Sensorless Affect Detectors on Digital Tutoring Systems

Shamya Karumbaiah
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01002
shamya@cs.umass.edu

Rafael Lizarralde
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01002
rezecib@cs.umass.edu

1 Introduction

An intelligent tutoring system is a piece of software that aims to deliver differential teaching. The teaching response is adapted to student needs after reasoning about domain knowledge, student mastery, and pedagogy. Many studies suggest that an emotional state of a student interacts with his/her engagement with the system and learning [8]. Affect detection is an important step towards improving student engagement in such a system. An accurate detector of negative emotional states would allow for a variety of improved responses in tutors, such as interventions for disengagement repair. The majority of the intelligent tutors do not use affect at all, and studies on affect-based interventions have typically relied on self-reports from the students, which runs the risk of annoying the student. Most of the work on automatic affect detection relies on the data from the different sensors installed in an experiment. Practically, it would be unreasonable to deploy most of these sensors in an actual classroom. Recently, there has been some work on sensor-less affect prediction. The student interaction logs provides insights for a easily adaptable, sensor-independent method of affect detection.

This is a classification problem. Given the interaction log of a student, we want a model that can predict whether they have a particular affect (are feeling specific relevant emotions). The emotional states of interest are - concentrating, confused, bored and frustrated. A successful model will most likely require a short history of student interactions to make accurate predictions. An interpretable model may be useful, depending on the types of interventions that might be desired in the tutoring systems, although many interventions could be done without an interpretable model. We dont need an extremely fast model, as predictions can only be made after student actions, and these are at a rate of around one action per student per 10 seconds, with no more than a few thousand students.

Our final solution involved the selection of 57 features, including several of the affect history features we added. Using a random forest with 40 trees, it scored a Cohen's Kappa of 0.306, and scored 0.230 on the test set. The cross-validation score is significantly better than the cross-validation score reported by Wang et al. [7], but the test score suggests that the data set desperately needs expansion.

2 Related Work

Towards an automated affect detection, there has been significant research on the use of sensors for affect detection. Usage of physiological sensors have been prevalent and have lead to some successful detectors for a set of emotions [9]. A similar approach [10] has produced detectors using vocal patterns in the interaction between the student and the tutor. Facial expressions and body language are yet other features used for affect detection [11]. A real world experiment with these detectors were conducted in an urban school and demonstrated a successful detection as compared with the student self reports [12]. The major issue with most of these sensor-based approaches is their dependency on the sensors which restricts their usage by the infrastructure availability, cost,

and set-up. Therefore, the ideal scenario would be to automate affect detection with the data already accessible directly from within the software of intelligent tutoring systems. Towards this, the most successful affect detector is the one built on the cognitive algebra tutor [2]. It looks solely at the log data on student usage in the system.

Most of the current work on sensor-less affect detection concentrates mainly on feature engineering. Wang et al. [7] worked on the same ASSISTment data as used by this project; they try to add missing skill tags (an ostensibly important feature) and find that these additions do not improve the performance much. Their base results form a basis of comparison for our models. Baker (2007) [2] and Wixon et al. [6] use simplified tools like RapidMiner for developing models. Paquette et al. [5] compared the sensor-less method to a sensor-based method in their vMedic tutor and observed that the interaction-based detectors outperform posture-based detectors for their tutor population. Bosch (2015) [3] talks about the generalizability of the model across demographics and time. Several researchers we spoke to expressed pessimism about developing accurate affect detectors, but the literature suggests that only a limited range of models and optimizations have been tried. Most of these sensor free detectors are only slightly better than chance. There is a clear need for an improved machine learning pipeline to model these detectors efficiently, using more sophisticated models and pipeline optimization techniques. In this project, we have concentrated mainly on customizing different models to fit better to this problem domain.

3 Data Sets

We explored the data sets from two different online tutoring systems that cover elementary to high school level mathematics topics: ASSISTments and MathSpring. Both data sets consist of logs of student actions in the system, including some observations of student emotional state (”affect”, such as boredom, frustration, interest). The ASSISTments data set has affect annotated with the BROMP 1.0 protocol [4]. Features include things such as the number of hints used, problem topic, and time spent on problems. A more detailed description of the features can be found on the ASSISTments website [1].

There were several different copies of this data set in various levels of agreement. The one we found the most usable had separated the data into sets for each of the four affects being studied (boredom, concentration, confusion, and frustration), and resampled each of them to produced similar numbers of data cases for the presence and absence of each affect. The boredom data set was missing several features, so to even out the data set we removed these from the other sets first. This left us with 3019 unique data cases and 188 features. Each of these 188 features are one of four aggregations (sum, average, min, and max) of 47 fundamental features described above. These features were aggregated over clips of rows up to and including where affect annotations were hand-coded by human observers.

The second tutor, MathSpring has similar sparse annotation of affect, but this is obtained by a pop-up question that asks a student how they are feeling with respect to a particular emotion (e.g. are you feeling somewhat frustrated, very frustrated, or not at all). The raw data was extracted from the tutors problem history log. The system has evolved over time and same is with the data collection in the relational tables. A majority of the tables violated the foreign key constraint while referencing another table. This lack of referential integrity made joining the tables to get meaningful data strenuous. Another major challenge with this data was that most of it was generated while conducting experiments, while the developers were testing the system, or from guest users trying to use the system. There are no real deployments of this system in classrooms. Hence, we had to discard this data.

Another alternative we tried is to use the data from the Cognitive Algebra Tutor [13]. This data set had very similar research done [14], however the only data set for it we were able to find was very unprocessed (for example, each data row contained an action the student made in the system, while a model would need secondary features processed from these, such as the number of hints used recently), and there were insufficient descriptions of the feature engineering processes that resulted in features useful for a machine learning algorithm.

4 Proposed Solution

Our first step in processing the data was to combine all four affects into a single data set. Although Wang et al. [7] developed separate affect detectors for each affect, the labeling itself was produced by the BROMP protocol [4], which is explicitly multiclass; at each labeling step the coder labels with the first affect they can identify, or writes that they were unable to identify one. This means that in addition to trying to predict the four affects they examined, we also have a class for unknown affect, which could include emotions like eureka, delight, and surprise. It also includes cases where it was impossible to code affect, such as when a student went to the bathroom or the software crashed.[14] Combining the data involved some rearrangement of columns and removal of a few features that were not present in the boredom data set, which was done manually in Excel. We have included this minimally-processed data set and the scripts used for all further processing. To combine these data sets, we needed to use the unique row identifiers to remove duplicates and match the rows between the different files. Fortunately, there were no ambiguities and each unique row had only been labeled as having one affect present in any of the data sets, so we could unambiguously convert the data into a five-class labeling: unknown, bored, concentrating, confused, and frustrated.

Our next data processing step involved attempting to capture some of the time-series nature of the domain (presumably each affect is influenced by previous affects). First, we had to separate the data into the sessions in which students were using the system, and within sessions, separate rows by individual students, ordering the rows by the time at which they were labeled. We then added five features, one for each of the classes. Each feature represents a weighted average of whether the affect was present or not in previous rows for that student in that session. Initially, all features are set to zero. For each row in the sequence, the values are an average of the previous row's value and 1, if that affect was the last recorded affect (for the first row this is assumed to be "unknown"). This results in an exponential decay of the value if the affect has not been seen recently, while a repeated affect causes the value to approach 1.

The scoring system used in [7][14] were Cohens Kappa and A. Cohens Kappa indicates the degree to which the detector is better than chance at identifying which data set involve a specific affective state. It takes a value between -1 to 1 with 0 indicating the performance at chance; a value closer to 1 is desired. A gives the probability that the algorithm will correctly identify whether a specific affective state is present or absent in a timestamp. This is stated to be equivalent to the area under ROC curve (auc_roc). A model with an auc_roc value of 0.5 performs at chance. A value closer to 1 is desired. However, this is defined for single class or multilabel classification only, we have assessed our model by modifying it to be single class detector for each affective state. We found that the changes in accuracy and F1 scores to be proportional to the improvement in these scores. Hence, to be consistent with the model comparison, we have chosen the same scoring scheme.

Table 1: Distribution of affect labels

Affect	**Frequency (out of 3019)**
Unknown	386
Bored	325
Frustrated	94
Confused	97
Concentrating	2117

As mentioned before, we dont need an extremely fast model, as predictions can only be made after student actions, and these are at a rate of around one action per student per 10 seconds, with no more than a few thousand students. We fit some models to compare their performances (results in the next section) and random forests seemed to perform the best. Considering the small size of training data, a neural network was not considered. Also, as we see in the Table 1, the negative emotions are relatively few than the affective state of concentrating. With such skewed classes, an ensemble like random forest was expected to perform well.

We used random forests, combined with forward stepwise feature selection. The primary hyper-parameter of random forests are the number of decision tree estimators it uses, so we select that first, and then do a smaller search of the other hyperparameters. Due to certain affects being under-

represented in the data (notably confused and frustrated, which each have only around 90 data cases), in order to get accurate cross-validation assessment, we need to use a stratified split. We chose stratified shuffle splits due to the greater flexibility of changing the number of splits without significantly changing the quality of each split's training. We have held out 10% data as test set and is untouched until the model's generalizability is tested.

We also investigated dimensionality reduction, exponentially weighted moving averages, basis expansion and resampling schemes, but as we will show later on, these turned out to be dead ends. We had also planned to try to reproduce several papers' results more closely with RapidMiner, but the versions they used are no longer available and the model files they provided are not recognized by available versions of RapidMiner.

5 Experiments and Results

Intuitively random forests were the perfect fit to the problem. To verify our assumption, we fit a set of basic classifiers on the data. Table 2 gives the list of these models and their Cohen's kappa. As expected, random forests had the best score.

Table 2: Classifier performance as measured with Cohen's Kappa score

Classifier	Cohen's Kappa
Random Forest	0.205
Logistic Regression	0.006
Decision Tree	0.193
KNN	0.130
Naive Bayes	0.002
LDA	0.197
RBF SVM	0.024

Due to several of the papers [14] [7] using resampling as a method of balancing the skewed classes in the data set, we initially investigated some resampling schemes to try to improve our results. However, after some preposterously good-looking cross-validation results, we realized that heavy resampling that occurs independent of splitting meant that every split was likely to have the same data cases present in both the training and the test sets. To rectify this, we implemented a `ResampledKFold` splitter that first splits, then resamples within each side of the split to balance out the classes. However, using this approach had little to no effect on the results, so we reverted to `StratifiedShuffleSplit`.

We also wanted to capture time-series aspects of the domain, so we tried adding an exponentially-weighted moving average of the labels as a feature (the idea being that this approximates a history of labels assigned by a relatively good classifier). This also produced very good results, although they were not quite as suspicious as the initial resampling results. However, we eventually realized that we were inadvertently including the current row's label in the moving average, and in effect providing the partial label directly in the features. Fixing this, we found that it had virtually no effect on the results.

Table 3: Performance of current detectors as measured with Cohen's Kappa score [14]

Classifier	A'	Cohen's Kappa
Concentrating	0.731	0.417
Confusion	0.625	0.146
Frustration	0.597	0.151
Boredom	0.662	0.243
Average	0.654	0.239

We were able to effectively capture some of the time-series nature of affect with the affect history features we described earlier, which are similar to an exponentially-weighted moving average, but were confined to individual students in individual sessions. The default model with the merged data (without the affect history features) scores a Cohen's Kappa of 0.226. After we added the affect history features, this was improved to 0.254. This provides a good baseline for the remaining experiments. Table 4 gives the scores for the current detectors. Note that these are cross validation scores and the paper doesn't mention about leaving a test set out. So, the generalizability of this model is questionable.

Principal component analysis (PCA) had very lackluster results. All numbers of components extracted by PCA floated around a Kappa of 0.200 or worse.

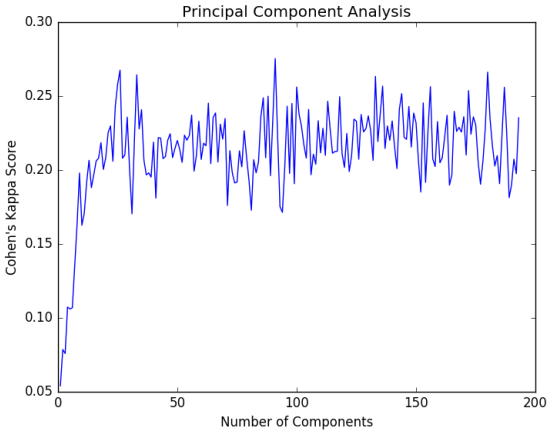

Figure 1: Principal component analysis had poor results for low numbers of components, and stabilized to normal performance without PCA.

Feature selection produced quite interesting results (Figure 2 and Table 4), with substantial performance improvements and pretty incredible performance even with only a few features selected. Additionally, the several of the affect history features we added were selected very early on, highlighting their importance in predicting affect. Overall, the best results were achieved with 57 features, the top ten of which we have listed below.

Table 4: Top features selected by forward stepwise feature selection. We added bold features.

Feature Name
BoredHistory
UnknownAffectHistory
ConcentratingHistory
MaxtotalFrPercentPastWrong
SumtimeGreater10AndPrevActionWrong
AveragefrIsHelpRequestScaffolding
AveragetimeSinceSkill
MaxtimeGreater10AndPrevActionWrong
MintotalFrPastWrongCount
SumtimeGreater10SecAndNextActionRight

Hyperparameter selection selected a surprisingly low number of trees (Figure 3), choosing only 40 trees. However, we can see that the results of the number of trees are quite unpredictable, so this likely depends heavily on the particular train/test split.

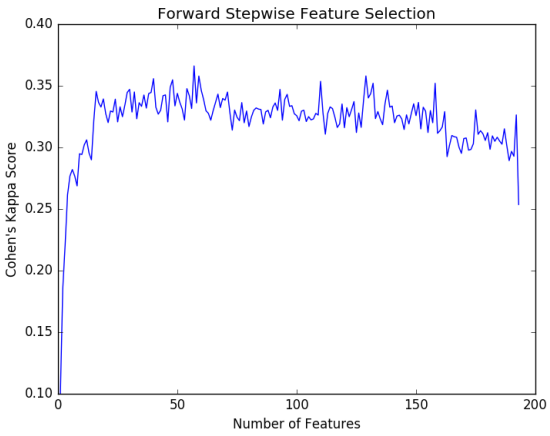

Figure 2: Feature selection resulted in modestly improved performance, although a surprisingly small number of features were required to achieve good performance.

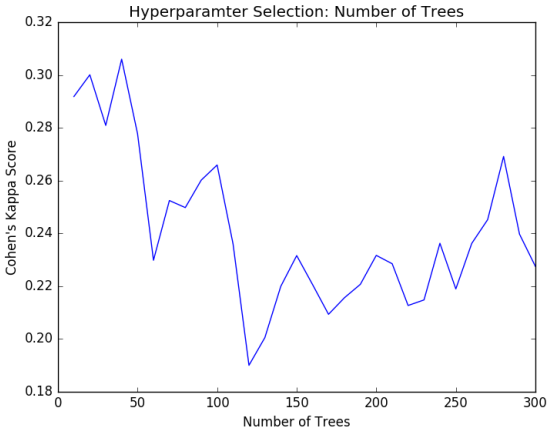

Figure 3: Hyperparameter selection for random forest, selecting the number of trees.

The final pipeline scored a Cohen's Kappa of 0.306 through cross-validation, and a Kappa of 0.230 on the test set.

6 Discussion and Conclusion

In this project, we have tried to use interaction logs of an intelligent tutoring system (ASSISTments) to detect the emotional state of a student. We have stated it as a multiclass classification problem predicting a single affect among the five possible states for each row in the log. The final model works moderately well; better than chance with a Cohens Kappa of 0.306. We see an overall improvement in the scores from the baseline. By cross-validation, we saw a 28% improvement in the Cohens Kappa score from the best performing model currently known on this data (which was also assessed by only cross-validation). However, the test result suggests that the data set is far too small, as it scored only 0.230, just under the current best result of 0.239. This is an indication that a customized machine learning pipeline could improve the sensor less affect detection. We hope that a better affect detection technique of this sort would aid in choosing better interventions to improve student engagement.

In our future work, we would like to try some time series models as this problem seems like a good fit for time series analysis. Secondly, it would be good to contemplate on the possibility of a student having multiple emotions at the same time. Formulating this problem as mutli label classification would need a new experimental design for data collection as the current BROMP 1.0 doesnt have a provision for this. We arent considering demographics information or student performance (as pre/post test scores) as yet. It would be interesting to add features like gender, age, socio-economic background, ethnicity, etc to reason about a particular behaviour of a student. Also, explore the correlation of a students performance measure with his affect.

References

[1] ASSISTmentsData. Accessed on March 24, 2016. https://sites.google.com/site/assistmentsdata/home/2012-13-school-data-with-affect

[2] R.S.J.d. Baker. Modeling and Understanding Students' Off-Task Behavior in Intelligent Tutoring Systems. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1059-1068, 2007. ACM.

[3] N. Bosch. Multimodal Affect Detection in the Wild: Accuracy, Availability, and Generalizability. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 645-649, 2015. ACM. http://pnigel.com/papers/bosch-dc-icmi15-camera.pdf

[4] J. Ocumpaugh, R.S.J.d Baker, and M.M.T. Rodrigo. Baker-Rodrigo Observation Method Protocol (BROMP) 1.0. Training Manual version 1.0. Technical Report, 2012. http://www.academia.edu/2282729/Baker-Rodrigo_Observation_Method_Protocol_BROMP_1.0_Training_Manual_version_1.0_

[5] L. Paquette, B. Mott, K. Brawner, J. Rowe, J. Lester, R. Sottilare, R.S.J.d Baker, J. Defalco, V. Georgoulas. Sensor-Free or Sensor-Full: A Comparison of Data Modalities in Multi-Channel Affect Detection. Under review for the *8th International Conference on Educational Data Mining*. http://www.columbia.edu/~rsb2162/2015paper150.pdf

[6] M. Wixon, I. Arroyo, K. Muldner, W. Burleson, C. Lozano, and B. Woolf. The Opportunities and Limitations of Scaling Up Sensor-Free Affect Detection. In *Proceedings of the 7th International Conference on Educational Data Mining*, pages 145-152, 2014. http://educationaldatamining.org/EDM2014/uploads/procs2014/long%20papers/145_EDM-2014-Full.pdf

[7] Y. Wang, N.T. Heffernan, C. Heffernan. Towards better affect detectors: effect of missing skills, class features and common wrong answers. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 31-35, 2015. ACM.

[8] Dragon, T., Arroyo, I., Woolf, B.P., Burleson, W., El Kaliouby, R., and Eydgahi, H. Viewing Student Affect and Learning through Classroom Observation and Physical Sensors. In *International Conference on Intelligent Tutoring Systems*, pp. 29-39, 2008.

[9] Calvo, R.A., and DMello, S.K. eds. Viewing Student Affect and Learning through Classroom Observation and Physical Sensors. In *New Perspectives on Affect and learning technologies*, 2011, New York: Springer.

[10] Litman, D.J., and Forbes-Riley, K. Recognizing Student Emotions and Attitudes on the Basis of Utterances in Spoken Tutoring Dialogues with both Human and Computer Tutors. In *Speech Communication*, 48 (5), pp. 559-590, 2006.

[11] DMello, S.K., and Graesser, A.C. Multimodal semi- automated affect detection from conversational cues, gross body language, and facial features. In *User Modeling and User- adapted Interaction.*, 20 (2), pp. 147-187, 2010.

[12] Arroyo, I., Woolf, B.P., Cooper, D., Burleson, W., Muldner, K., and Christopherson, R. Emotion Sensors Go To School. In *14th International Conference on Artificial Intelligence In Education*, 2009.

[13] PLSC. Accessed on April 1, 2016. https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=479

[14] R.S.J.d Baker et al. Towards Sensor-Free Affect Detection in Cognitive Tutor Algebra. In *5th International Conference on Educational Data Mining*, pages 126-133, 2012.