Chang, M. et al. (Eds.) (2019). Proceedings of the 27th International Conference on Computers in Education. Taiwan: Asia-Pacific Society for Computers in Education

Classification of Emotions in Programming from Face and Log Features Using Representative Intervals

Thomas James TIAM-LEE* & Kaoru SUMI
Future University Hakodate, Japan
*g3117002@fun.ac.jp

Abstract: This paper discusses a machine learning approach for classifying student emotions while doing programming exercises. Detection of academic emotions in programming from face features has previously been shown to be a difficult task because people don't tend to display as much expression as compared to more social activities. In our approach, we show that adding log features in addition to face features can improve the performance of classifiers. Furthermore, we show that identifying representative intervals of each emotion type based on human annotations can be used to build models to classify emotion over longer periods of time. We believe that our study can contribute in the development of better intelligent programming tutors that can respond to the affective state of students.

Keywords: Affective computing, Student modelling, Education

1. Introduction

Recent advances in artificial intelligence have allowed for intelligent tutoring systems to respond to the affective states of students. Many of these so-called ``affective tutoring systems" have been motivated by several studies that correlate the emotional experience of students to their achievement (Rodrigo et al., 2009; Daniels et al., 2009) and self-regulated learning (Mega, Ronconi, & De Beni, 2014; Cho, & Heron, 2015). The ability to observe and recognize the student's affective states is part of why human tutors are very effective (Lehman et al., 2008). Likewise, intelligent tutoring systems that can respond to student emotions have been shown to yield a variety of benefits in the students' learning (Malekdazeh, Mustafa, & Lahsasna, 2015).

A key part of these types of systems is the automatic detection of student affect. A comprehensive meta-analysis of various approaches for affect detection can be found in the work of D'Mello and Kory (D'Mello, & Kory, 2015). A number of these studies used cameras are used to extract facial features to detect basic emotions such as joy, sadness, and anger (Zatarain-Cabada et al., 2015). However, in addition to these basic emotions, there has also been an interest in detecting what are referred to as "academic emotions" (Pekrun et al., 2002). These are emotions that are prevalent in academic settings, such as engagement and boredom. In AutoTutor (D'Mello, & Graesser, 2012), the student learns by interacting with a virtual conversational tutor avatar while an affect detector module uses a combination of posture, face, and log data to infer academic emotions. In VALERIE (Palearie et al., 2005), academic emotions are detected using a wide array of sensors including a video camera, a speech recognizer, and a psychological signal sensor.

However, there are some domains in which detection of affect has remained difficult. One of these is the domain of programming. Intelligent programming tutors (IPT) are a subclass of intelligent tutoring systems that teach programming. IPTs pose challenges that are not present in traditional ITSs (Crow, Luxton-Reilly, & Wuensche, et al., 2018). In these systems, students spend the majority of their time writing, testing, and debugging code instead of interacting with a virtual agent. Despite this, previous studies have shown that students frequently transition between different academic emotions while taking part in this kind of activity (Bosch, D'Mello, & Mills., 2013; Bosch, & D'Mello, 2013). However, the displays of affect are more naturalistic and difficult to detect.

Studies on detection of affect in programming are relatively limited. In a series of studies (Grafsgaard et al., 2011; Grafsgaard et al., 2013a; Grafsgaard et al., 2013b), displays of affect are detected using posture, gesture, and facial features, but are also induced by the nature of the experimental setup which involved a human tutor communicating with the student through an interface. An attempt to classify emotional states in a purely-coding setup was done by Bosch, Chen, and D'Mello by training classifier models using face features captured from a video camera (2014). They found that fine-grained and fixed judgment emotions was difficult, with the classifiers failing to perform above chance. They achieved better results for spontaneous judgments on confusion and frustration. More recently, negative affect was detected from students' keyboard and mouse actions in programming (Vea & Rodrigo, 2016).

Despite its difficulty, there is value in being able to infer emotions of students while doing programming tasks. In our previous studies, we have shown that a system that can provide adaptive feedback when confusion is detected in the context of programming can potentially help students solve more problems in programming practice (Tiam-Lee & Sumi, 2017, 2018a, 2018b). However, the model in this study was trained on a small dataset and was not rigorously evaluated for its accuracy. Expanding on this idea, we performed analysis on a larger dataset and found that both face features and log features can be meaningful in detecting emotions in programming (Tiam-Lee & Sumi, 2019). In this study, we continue to explore this idea by showing that combining log features with face features can improve the detection of emotions on fixed point judgment intervals. Furthermore, we show that identifying representative intervals of each emotion can potentially improve the classification of emotion over longer intervals, an insight that can be useful in building better emotion detectors in programming.

2. Methodology

2.1 Data Collection

We collected data from students by asking them to participate in a simulated programming session in which they must solve a series of coding exercises. In each exercise, they must write the body of a function according to a given specification. For example, in one of the exercises the function should return the area of a square given the length of its side as an argument. A custom application was used for this task which automatically logged all code changes and compilations. It also recorded a video of the student's face. Figure 1 shows the setup of the data collection.

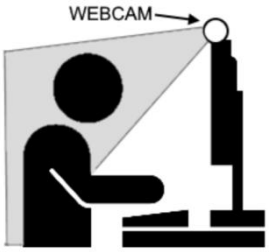

Figure 1. Setup of the programming session.

Each student participated for 45 minutes or until all exercises were solved. After this, the student was shown replays of different time intervals of the session. These were selected based on key moments such as compilations, and when the student starts or stops typing. For each time interval, the student was shown the video of his face and the snapshot of his or her code at that time. Based on this, the student was asked to provide an emotion label which describes best what he or she felt during that time interval. The options were "engaged", "confused", "frustrated", "bored", and "neutral" (no apparent emotion). This is based on a previous study which identified the common emotions that are prevalent in programming tasks (Bosch, D'Mello, & Mills., 2013). To minimize subjectivity, we provided the definitions for each emotion, shown in Table 1. Our methodology produced self-report affect judgments that are fixed (intervals were chosen by the system, and not by the student).

A total of 73 students from Japan and the Philippines participated in the study. All students were enrolled in a freshman introductory programming course at the time of the study. All participants signed informed consent forms which contained and explanation about the study, the data to be

collected, and how it will be used. After removing bad videos (e.g., face not captured properly, face not detected well), we were left with session data from 39 students.

Table 1

Definition of Affective State Labels

Emotion	Definition
Engaged	You are immersed in the activity and enjoying it.
Confused	You have feelings of uncertainty on how to proceed.
Frustrated	You have strong feelings of anger or disappointment.
Bored	You feel a lack of interest in continuing with the activity.
Neutral	There's no apparent feeling.

2.2 Data Processing

We extract log features by taking the number of code insertions, code deletions, compilations, and submissions. We also consider if the compilation was successful or not (e.g., syntax or runtime error), and if the submission was passed all test cases or failed some of them.

To extract face features from the videos, we used OpenFace, a toolkit which can estimate facial landmarks and detect the presence of action units from images and videos using state-of-the-art machine learning techniques (Baltrusaitis, Zadeh, & Lim, 2018). Figure 2 shows an example of the estimation of facial landmarks. Action units (AUs) are a taxonomy of fundamental actions of facial muscles, some examples of which are opening the mouth or raising the eyebrows (Ekman and Friesen, 2003). We processed each video per frame and extracted the head pose features, which included the x, y, and z positions of the head in 3-D space and the x, y, and z rotation of the head. We also extracted the intensity of 17 AUs that OpenFace can detect. The intensity is a numerical value representing how strongly the AU was displayed in the frame.

We merged the consecutive intervals with the same reported emotion, resulting in intervals of widely varying lengths. An instance representing a time interval of the session is constructed by aggregating all observations within the time interval. First, the head pose and AU features were z-standardized on a student level to account for individual differences. We then take the sum of each log feature and the median, maximum, and standard deviation of all face features across the interval. We also perform some data cleaning by disregarding frames where OpenFace failed to detect the face properly using rule-based conditions.

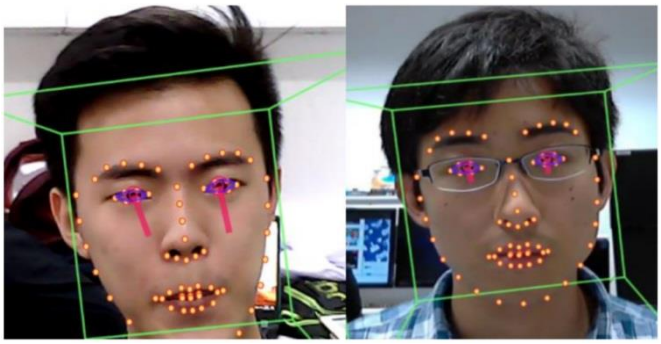

Figure 2. Estimation of facial landmarks by OpenFace.

2.3 Model Training

We trained machine learning models to classify affect on fixed-length, fine-grained sub-intervals. To do this, we used the caret package in R. We did not consider those labelled “neutral”, as we were interested in detecting the type of emotion, if it is present. We obtained the sub-intervals by splitting each interval into fixed-length sub-intervals, with a sliding window of $L/2$, where L is the length of the sub-interval. We trained models using 10, 20 and 60 second fixed-length sub-intervals to compare the performance of the classifiers across different L values. The L values were selected manually to test the ideal length of the interval for detection of affect.

To prevent overfitting and to ensure that the models can generalize to other unknown students, we used a student-level cross-fold validation. We randomly divided the students into 3 sets (each set contained 13 students), and used 2 sets for training and 1 set for evaluation in each fold. We also repeated this process 10 times to account for random sampling error. We built a separate classifier for each academic affective state, and tried different classifier models, namely: C4.5 Decision Tree, Naive Bayes, Logistic Regression, Support Vector Machine, and Multilayer Perceptron. We selected the best fit performing model for each. Our baseline is using face features only, as was presented in previous studies on emotion detection in programming, so we first, we trained models using face features only, and then compared the performance against models that were trained with both face and log features to see if the addition of log features can improve performance.

2.4 Models for Classification of Longer Intervals

We also performed detection of affect over the original intervals prior to splitting. These intervals varied widely in terms of the length. To perform the classification, we trained hidden Markov models for each of the affective state labels based on the sequence of classifications produced by the localized classifiers, i.e. we combine the prediction output of each of the affective state classifiers for each sub-interval and used it to train a discrete state hidden Markov model for each affective state.

To classify the affective state of an unknown interval, first we divide it into several fixed-length sub-intervals of length L, with a sliding window of L/2. Then, we use the localized affect classifiers to classify each sub-interval for the presence of each of the affective states. We then use this an input to each of the hidden Markov models for each affective state and classify it based on the model that yields the highest probability of generating that sequence of states. Figure 3 shows a diagram of this process. We found that this approach can yield good results when the localized affect classifiers are trained on representative intervals, which can be obtained from judgments from external human annotators. We discuss more of this on the next section.

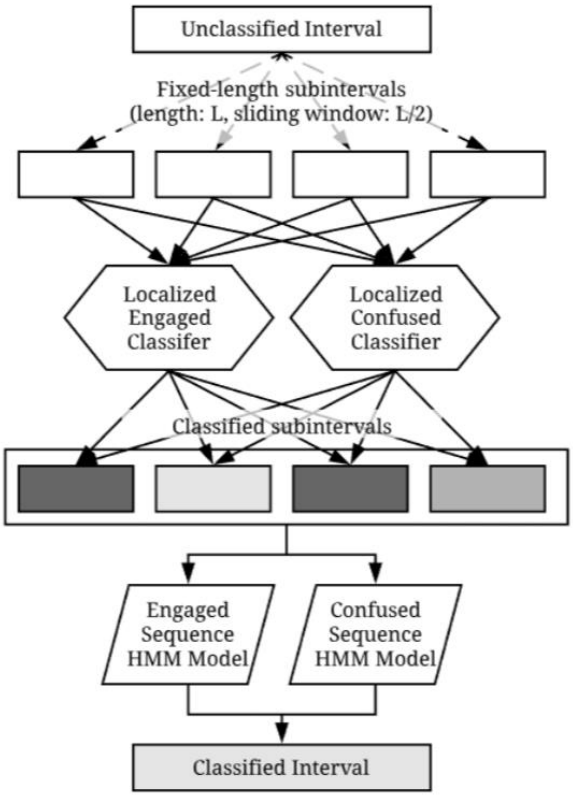

Figure 3. Classifying longer intervals from sequence of sub-interval classifications (engagement and confusion).

3. Results

3.1 Classification of Fine-Grained Intervals

Figure 4 shows the performance of the best classifiers for each affective state label on a student-level cross-fold validation. We treated each affective state as a binary classification problem (e.g., "engaged" or "not engaged", where all intervals labelled "confused", "frustrated" and "bored" were considered as the negative class). Because we are dealing with imbalanced datasets, we use Cohen's kappa κ as the metric for performance. $\kappa = 0$ means that the model is not any better than a model that predicts randomly by chance, while higher values are better.

By using face features alone, it is very difficult to predict the presence of affective states beyond random chance. This is not very surprising as it is consistent with previous findings in the literature. However, by combining log features and face features, the performance for classification on the 20 second and 60 second sub-intervals could be improved. Among the sub-interval lengths, L = 60 had the best performance. Among the affective states, engagement yielded the best performance, reaching $\kappa = 0.30$ on the 20 second interval and $\kappa = 0.45$ on the 60 second interval. Both of these used the multilayer perceptron classifier. However, detection for the other affective states remained poor despite the improvement.

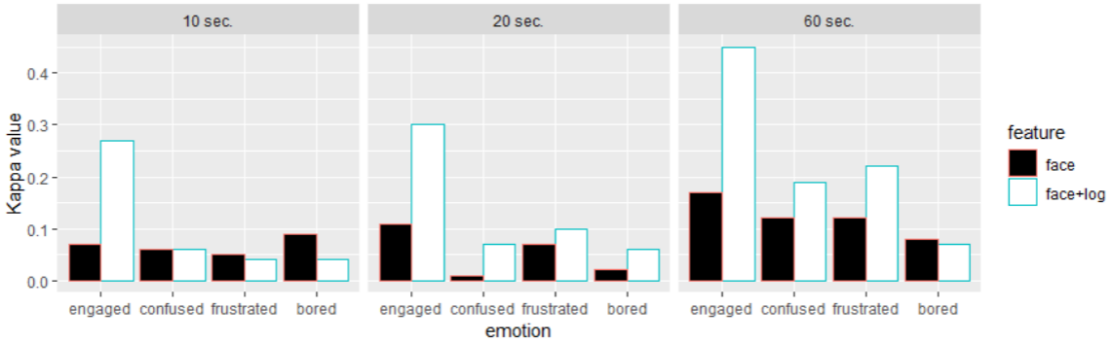

Figure 4. Performance of Best Classifiers on Fine-Grained Detection for Students' Self-Report.

We posit that one of the reasons for the difficulty of fine-grained affect classification is the sparseness of the displays of affect over an emotional episode. For example, if the student reports confusion for several consecutive intervals, he is reporting a persistent emotion that is not necessarily displayed to an observer in those intervals. Even though the student was asked to do the self-report for each individual interval locally, it is impossible to separate it from the emotions experienced in the surrounding intervals and from the context of the situation. Thus, we think that only a subset of the intervals are truly representative of the reported emotion in terms of the display of affect.

In order to determine these representative intervals, We tried to get a more localized judgment of affect by asking two human annotators to manually watch each sub-interval (L = 10, 20, 60), and provide an affect judgment of "engaged", "confused", "frustrated", or "bored". One of the annotators is a professor who has taught introductory programming courses, while the other one is a master course student who has served as a teaching assistant for the same course, so both of them have experiences on working with students who are learning how to code.

Because the annotators did not experience the programming session first-hand, they are free from any persistent emotion that might influence their judgments. We also shuffled the order the sub-intervals were presented to the annotators to further minimize the chance of associating the current judgment with the student or with the context of a particular episode, in an attempt to get localized affect judgments within each interval. We took all the sub-intervals where the two annotators and the student's self-report are all in agreement and considered them to be the representative sub-intervals of that affective state. The percentage of agreement between the two annotators is further discussed in Section 4.2.

We performed classification on the representative sub-intervals resulting into significantly better results. However, because the original data contained a low number of reports for frustration and boredom, there were very representative intervals identified to make a meaningful classification. The results for engagement and confusion for classifying representative intervals, both for face features only and both face and log features, are shown in Figure 5.

The performance of the models for representative intervals was better in both classifying engagement and confusion against all other emotions, reaching $\kappa = 0.65$ and $\kappa = 0.85$ for classifying engagement on 20 second and 60 second intervals, respectively and $\kappa = 0.49$ and $\kappa = 0.72$ for classifying confusion on 20 second and 60 second intervals, respectively. It should be noted that our intention for this is not to disregard all the other intervals that could not be classified easily, but to identify the important intervals which can be used later on to improve classification of longer sequences, which is discussed in the next section.

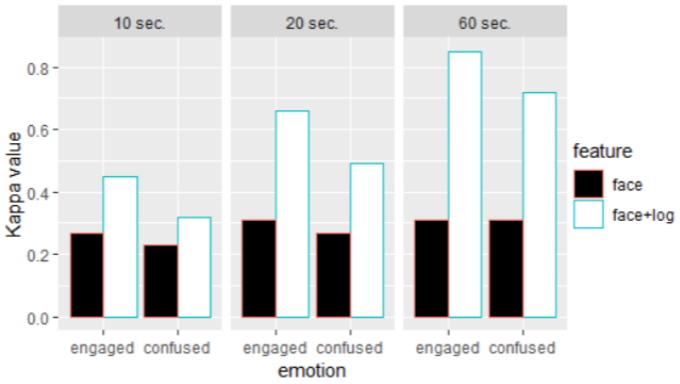

Figure 5. Performance of Best Classifiers on Fine-Grained Detection for Student Reports Combined with Annotator Judgment.

3.2 Classification of Longer Intervals

To classify longer intervals, we trained hidden Markov models from the sequence of predictions made by the localized classifiers discussed in the previous section. One hidden Markov model was trained for each affective state. Classification is performed by getting the model with the higher probability of generating the unknown sequence.

Because the localized interval classifiers trained on the actual student self-reports did not generalize very well on a student level apart from engagement, we used the localized classifiers trained on the representative intervals identified by the human annotators as described in the previous section. Unfortunately, since we could not build frustration and boredom models for representative intervals due to the lack of data, we could only perform classification for engagement and confusion. To ensure the models are generalizable, the models were still evaluated also on a student-level cross-fold validation, in which both the HMM models and the localized classifiers were trained only with instances that belonged to students that are in the training set.

Table 2 shows the results of the classification of the original intervals from the student reports using 10 second, 20 second, and 60 second sub-interval length for both face features only and face features combined with log features. We found that the best performing model was using both face and log features and splitting the interval into 20 second intervals with a sliding window of 10. This model was able to classify 76% of the intervals correctly, with $\kappa = 0.45$. It could also be observed that using face feature and log features improved the performance of the models when compared to using face features alone.

Table 2

HMM Classification Results

Face Features Only			**Face and Log Features**		
Interval	Acc.	κ	Interval	Acc.	κ
10 sec.	0.6	0.15	10 sec.	0.65	0.24
20 sec.	0.62	0.11	20 sec.	0.76	0.45
60 sec.	0.58	-0.14	60 sec.	0.58	0.22

4. Discussion

4.1 Contributions of this Study

In this study we performed two types of classification on affective states in programming: fine-grained classification of fixed-length intervals, and classification of longer intervals based on sequences of sub-intervals. Our study has two main contributions: (1) adding log features can improve the performance of classifier models for affect in activities like coding, and (2) using judgments of external observers to identify representative intervals can improve classification on longer intervals.

For the first contribution, on both types of classification, we have shown that adding log features in combination with face features can improve the prediction performance of classifier models. This suggests that the student activity is valuable in estimating emotion, especially in types of activities like programming where there is relatively more reserved and naturalistic displays of affect. For the second contribution, we have shown that the sequence of classifications from a model trained on representative sub-intervals can be used to classify longer sequences of affect. Although the models that we were able to train were far from excellent, it presents an improvement to previous attempts in the domain of affect detection in a programming setting.

4.2 Challenges in Fine-Grained Affect Detection

Fine-grained affect detection can generally be useful for intelligent programming tutors for systems to be constantly aware of the student's state, as opposed to being only aware in specific scenarios. In line with this, it is also interesting to determine the ideal length of the interval for discriminating different emotion types. In study, even though we were able to improve the performance of classifiers for fine-grained affect detection by adding log features, the resulting models for confusion, frustration, and boredom did not really achieve a significant performance above chance.

The difficulty of fine-grained affect detection may be caused by the fact that (1) emotions experienced by the students are not localized to short intervals, but rather influenced by the events that happen outside the interval, and (2) displays of affect do not necessarily occur constantly all throughout an emotional episode.

Table 3

Agreement Rate with Human Annotators (A/N: ratio of intervals where annotators made the same judgment over all intervals, A+S/N: ratio of all intervals where annotators and self-report are the same judgment over all intervals, A+S/S: percentage of all self-reports under the emotion that was correctly identified by both annotators)

Interval	Emotion	A/N	A+S/N	A+S/S
10 sec.	Engaged	52%	27.66%	65.1%
	Confused	17%	5.83%	25.55%
	Frustrated	0.67%	0.17%	0.57%
	Bored	0%	0%	0%
20 sec.	Engaged	52.67%	35.17%	66.57%
	Confused	17.17%	4.67%	25.69%
	Frustrated	1.17%	0.67%	2.96%
	Bored	0%	0%	0%
60 sec.	Engaged	53.27%	37.07%	72.12%
	Confused	11.53%	4.05%	21.31%
	Frustrated	1.25%	0.93%	3.95%
	Bored	0%	0%	0%

In this study, we asked external observers to identify which parts of an emotional episode are representative of that emotion by identifying the intervals where the observers' judgment and the student report agree with one another. Using these intervals, we were able to train fine-grained interval

classifiers that generalize significantly better on student-level cross-fold validation. This suggests that the representative sub-intervals are indeed much easier to discriminate from one another and can be considered as the sub-intervals that define each affective state. One of limitations that we encountered in this study, however, is that we did not find enough representative intervals for frustration and boredom to make a meaningful classification model. Table 3 shows the agreement of the two annotators and the student report for each of the affective states. Based on this data, around 65% to 70% of engaged sub-intervals could easily be recognized locally, but only 20% to 25% of the confused sub-intervals could easily be recognized locally.

The lack of agreement on frustration and boredom might have been caused by the fact that the student reports for these two affective states were relatively less than the other two, and thus there simply weren't enough examples for enough representative intervals to be identified. However, it is also a possibility that these two emotional states are not easy to detect for humans in short, localized intervals on face and log features alone, and must be supplemented with additional information.

4.3 Classification on Longer Sequences

In this study we also classified longer sequences based on prediction output on fixed-length sub-intervals. Figure 6 shows 10 randomly selected examples that were correctly classified by the HMM models from each class, as well as the corresponding classifier outputs for each sub-interval in the sequence. Each cell in the figure represents a 20 second sub-interval in the sequences, with a 10 second sliding window (i.e, half of the sub-interval overlaps with the previous one). There are four possible discrete symbols, since we considered two classes in this case: (1) both classifiers returned false, (2) only the "engaged" classifier returned true, (3) only the "confused" classifier returned true, and (4) both classifiers returns true.

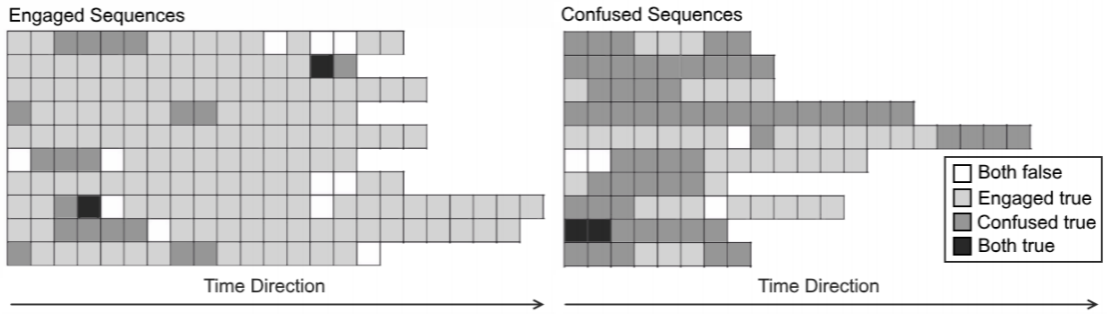

Figure 6. Examples of "engaged" and "confused" sequences correctly classified by the HMM models.

By visually inspecting the sequences, it could be seen that in the cases where the student reported engagement (i.e., the engaged sequences), most of the sub-intervals were classified as "engaged". However, there were also some cases wherein some parts were classified as "confused", but in these cases the confusion eventually turns into engagement. These sequences most likely represent confusions that were eventually resolved. On the other hand, for the "confused" sequences, there appeared to be more prevalent sub-intervals that were classified as "confused".

It can be said that both "engaged" and "confused" sub-intervals could be observed from within both types of sequence, and this further supports the idea that affect judgments made locally within an interval may not always reflect the actual emotional experience experienced. Furthermore, it should also be noted that one limitation of our study was that we did not consider co-occurring affective states.

4.4 Useful Features on the Prediction of Affect

To investigate which features are useful in predicting each affect, we used the RELIEF-F feature ranking algorithm. RELIEF-F ranks the features based on how good they are at discriminating each instances from their nearest neighbors of a different class.

For engagement, document insertions was ranked highly across all sub-interval lengths, both by using all sub-intervals and by using representative sub-intervals only. Document insertions tend to

occur more in moments of engagement. For example, in the 20 second fixed-length sub-interval case, document insertions were significantly larger in "engaged" sub-intervals ($\mu = 16.4$) compared to the other sub-intervals ($\mu = 8.22$) with a $p < 2.2 \times 10^{-16}$.

Compilation errors was also ranked highly for classifying engagement and confusion. Upon further analysis, we found that compile errors occurred significantly less on "engaged" ($\mu = 0.01$) sub-intervals as compared to "not engaged" sub-intervals ($\mu = 0.03$, $p < 0.002$) and significantly more on "confused" ($\mu = 0.04$) sub-intervals compared to "not confused" intervals ($\mu = 0.02$, $p < 0.01$).

AU04 (brow lowerer) was ranked highly in the classification of confusion and frustration. AU04 was observed in the data when the eyebrow is furrowed, as well as when the subjects gaze downwards. This often happened during typing when the eyes quickly shift between looking at the screen and at the keyboard without turning the head down. Figure 7 shows some examples of AU04. Aside from this, the ranking of the other face features varied depending on the length of the sub-interval, even when only representative intervals are considered.

Figure 7. Displays of AU04 (brow lowerer) by a Filipino (left) and Japanese (right) student.

5. Conclusion

In this study, we have presented improvements to the detection of academic emotions in a programming setting by combining log information to the face features. We have also developed an approach for classifying longer sequences by training models on representative intervals which combine students' self-reports with the localized judgments of external annotators.

Acknowledgements

The authors would like to thank Mr. Fritz Kevin Flores, Mr. Manuel Carl Toleran, and Mr. Kayle Anjelo Tiu for assisting in the facilitation of the data collection process in the Philippines.

References

Baltrusaitis, T., Zadeh, A., Lim, Y. C., & Morency, L. P. (2018, May). Openface 2.0: Facial behavior analysis toolkit. In 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018) (pp. 59-66). IEEE.

Bosch, N., & D'Mello, S. (2013, June). Sequential patterns of affective states of novice programmers. In The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013) (pp. 1-10).

Bosch, N., D'Mello, S., & Mills, C. (2013, July). What emotions do novices experience during their first computer programming learning session?. In International Conference on Artificial Intelligence in Education (pp. 11-20). Springer, Berlin, Heidelberg.

Bosch, N., Chen, Y., & D'Mello, S. (2014, June). It's written on your face: detecting affective states from facial expressions while learning computer programming. In International Conference on Intelligent Tutoring Systems (pp. 39-44). Springer, Cham.

Cabada, R. Z., Estrada, M. L. B., Hernández, F. G., Bustillos, R. O., & Reyes-García, C. A. (2018). An affective and Web 3.0-based learning environment for a programming language. Telematics and Informatics, 35(3), 611-628.

Cho, M. H., & Heron, M. L. (2015). Self-regulated learning: the role of motivation, emotion, and use of learning strategies in students' learning experiences in a self-paced online mathematics course. Distance Education, 36(1), 80-99.

Crow, T., Luxton-Reilly, A., & Wuensche, B. (2018, January). Intelligent tutoring systems for programming education: a systematic review. In Proceedings of the 20th Australasian Computing Education Conference (pp. 53-62). ACM.

Daniels, L. M., Stupnisky, R. H., Pekrun, R., Haynes, T. L., Perry, R. P., & Newall, N. E. (2009). A longitudinal analysis of achievement goals: From affective antecedents to emotional effects and achievement outcomes. Journal of Educational Psychology, 101(4), 948.

D'Mello, S., & Graesser, A. (2012). AutoTutor and affective AutoTutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. ACM Transactions on Interactive Intelligent Systems (TiiS), 2(4), 23.

D'Mello, S. K., & Kory, J. (2015). A review and meta-analysis of multimodal affect detection systems. ACM Computing Surveys (CSUR), 47(3), 43.

Ekman, P., & Friesen, W. V. (2003). Unmasking the face: A guide to recognizing emotions from facial clues. Ishk.

Grafsgaard, J. F., Boyer, K. E., & Lester, J. C. (2011, October). Predicting facial indicators of confusion with hidden Markov models. In International Conference on Affective computing and intelligent interaction (pp. 97-106). Springer, Berlin, Heidelberg.

Grafsgaard, J. F., Wiggins, J. B., Boyer, K. E., Wiebe, E. N., & Lester, J. C. (2013, September). Automatically recognizing facial indicators of frustration: a learning-centric analysis. In 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (pp. 159-165). IEEE.

Grafsgaard, J. F., Wiggins, J. B., Boyer, K. E., Wiebe, E. N., & Lester, J. C. (2013, July). Embodied affect in tutorial dialogue: student gesture and posture. In International Conference on Artificial Intelligence in Education (pp. 1-10). Springer, Berlin, Heidelberg.

Lehman, B., Matthews, M., D'Mello, S., & Person, N. (2008, June). What are you feeling? Investigating student affective states during expert human tutoring sessions. In International conference on intelligent tutoring systems (pp. 50-59). Springer, Berlin, Heidelberg.

Malekzadeh, M., Mustafa, M. B., & Lahsasna, A. (2015). A review of emotion regulation in intelligent tutoring systems. Journal of Educational Technology & Society, 18(4), 435-445.

Mega, C., Ronconi, L., & De Beni, R. (2014). What makes a good student? How emotions, self-regulated learning, and motivation contribute to academic achievement. Journal of educational psychology, 106(1), 121.

Paleari, M., Lisetti, C., & Lethonen, M. (2005). Virtual Agent for Learning Environment Reacting and Interacting Emotionally. Eurecom. Fr, 3-5.

Pekrun, R., Goetz, T., Titz, W., & Perry, R. P. (2002). Academic emotions in students' self-regulated learning and achievement: A program of qualitative and quantitative research. Educational psychologist, 37(2), 91-105.

Rodrigo, M. M. T., Baker, R. S., Jadud, M. C., Amarra, A. C. M., Dy, T., Espejo-Lahoz, M. B. V., ... & Tabanao, E. S. (2009, July). Affective and behavioral predictors of novice programmer achievement. In ACM SIGCSE Bulletin (Vol. 41, No. 3, pp. 156-160). ACM.

Tiam-Lee, T. J., & Sumi, K. (2017, September). A comparison of Filipino and Japanese facial expressions and hand gestures in relation to affective states in programming sessions. In Workshop for Computation: Theory and Practice (WCTP) (pp. 143-157). World Scientific, Singapore.

Tiam-Lee, T. J., & Sumi, K. (2018a, June). Adaptive feedback based on student emotion in a system for programming practice. In International Conference on Intelligent Tutoring Systems (pp. 243-255). Springer, Cham.

Tiam-Lee, T. J., & Sumi, K. (2018, October). Procedural Generation of Programming Exercises with Guides Based on the Student's Emotion. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 1465-1470). IEEE.

Tiam-Lee, T. J., & Sumi, K. (2019, June). Analysis and Prediction of Student Emotions While Doing Programming Exercises. In International Conference on Intelligent Tutoring Systems (pp. 24-33). Springer, Cham.

Vea, L., & Rodrigo, M. M. (2016, August). Modeling negative affect detector of novice programming students using keyboard dynamics and mouse behavior. In Pacific Rim International Conference on Artificial Intelligence (pp. 127-138). Springer, Cham.

Zatarain-Cabada, R., Barrón-Estrada, M. L., Camacho, J. L. O., & Reyes-García, C. A. (2014, August). Affective Tutoring System for Android Mobiles. In International Conference on Intelligent Computing (pp. 1-10). Springer, Cham.