Model Predicting Odds for a Student Solving Correctly a Problem

Predictor Variable	B	$Wald$	df	p value	$exp(B)$
Number of Interventions (NOI)	0.068	24.019	1	0.001	0.934
Times numbers were mentioned (TNM)	0.175	23.816	1	0.001	1.192
Times commands were pronounced (TCP)	0.329	4.956	1	0.026	1.390
Proportion of Calculator Usage (PCU)	1.287	25.622	1	0.001	3.622
Fastest Student in the Group (FW)	0.924	18.889	1	0.001	2.519
$Constant$	1.654	53.462	1	0.001	0.191

Average Number of Points (ANP): Represents, in average, the number of points that compose each stroke drawn.

Average Stroke Time Length ($ASTL$): Accounts for the number of milliseconds that the student needed, in average, to complete each stroke.

Average Stroke Path Length ($ASPL$): Represents the average number of pixels that the trajectory of strokes drawn had.

Average Stroke Displacement (ASD): Accounts for the average displacement defined by the starting and ending points of each stroke.

Average Stroke Pressure (ASP): Represents the average pressure with which each stroke was drawn by the student.

Figure 4 shows some of the features described above when they are observed for one single stroke. The scheme illustrates a stroke starting at point (x_0, y_0) and ending at (x_1, y_1) that has been drawn between the millisecond t_0 and millisecond t_1. This trace is composed by a sequence of individual points (as the one indicated with the green circle) each of which has a timestamp associated. The corresponding formulas for the path length, the displacement and the time length are also shown.

Using the stroke classification features from the Strontium library, the following features were computed for each student problem solving session: number of lines sketched (NOL), number of rectangles sketched (NOR), number of circles sketched (NOC), number of ellipses sketched (NOE), number of arrows sketched (NOA) and, finally, the addition of all of the geometrical figures sketched (NOF).

4. RESULTS

To answer the research question, two approaches have been followed. First, most of the variables described in the previous section were used to predict the odds and probability of a student solving correctly a problem. 567 units of analysis were included in a logistic regression analysis that used SPSS version 20 for Mac OS. Second, to predict who the expert is in each group, the values of the different variables were averaged by session and student. 36 different unit of analysis were obtained for this second approach. Moreover, Group 2 of the dataset has no defined Expert. All students from this group were removed from the dataset, leaving only 30 valid cases. Due to this low number of cases, traditional statistical methods, such as logistic regression, were not sufficient to create a model to predict if a student is an expert in the group. Instead, the technique of Classification Trees [19] was used to identify which variables are able to discriminate between Experts and Non-Experts. This technique creates binary trees, determining which values of the variables create the best partitioning in the dataset and its subsequent sub-sets. Classification Trees, provided by rpart library [20] in the R statistical software [21] for Mac, were used in this second part of the analysis.

4.1 Odds of a student solving correctly a problem

A Logistic regression was run with Student Solving Correctly a Problem (SSP) as the dependent variable and DHT, $SD-DHT$, TM, NOI, TSD, TNM, PNM, $TMTM$, $PMTM$, TCP, $NTUC$, PCU, TNS, NOL, NOR, NOC, NOE, NOA, $ASPL$, ANP, ASD, $ASTL$ and ASP as predictor variables. Additionally a derived variable to mark the fastest writing student FW was added to compensate for the variability of the original variable $ASTL$. The resulting model was significant reliable ($\chi^2 = 100.67$, $df = 24$, $p < 0.001$).This model accounted for between 16.3% (Cox and Snell's R-square) and 23.5% (Nagelkerke's R-square) of the variance in problem solved correctly status, with 63.5% of the correctly solved problems successfully predicted. However, 72.7% of the incorrectly solved problems were accurate. The overall accuracy of the model was 70.2%, a cutoff value of 0.3 was used due to the fact that the dataset included around 70% of incorrectly solved problems. In this model, the following variables were significant predictors of correctly solving a problem by a student: NOI ($p = 0.039$), TNM ($p = 0.013$), TCP ($p = 0.021$), PCU ($p < 0.001$) and FW ($p < 0.001$). Once these variables were identified, a second run for building a new model was performed; however, in this run the predictor variables were only those identified as significant predictors in the previous run. The resulting model was significant reliable ($\chi^2 = 88.35$, $df = 5$, $p < 0.001$) and accounted for between 14.4% and 20.9% of the variance of problems solved correctly by a student. The proportion of cases correctly predicted as solved were 60.9% whereas 71.8% of the cases predicted as incorrectly solved were accurate. The overall percentage of accuracy in the model was 68.8% and the same previous cutoff value was used. Table 1 presents the coefficients, Wald statistic, degrees of freedom and level of significance associated to each predictor variable in this model. The values of the coefficients reveal that an increase of 1 intervention when solving a problem is associated with a decrease in the odds of correctly solving a problem by a factor of 0.93, and that each unit increase in: times numbers were mentioned, times commands were pronounced, proportion of calculator usage and the fastest student writing a stroke increases the odds of correctly solving a problem by a factor of 1.19,1.39, 3.62, 2.52, respectively.

Table 2: Classification tree splits with only non-normalized features

Variable	Value for Experts	Discrimination Power
PCU	> 0.41	4.44
PNM	> 34.74	3.19
$ASPL$	< 38.05	2.86
NOR	< 0.13	2.86
$TMTM$	> 6.25	2.65

To calculate the probability of correctly solving a problem by a student (P) the following formula should be used:

$$P = \frac{e^{-11.7-0.1NOI+0.2TNM+0.3TCP+1.3PCU+0.9FW}}{1+e^{-11.7-0.1NOI+0.2TNM+0.3TCP+1.3PCU+0.9FW}} \quad (1)$$

4.2 Expert prediction

All the features described in section four were input in the Classification Tree algorithm. The results, shown in Table 2, suggest that PCU, PNM, $ASPL$, NOR and $TMTM$ have the higher discriminant values. Using the highest discriminant (PCU), the produced tree classified correctly in the dataset 80% of the time (8 out of 10) and identified non-experts correctly also 80% of the time (16 out of 20). Converting the model to words, it says that if a student uses the calculator in the session more than 40% of time, he or she is an expert. Selection by chance would be 33% for experts and 67% for non-experts. The classification tree fares much better, especially identifying experts.

The precision of the classification could be improved if the variables were normalized, that is, if they were comparable among sessions. To normalize the features, they were converted to a binary value in the form of: 1 if the students has the highest or lowest value of that feature in the session, 0 if not. The selection of highest or lowest was determined by the perceived belief of correlation between a given variable and the expertise. For example, in the case of the Stroke Time ($ASTL$), it is believed that a shortest time is indicative of expertise, then the student with the lowest $ASTL$ in the group was assigned 1. On the other hand, the numbers of times numbers are mentioned (TNM) correlates positively with the expertise and as such, a 1 was assigned to the student with the highest TNM. Table 3 shows the new calculated features. The resulting classification tree, mixing the original variables with the binary variables provides a better discrimination. Table 4 presents the new discriminant value of the variables. The fastest writer (FW) seems to dominate the best discriminants for expertise, improving over PCU. Using FW, the tree is able to identify correctly the experts in 80% of the cases (8 out of 10) and the non-experts in 90% of the cases (18 out of 20).

Even if the last classification tree is able to highly discriminate experts and non-experts after the session is over, it is interesting to explore how fast this conclusion could be reached. For that, the last classification tree is applied to the values of the first problem, then the average of the first and the second, then to the average of the first, the second and the third, and similarly until all the values for all the problems are averaged. The results, presented in Figure 5, suggest that as early as the 4th problem a high level of correct classification is reached. Also the percentage of correct classification is maintained and plateau at the final value around the 12th problem.

Table 3: Non-normalized features used in the tree classification

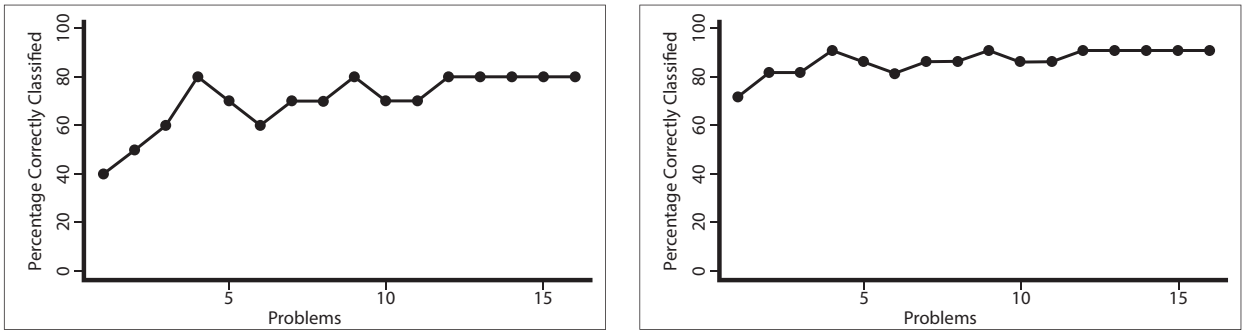
Feature	Normalized	Method
PCU	MC	Highest value
DHT	MMO/LMO	Highest / Lowest value
$SD-DHT$	LMV	Lowest value
TM	MM	Highest value
NOI	MI	Highest value
TSD	MSD	Highest value
TNM	MN	Highest value
$TMTM$	MM	Highest value
TCP	MCP	Highest value
TNS	MS	Highest value
$ASPL$	SS	Lowest value
ANP	LP	Lowest value
ASD	MD	Highest value
$ASTL$	FW	Lowest value
ASP	MP	Highest value

Table 4: Classification tree splits with normalized and non-normalized features

Variable	Value for Experts	Discrimination Power
FW	> 0.5	6.53
LP	> 34.74	6.53
PCU	> 38.05	4.44
MN	> 0.13	4.03
PNM	> 6.25	3.19

5. DISCUSSION

Based upon the results previously presented, it is evident that variables such as the fastest student in a group, the percentage the calculator is used, and the times numbers are mentioned while solving a problem, are key variables to estimate whether a student is able to solve a problem or not. These results might suggest that how fast the student writes is maybe an indicator of how certain the student is about how to solve a problem. Even though, this variable was not initially used as part of the predictor variables, its inclusion in the model made sense in the process. Another critical variable predicting success in solving problems is the percentage of using the calculator in a group. It is perhaps natural, that a young student that knows how to solve a problem, does not let others to use a tool that allows him or her to succeed. The regression analysis also shed light as to understand how interactions or interventions in a group are related to the probability of solving a problem. On the contrary of what might be commonly thought, students with few interventions, mentioning numbers are more likely to correctly solve problems. Again, someone that knows the solution to a math problem (an expert) probably does not speak that much, but his or her intervention is used to indicate a focused and precise affirmation about numbers and the way to solve a given problem. The times a student mentioned commands when solving problems is a very specific variable of the sessions recorded; it is definitely related to the leader of the group interacting with the computer sys-



(a) Evolution of Expert Classification

(b) Evolution of Non-Expert Classification

Figure 5: Evolution of the classification results for expert and non-expert students using a variable number of problems.

tem. However, an expert that solves problems could take the role of leader when indicated the computer the solution to a problem or prompted for a new one. Surprisingly, other variables, such as the type of strokes a student draws, total movement when engaged in solving a problem, or the proximity to the center of the table while collaborating in a solution, were features that did not contribute in the identification of an expert or a successful problem solver. The positive predictor variables were able to determine with a 63% probability when a student would solve the problem correctly with only 30% of the non-correctly-solving cases, incorrectly classified as positive. This is much higher than selection by chance.

The analysis to determine if a student is or not an expert, predictably found similar factors that those for solving a problem correctly, given that solving a problem correctly is an indicator of expertise. However, when the individual problems are averaged and the variables are normalized through a binarization, the noise in the data caused by spurious errors and distractions is reduced, increasing the level of correct classification. The 80% of the experts are found with just 10% of non-experts being incorrectly classified. Again, these numbers are much higher than selection by chance.

The fact that the same variables are highly discriminant at the problem and session level, made it possible to determine the expertise of the students, with a level of precision above random choice, from the first approach. This is shown in the Expertise Estimation analysis, where the correct classification level stabilized when enough problems (in this case four) are averaged.

A positive aspect to underline is that the answering of the research question took two methodological approaches that resulted in a good confirmatory analysis.

It is important to note that the results obtained are difficult to generalize due to the low number of recordings available, especially those related to incorrectly solved problems (more than 50% of the cases) and the ones used for Expert Estimation (averaged by session). While it is improbable that the results found are due to chance, this low number of records, made impossible to assert that the levels of prediction will stay the same if applied to a larger dataset. Moreover, the use of very specific features, such as the use of a calculator and times numbers and mathematical terms were mentioned, impedes the findings to be applied to other settings. However, the pen-based measurement of the speed of writing, could be found also significant to determine expertise, or at least, confidence in other types of problems.

6. RELATED WORK

Several studies are closely related to leadership, dominance or group collaboration (See [22]), but few are focused on determining the level of expertise. The work of Scherer et al. [3], similarly to the approach followed in the present study, researched features more based on what a human could superficially hear in audios and interpret from drawings and found that the Peak Slope of the audio could discriminate between Experts and Non-Experts. Some studies suggest that it is possible to build models that can make predictions of the user's level of knowledge or expertise, based on real-time measurements of eye movement patterns during a task session [23]. In this line, distinguishing levels of expertise, based on features gathered from video while building solutions based on physics knowledge, was studied by Worsley and Blikstein [24]; their research concluded that two-handed interaction is positively correlated to expertise. The interaction of both hands when solving problem is related to the interaction of both hemispheres of the brain; thus an expert is more likely to show this type of interaction when solving problems. Conversely, [25] states that expertise depends heavily on implicit pattern recognition and selective extraction, which are skills acquired through perceptual learning. Thus, experts might have absences of verbalization or no interaction (no movement), when they used the "mind as pattern recognizer". As for the way experts and novices represent solutions, the main difference is related to poorly defined or qualitatively different representations on the side of the novice. The experts quickly establish correspondence between external events and internal models of these events, which in turn correlates to the quick establishment/writing of a solution [26]. Therefore, the fast representation of solutions seems to be related to expertise level. The present study also investigated the use of calculators while solving problems and its relation to expertise. The use of tools in learning environments was explored by [27], they studied the variation in tool use and its relation to prior knowledge and goal orientation and how this variation affects performance. They found that there were no differences in performance regardless the frequency of tools usage. Even more, prior knowledge was not either related to tool use.

7. CONCLUSIONS AND FUTURE WORK

The main conclusion of this work is that simple features, derived from multimodal recordings of working sessions are able to discriminate with a high degree of success experts from non-experts in mathematics problem-solving sessions. While a perfect classification currently requires semantic understanding of the recordings, good enough levels (80%) can be obtained with features extracted with automatic algorithms that barely touch the semantics of what is being recorded. Also important to mention is that these algorithms are widely available and are not specifically made or tailored to the given problem.

The way in which the features were initially selected also provides a conclusion that could be taken to other works in the field. Some of what resulted relevant features were selected based on common sense. For example, it was clear to the authors that whoever uses the calculator is bound, in more cases than not, to be the one solving the problem. The same with the pronunciation of numbers and mathematical terms. However, the most discriminant variable, the speed of writing, was initially added only because the data was readily available, but the authors did not considered it to be suitable. After the results were obtained, it became clear that the variable was measuring something relevant, the authors hypothesize that it is the confidence of knowing how to solve the problem. It is proposed that both techniques, purposeful search or calculation of features, together with recording whatever is recordable, should be used in learning analytics in general.

One important conclusion for the field of multimodal learning analytics is that good predictors were found in each one of the media analyzed. Video contributed the use of the calculator, audio contributed the number of times the student talk about numbers and the pen-recording provided information about the speed of writing. Each feature seems to be measuring an independent factor in the process of mathematical problem solving. While the analysis made in this work were just to determine statistical significance, a larger effort, paired with a larger dataset with more media, for example biometrics, could uncover a more general model of how to determine expertise in problem-solving.

Finally, the authors want to stress the importance of openly available datasets in the progress of the learning analytics field. The ability of objectively compare different algorithms and models between researchers is what could lead Learning Analytics to not only propose good ideas, but to test them and standardize them for their easy transfer to practitioners. Without a common testing ground, Learning Analytics risks becoming an art and not a science.

8. REFERENCES

[1] P. Blikstein, "Multimodal learning analytics," in *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pp. 102–106, ACM, 2013.

[2] M. Worsley, "Multimodal learning analytics: enabling the future of learning through multimodal data analysis and interfaces," in *Proceedings of the 14th ACM international conference on Multimodal interaction*, pp. 353–356, ACM, 2012.

[3] S. Scherer, N. Weibel, L.-P. Morency, and S. Oviatt, "Multimodal prediction of expertise and leadership in learning groups," in *Proceedings of the 1st International Workshop on Multimodal Learning Analytics*, MLA '12, p. 1:8, ACM, 2012.

[4] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm Computing Surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.

[5] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," in *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, pp. 8–14, IEEE, 1998.

[6] T. Hammond and R. Davis, "Tahuti: A geometrical sketch recognition system for uml class diagrams," in *ACM SIGGRAPH 2006 Courses*, p. 25, ACM, 2006.

[7] S. Oviatt, A. Cohen, and N. Weibel, "Multimodal learning analytics: Description of math data corpus for icmi grand challenge workshop," *Second International Workshop on Multimodal Learning Analytics*, December 2013.

[8] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: speeded up robust features," in *Computer Vision - ECCV 2006*, no. 3951 in Lecture Notes in Computer Science, pp. 404–417, Springer Berlin Heidelberg, Jan. 2006.

[9] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[10] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *In VISAPP International Conference on Computer Vision Theory and Applications*, pp. 331–340, 2009.

[11] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, pp. 172–185, June 2005.

[12] G. R. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*. Oreilly & Associates Incorporated, Mar. 2013.

[13] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[14] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints," *Conference on Computer Vision and Pattern Recognition*, 2010.

[15] C. Chelba, D. Bikel, M. Shugrina, P. Nguyen, and S. Kumar, "Large scale language modeling in automatic speech recognition," tech. rep., Google, 2012.

[16] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope, ""your word is my command": Google search by voice: A case study," in *Advances in Speech Recognition*, pp. 61–90, Springer, 2010.

[17] J. B. Lovins, *Development of a Stemming Algorithm*. M.I.T. Information Processing Group, Electronic Systems Laboratory, 1968.

[18] B. Paulson and T. Hammond, "Paleosketch: accurate primitive sketch recognition and beautification," in *Proceedings of the 13th international conference on Intelligent user interfaces*, IUI '08, pp. 1–10, ACM, 2008.

[19] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees. wadsworth & brooks," *Monterey, CA*, 1984.

[20] T. Therneau, B. Atkinson, and B. Ripley, *rpart: Recursive Partitioning*, 2013. R package version 4.1-1.

[21] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[22] D. Sanchez-Cortes, O. Aran, M. S. Mast, and D. Gatica-Perez, "A nonverbal behavior approach to identify emergent leaders in small groups," *Multimedia, IEEE Transactions on*, vol. 14, no. 3, pp. 816–832, 2012.

[23] M. J. Cole, J. Gwizdka, C. Liu, N. J. Belkin, and X. Zhang, "Inferring user knowledge level from eye movement patterns," *Information Processing & Management*, 2012.

[24] M. Worsley and P. Blikstein, "Towards the development of multimodal action based assessment," in *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pp. 94–101, ACM, 2013.

[25] P. J. Kellman and P. Garrigan, "Perceptual learning and human expertise," *Physics of life reviews*, vol. 6, no. 2, pp. 53–84, 2009.

[26] M. T. Chi, P. J. Feltovich, and R. Glaser, "Categorization and representation of physics problems by experts and novices," *Cognitive science*, vol. 5, no. 2, pp. 121–152, 1981.

[27] L. Jiang, J. Elen, and G. Clarebout, "The relationships between learner variables, tool-usage behaviour and performance," *Computers in Human Behavior*, vol. 25, no. 2, pp. 501–509, 2009.