Classifying Wake Stage with Logistic Regression

Marvin Cangcianno, BS
Georgia Institute of Technology

Abstract

This paper utilizes classification algorithm to classify wake stage using EEG channels with no prior knowledge. Existing models use fix calculation or unreliable tracking to determine the best time to wake up. Most of them also don't explain clearly how they actually come up with the decision. In this paper, I try to make a classification model to classify whether the person is in Wake stage, at any given second during their sleep. With clear and well-explained classification algorithm, I hope it is easier for people to trust this calculation for choosing the best time to wake up that fits their preferred time window.

Presentation link via Youtube: https://www.youtube.com/uUgaKNPIZF8

Introduction

How many times our morning alarm woke us up but we still feel exhausted or groggy, even though we believe we had enough sleep? Scientifically, there are multiple reasons why sometimes people wake up tired. One reason is because we were woken up in the middle of our Non-REM stage, or more commonly known as deep sleep stage. For us to wake up fresh and energized, we need to wake up when we are in our lightest sleeping stage, which is when we start a new sleep cycle. This stage is called the Wake stage. However, it is not possible for people to accurately know the time when they are in the Wake stage beforehand and adjust the alarm accordingly, because well, they are sleeping.

To determine these stages, sleep experts record electrical activity emitted from people's brain during their sleep. These activities are recorded in a form of Electroencephalograph (EEG). For a person to fully analyze these activities and find out the wake stages, they need to have a lot of knowledge in neurology. Currently, there are a lot of existing models that use this fixed rule-based method to calculate when your alarm should ring in the morning or when you should hit the bed in the evening. However, it is difficult to figure out how precise these calculations are because this method is widely accepted by sleep experts in the world as the only available method to classify sleep stages.

In this paper, I propose to create a classification model to classify specifically the Wake stage, which is the best time to wake up, given a person's EEG data, and apply them to an alarm system that accepts the label and acts accordingly. If the the user is in Wake stage and the time fits user's preferred time window, it will ring the alarm.

Literature Survey

Most of the times when we wake up tired, we immediately assume that it has to do with our irregular sleep schedules. For normal people, we consider our sleep quality based only on the number of hours we sleep. Indeed, it is one of the factor. Kang et al. (2009) [1] conclude that there is a high correlation between bedtime schedule irregularity and insufficient sleep, because of their bad sleep qualities. But for this experiment, I decided to measure sleep quality by the how fresh people feel when waking up.

Rosenberg et al. (2014, p. 9-11) [2] state that “a good-eight hour sleep allows us to move through five cycles.” Rapid Eye Movement (REM) sleep is when most of our muscles are paralyzed and we dream and people output Alpha wave. It makes up only 20% of our overall sleep, but it is critical that we avoid waking up during Theta or Delta stage, both are part of Non-REM, since we “could feel drowsy, groggy, or confused when coming out of this sleep.” The best time to wake up is during Alpha stage. However, it is the shortest cycle of the sleep, so it is really hard to manually calculate the time accurately, which is why I’m proposing a model to predict those times.

Machine learning techniques being applied to determine sleep stages from EEG have been done before, by Yu et al. (2009) [3]. Their experiment was on rats, not human. In their pipeline, two machine learning algorithms were used, principal component analysis (PCA) to reduce the 2 dimension of the feature and k-nearest neighbors (k-NN) to classify the stages. Their system was able to achieve a 95.43% accuracy on predicting current sleep stage for every second. Eight years later, in 2016, a more modern technique was experimented by Biswal et al. (2016) [4]. They created SLEEPNET, an automated sleep staging system with deep learning. SLEEPNET achieved a respectable 86% accuracy, which is comparable to human-level scoring performance.

Method

- **Ground truths**

The ground truths can be extracted from the annotation which is in XML format. Since this experimentation is a binary classification, for each second, the label is either 1 (Wake Stage) or 0 (Other Stages).

- **Feature classes**

For each of the EEG data, I extract the first 8 channels to create 8 features. With that, I get a nx8 matrix, where n is the duration of the sleep in second. Then, each feature column is normalized with Min-Max Normalization, so they ranges equally from 0 to 1. Then, these classes are joined with the corresponding labels from the ground truths. Finally, the matrix is converted into svmlight format to be loaded for the training process.

- **Classification Model**

From the whole svmlight features, they are split into training set and test set. The training set contains 4000 records (69%), while the test set contains 1788 records (21%). Two records was inconsistent with the others and causes unknown error, therefore, they were ignored from the experiment. After that, the training set is split into folds to train the model a few combined records at a time. Various linear classification algorithms are tried in this experiment, such as logistic regression classifier, linear SVM, neural network, and random forest. Since the number of features is small, no feature reduction or selector algorithm is used in the process [6].

Every 100 records, the model is saved to the machine. While the model is being fitted with the folds, the performance of the model is evaluated every 500 records using the testing set, with the purpose to see the performance trend with the number of training data.

Experimental Results

- **Data**

With given permission, the dataset is provided by the National Sleep Research Resource (NSRR) and collected for the Sleep Health Heart Study (SHHS). The Sleep Heart Health Study is a multi-center cohort study implemented by the National Heart Lung & Blood Institute to determine the cardiovascular and other consequences of sleep-disordered breathing. It tests whether sleep-related breathing is associated with an increased risk of coronary heart disease, stroke, all cause mortality, and hypertension. There are 5790 records of EEG data, each from distinct sleep session. There are 14 channels per EEG records. NSRR also provides their annotation of each record. These annotations contain the sleep stages for the person each second during their sleep. The whole dataset is quite huge, around 350GB. It took me two days to finish downloading the whole dataset.

- **Model Implementation**

The hardware environment of the machine is a 4.0 GHz Intel i7-6700K 8M Skylake Quad-Core with 16 GB of RAM and powered by a GTX 980. Initially, the model was fitted with one by one records, or in other words, 1 record per fold. After trying various algorithm, the remaining choices left were linear SVM or logistic regression classifier, which surprisingly aligned with the research done by Manogram & Lopez (2018) [7]. Neural network and random forest took too long to train and their training time aren't justified by their results. Then, after some parameter tuning, logistic regression performs better than linear SVM. To further improve the accuracy, I tried to enlarge the fold size. With larger fold size, the result does become better and more consistent.

- **Results**

	SVM	Logistic Regression	Tuned SVM	Tuned Logistic Regression
Accuracy (%)	65.2	59.25	67	73.25

Table 1. Accuracies from different algorithms with default and tuned parameters

# of training data	Accuracy (%)	Precision (%)	Recall (%)
500	71	74	35
1000	76	94	15
1500	72	0	0
2000	72	0	0
2500	72	22	0
3000	72	35	0
3500	76	95	14
4000	75	82	25
Average	73.25	50.25	11.13

# of training data	Accuracy (%)	Precision (%)	Recall (%)
500	77.3	93.23	17.07
1000	77.24	94.59	16.57
1500	77.27	94.02	16.82
2000	77.25	94.3	16.7
2500	77.31	94.39	17.32
3000	77.27	93.86	16.87
3500	77.26	94.3	16.73
4000	77.29	93.44	16.98
Average	77.27	93.77	16.88

Table 2 & 3. Metrics for logistic regression with 1 records/fold (left) and 100 records/fold (right)

To tune the parameter, I used Grid Search to find the best parameter for each model. As seen, logistic regression performed best after tuned. Since in the library, there are two ways to make a logistic regression classifier, I also compared both of them [8]. In the end, one slightly performs better, and that is using SGDClassfier with “log” loss function.

These metrics are tested with testing set and they are averaged. Quite significant improvement can be easily recognized with enlarging the fold size. As can be seen, the final model achieved

77.27% accuracy, which is pretty respectable. The precision is also nicely high, 93.77%, while the precision is unfortunately quite low, 16.88%. The training time took 75.08 minutes. In total, the final model was fitted with ~120 million rows.

Discussion

As a final result, I am very happy with the result. One thing that I immediately noticed is that, even though the recall was quite low, high precision is more important because it means low false positives, which leads to low chance of mislabeling other stages as wake stages. It is very important to have achieve such confidence so that the system won't accidentally ring alarm when people are in their deepest sleep, which is not desirable. Additionally, I believe the reason the model could achieve such accuracy, even with few features and simple linear algorithms, is because the labels can be done by linearly separating the EEG frequency, which I used as features.

These results does not directly show that this model is better than current existing models since I was only able to do few tries on improving the model. However, these results does show a great potential to be really accurate with more appropriate algorithm and parameter.

Conclusion and Future Plans

This project has a really humble purpose to classify wake stages during a sleep. It can be very useful someday. Ideally on that day, it is easier to get EEG data individually while the user is sleeping. Currently, the most common way to classify Wake stages is to use EEG frequencies, but it takes certain neurological knowledge to do so. With this work, I propose to use a machine learning approach to classify the Wake stage based on EEG frequencies annotated by human experts. My hope is to have this model to perform as good as SLEEPNET or DeepSleepNet which perform to about 86% classifying all the sleep stages using deep learning. With this model focusing only on binary classification, I believe such goal is very achievable.

References

[1] Kang, Jiunn-Horng, and Shih-Ching Chen. "Effects of an Irregular Bedtime Schedule on Sleep Quality, Daytime Sleepiness, and Fatigue among University Students in Taiwan." BMC Public Health, vol. 9, no. 1, 2009, doi:10.1186/1471-2458-9-248.

[2] Rosenberg, Robert, and Julia Pastore. Sleep Soundly Every Night, Feel Fantastic Every Day : a Doctor's Guide to Solving Your Sleep Problems. 2014.

[3] Yu, Zong-En, et al. "A machine learning approach to classify sleep stages of rats." WSEAS International Conference. Proceedings. Recent Advances in Computer Engineering. No. 9. WSEAS, 2009.

[4] S. Biswal, J. Kulas, H. Sun, B. Goparaju, M. Brandon Westover, M. T. Bianchi, and J. Sun. SLEEPNET: Automated sleep staging system via deep learning. 26 July 2017.

[5] Supratak, Akara, et al. "DeepSleepNet: A Model for Automatic Sleep Stage Scoring Based on Raw Single-Channel EEG." *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, 2017, pp. 1998–2008., doi:10.1109/tnsre.2017.2721116.

[6] Rüger S. M., and S. E. Gauch. Feature Reduction for Document Clustering and Classification. Imperial College of Science, Technology and Medicine, Department of Computing, 2000.

[7] Manogaran, Gunasekaran, and Daphne Lopez. "Health data analytics using scalable logistic regression with stochastic gradient descent." International Journal of Advanced Intelligence Paradigms 10.1-2 (2018): 118-132.

[8] Zhang, Tong. "Solving large scale linear prediction problems using stochastic gradient descent algorithms." Proceedings of the twenty-first international conference on Machine learning. ACM, 2004.