Recognizing Students' Confusion Using Pattern Recognition Methods

ECE471/571 (Pattern Recognition)

Kirolos Shahat(471), David Clevenger(471), Brett Brownlee(571)

Abstract

Confusion isn't just a part of learning, it is the process you learn from. It only becomes a hindrance if it isn't properly addressed. Courses are often taught where either the material is difficult to comprehend or the students are not able to see it in the same light as the professor. Thus, it would ideally be best if the professor were able to be notified, through a system, when students were not fully comprehending a topic in order to remedy this gap in knowledge early. Electroencephalography (EEG) is a signal monitoring method to record electrical activity of the brain. With the use of EEG one could theoretically create a model which could predict whether a student, given the electrical activity information, was confused or not in order to convey the method in a more clear way to the students.

We implemented Maximum Posterior Probability, k-Nearest Neighbors, and Naive Bayes classifier fusion and integrated them with the Sklearn Decision Tree and Back Propagation Neural Network libraries as well as the Matlab Support Vector Machine library in order to attempt to classify a confused versus not confused student. Our best performing accuracy ended up being SVM with 63.3% followed by BPNN with 62.4% while the rest did not surpass 59% accuracy. We believe this is caused by the method of obtaining the samples because of binary classification over a time interval which could generate both types of results. Thus, ideally a way to better decipher whether a student was actually confused or not would be to take data at time stamps where a student would be guaranteed to not understand.

Introduction

Background

Adding to Kaggle's repository of machine learning datasets, Haohan Wang generated signal data from ten college students as they watched Massive Open Online Course (MOOC) videos, where there were twenty in total. The videos were hand picked such that half of them were labeled likely to be confusing by the standards of the researchers, an example being Quantum Mechanics. The other half were labeled as likely to not be confusing, an example being basic Algebra. The students tested during the experiment were between the ages of 21 and 31. Most were male and Han Chinese with the exception of four which were of different ethnicities or genders.

Collection

Using the MindSet devices, students' brainwave data was gathered using software that was implemented by Wang and others. At the end of a session, students rated their confusion on a scale of 1-7. These values were then translated by Wang and others to a binary label, either confused or not confused. Additionally, Wang and others added a predefined label of whether or not they believed the students were confused, which was based on the EEG data. The videos watched by the students had a length of two minutes in duration. In case the student was not prepared, the first and last 30 seconds were ignored so the data was obtained in the intermediate 60 seconds of the video.

Equipment

The device used was a single sensor portable EEG device. Collected data is broken up into two main named categories by the sensors designers into Attention and Meditation. If their tuning is well done, results from our analysis may agree that these values are indicators of mental activity. There are additional values named after frequency region activities of the brain where the EEG data frequency range is broken up: Delta, Theta, Alpha 1&2, Beta 1&2, Gamma 1&2. Recorded data is from a single sensor placed over the forehead. This can make analysis a challenge because this isn't enough to claim any significant information from a human brain. State of the art Pattern Recognition with EEG data is done on large dense clusters of electrodes (128, 256, or 512 electrodes). However, muscular activity can strongly obscure EEG signals and which require processing in order to remove. This means the given data might just give very limited indication of mental states of the participants It may also show noise from muscle factors from their facial motor habits such as furrowing a brow. This means there may still be some correlation between physical user watching activity and the given data even if there isn't deeper cognitive significance. A study has shown that even with a single sensor you can get very limited but comparable brain signal data (1). While much less rich than a cluster of electrodes, a set up like this is quick, easy, and portable. This leads the way to cheap, accessible, and more experimental research. Eventually this could even induce future head mounted display equipment to integrate this technology to make personalized perceptual cognition experiments commonplace. Thus, being able to go through the process of analysing this kind of data to verify findings independently is a genuinely useful exercise.

Technical Approach

We set up a pipeline where all models are run through a 23-fold cross validation to prevent overfitting. Model wrappers enabled use of generic parameters, allowing us to simply change the model name to generate its respective predicted labels. The fusion function allowed us to pass the predicted labels of two classifiers and the actual labels of the test set. And would combine these labels using the Naive Bayes fusion method. For Maximum Posterior Probability, kMeans, and kNN, our own code was used. For Back-Propagation Neural Network, Decision Tree, we used Python's Scikit-Learn implementations. For Support Vector Machine, we used MATLAB's Statistic and Machine Learning Toolbox toolbox. When implementing by hand, we attempted to use parallel implementations where possible. This is especially helpful of the case in kNN. Results generated were then written to CSV files for plotting or Stdout.

Experiment and Results

M-Fold Cross Validation

In order to accurately classify and forego the risk of overfitting, we chose to use cross validation. Since we have 12811 samples in the data set we discovered that it it evenly divisible by 23 and thus chose to use that number of folds. This turned out to work well and gave us a good average accuracy for each classifier used.

Maximum Posterior Probability

Maximum Posterior Probability was implemented with the three cases of discriminant functions. Beginning with the least assumptions, Case 3 assumed the data set obeyed a Gaussian distribution. Based off of the data set, we initially believed that the features would all have good relevance but it was not the case because Case 3 was the worst performing classifier of the ones tested with an average accuracy of 54%, which is almost randomly guessing. When we then tried Case 2, which assumes that the covariance matrices are the same, the performance had a 5% jump to 60%. Thus, there must be some overlap between the classes and their structures. Afterward, we then tried case 1 which assumes independence between features and it was a 3% increase from case 1 achieving an average accuracy of 57%. Thus, Case 2 seemed to be doing well compared to the accuracy the developers achieved which was 65%.

Case 1 Discriminant function confusion matrix

Case 2 Discriminant function confusion matrix

Case 3 Discriminant function confusion matrix

3 discriminant functions average ROC curve over 23 folds

k-Nearest Neighbors

K-Nearest Neighbors attempts to classify a sample based on the nearest neighbors, for some predefined value k, using Minkowski Distance. When initially implemented and run on the EEG dataset provided, after preprocessing and 23-fold cross validation, it was observed that the run time was very long and results were not feasible to obtain. Thus, varying the parameters of kNN ended up being too time intensive to actually implement. Thus, we had to stick with a guess at what the optimal k value and Minkowski distance would be. The values chosen were 15 neighbor voting using Manhattan distance.

We attempted to optimize kNN’s distance calculation because of how inefficient kNN is to classify, especially given that it was implemented in Python. To do this, we parallelized the distance calculation function by running it as an array over each training data point. This turned the loop over all training data into a function of an array. This was able to speed up runtime with smaller datasets in previous instances. However, when running it on our large 14,000 dataset it was not able to improve performance like it did with our smaller dataset.

The results of this algorithm were average among the algorithms. Over the span of the 23 folds of cross validation, the average accuracy attained was 60.9% which, given how computationally intensive this classifier takes to run, is not worth the for a 1% increase.

KNN(k=15, d=1) confusion matrix

KMeans

KMeans was used as a way of generating cluster centers for the data. These clusters were then used in a minimum distance classifier on the testing set using Euclidean Distance. We found that results were similar to those of other models.

KMeans+MD confusion matrix

Decision Tree

The decision tree used Gini impurity as the criterion for splitting. As an added benefit, the python shows the importance of each of the features. In the case of our dataset, *Delta* and *Gamma2* were the most important features.

Normalized Decision Tree Confusion Matrix

Back Propagation Neural Network

The neural network initially used stochastic gradient descent and the sigmoid activation function as its parameters. These were later swapped out for Adam and Rectified Linear Units in favor of higher overall accuracy across the folds. The network structure was 11-100-1, by default but was also tested with 5 nodes and 100 nodes. In each case the accuracy changed by less than .2%, which could be attributed purely to randomness.

Normalized BPNN Confusion Matrix

Support Vector Machine

Our results gave a performance accuracy of 63.3% . This is the best performance out of all our classifiers. Our high end goal performance that other groups have gotten with this data was around 65%. We feel we have approached this with SVM.

Normalized Average Confusion Matrix Results of SVM(RBF Kernel) through Cross Validation

Above shows our confusion matrix output. Our algorithm is least likely to make False Negative predictions. In the case of our confusion dataset, it is optimal to lean towards checking if a student is confused instead of missing it.

Classification and Misclassification shown across Meditation and Attention aggregate values

After getting these positive results, we looked to see if they could be visualized across the main two features promoted by dataset. Above shows the false positives, false negatives, true positives, and true negatives all. This isn't as clearly understood as we hoped. Like the raw data, there is not an easy way to interpret the pattern through inspection.

3D output of SVM results from first 3 principal components

Our last attempt to visualize our SVM results was across the PCM dataset. By reducing the PCM to 3 dimensions, we could plot out entire SVM output.
Running SVM around PCM reduced accuracy to 55.59%. However, it still has a similar spread in its confusion matrix so we hoped we could visualize some formation.

3D output of SVM results from first 3 principal components zoomed in

Even with zooming, it is not clear if there are any regions where the classifier performs well. However, it does look someone clear that there are more false classifications (the red) where outliers occur. Ultimately, the SVM performed well but was not able to help us find a deeper understanding of our highly complex feature. This was a side goal bu

Dimensionality Reduction - Fisher's Linear Discriminant and Principal Component Analysis

We applied dimensionality reduction to the data sets to see the types of effects it would have on the data set and for every tested case they underperformed compared to the the normal dataset thus we chose to not pursue it in for classification between the dimensionality is fairly low. The highest accuracy achieved was 58% using case 2 MPP and FLD, which is not a significant increase but since the constraint is not high, the computational savings are not worth it.

Naive Bayes Classifier Fusion

Naive Bayes Fusion was used as a way to optimally combine the results of two classifiers, in the bayesian sense, of course. This allowed the predictions from the models that was "best believed" to be correct to be chosen as the predicted label for any given test sample. However, fusion of the best models did not increase the accuracy of the models. The models we chose to fuse were BPNN, Case 2 MPP, and Decision Trees because they were the best performing. We did not fuse SVM because it was implemented in matlab and would have been too difficult to integrate with the Python code, which is where fusion was implemented. The accuracy for BPNN and Case 2 was 62.4%, which is equivalent to just running BPNN alone. Fusion between BPNN and decision trees had an accuracy of 61.4% which is worse than just running BPNN or SVM. This is likely due to low accuracy in the models and the potential that BPNN's classification area encompasses the other two models. Thus, we chose to forego the use of fusion.

Task Allocation

- Kirolos Shahat contributed to preprocessing steps, implementation of kNN, MPP, and PCA
- David Clevenger contributed to preprocessing steps, implementation of FLD, BPNN, decision trees, and clustering
- Brett Brownlee contributed to data understanding, preprocessing steps, visualization, SVM, and kNN parallelization

Discussion

There were a lot of classifiers which we tried as well as also trying multiple fusions of multiple different classifiers but the results all seemed underwhelming. They were all mostly comparable with the accuracy score achieved of the developers of the data set but the accuracy was not significantly better than random guessing. A belief as to why this might be is potentially due to the way the data is obtained. According to Haohan Wang, a student classified themselves as confused after fully watching a video. This is a problem because the data which is attained is not obtained once at the end of a video, but is continuous. Thus, this data could be filled with false positives and false negatives because a student can be confused for a portion of the video and understanding for the other portion of the video. This introduces a large amount of confusion for models which are trained and could be the root of the large amount of misclassification.

Summary and Conclusion

Eight different classifiers were implemented on a confused student EEG dataset from Kaggle. The goal was to determine whether a student was confused or not from brain wave information gathered from watching a video. Our goal was also to compare the performance of these classifiers with the state of the art accuracy score reported by the developers of the dataset of approximately 65%. The best two performing classifiers ended up being 63% for SVM and 62% for BPNN which are on a comparable scale with the developers of the dataset. Though the accuracy scores were comparable to their models, at 65%, the accuracy is not high enough to be used in practice as a way of determining whether or not a student is confused. Instead, it could be used as a minimalistic way for teachers and professors to gauge the understanding of their class on a rough basis. A goal for the future would be to analyze the data more and see if there is a better split in the data with more classes introduced rather than making it binary. This is inspired because of the method the researchers used to develop the dataset which is asking the students to rate their level of confusion on a scale after watching a video. This could potentially help split the data in better categories and help with classification. Another interest is using gradient boosted trees such as the XGBoost library. Given more time we could potentially look into deep learning methods as well to better extract the features that are the most relevant for classification.

Reference

(1) EEG From a Single-Channel Dry-Sensor Recording Device Stuart J. Johnstone, Russell Blackman, and Jason M. Bruggemann Clinical EEG and Neuroscience Vol 43, Issue 2, pp. 112 - 120 First Published March 27, 2012 https://doi.org/10.1177/1550059411435857
(2) https://www.kaggle.com/wanghaohan/confused-eeg
(3) http://scikit-learn.org/stable/