Investigating Various Approaches in Classification of EEG Signals Representing Distinct Cognitive States to Reach an Optimal Solution

A Thesis Presented to Prof. Audrey Duarte

By

Soroush Mirjalili

In Partial Fulfillment of the Requirements for the Degree Master of Science in Psychology
Department

Georgia Institute of Technology

May 2021

Investigating Various Approaches in Classification of EEG Signals Representing Distinct Cognitive States to Reach an Optimal Solution

Approved by:

Dr. Audrey Duarte, Advisor
School of Psychology
Georgia Institute of Technology

Dr. Dobromir Rahnev
School of Psychology
Georgia Institute of Technology

Dr. Christopher Rozell
School of School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: January 28, 2021

To my lovely parents, brother, and little sister who have always supported me to be where I am today.

ACKNOWLEDGEMENTS

I'd like to appreciate the support of my advisor, Prof. Audrey Duarte, who has helped me a lot during this study. I should also thank the members of the Memory and Aging Lab who collected the data, and our participants for making this study possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Summary

There are various cases in which cognitive neuroscientists might be interested in exploring the neural differences associated with distinct cognitive states such as whether an individual has remembered some information or not. While it is common to use event-related potentials (ERPs) to distinguish neural activities representing different cognitive states, it does not allow us to explore single events because of its averaging nature. Classification of brain states associated with single events using real-time signals holds great potential for real-world applications such as brain-computer intervention systems that could support everyday learning. However, the progress in reaching high classification accuracy is still in early stages and thus, moving to the next step and creating such interventions is not possible yet. Moreover, previous studies applying classification methods to decode cognitive states have not typically compared different methods or explained the reasons for their choices. As a result, in this study, I systematically compared different methods of feature extraction, feature selection, and choice of classifier in the same study to investigate which methods work the best for decoding different episodic memory and perceptual "brain states." Using an adult lifespan sample EEG dataset collected during encoding and retrieval of objects paired with color and scene contexts, I found that the Common Spatial Pattern (CSP)-based features could distinguish the trials of different memory classes (i.e. item remembered vs. forgotten; context correct vs. incorrect; red vs. green vs. brown context perception) better than other types of features (i.e., mean, variance, correlation, features based on AR model, and entropy), and the combination of filtering and sequential forward selection was the optimal method to select the effective features. Moreover, Bayesian classification performed better than other commonly used options (i.e., logistic regression, SVM, and LASSO). These methods were shown to outperform alternative approaches for an orthogonal dataset, supporting their generalizability. My systematic comparative analyses allow me to offer some recommendations for cognitive researchers to consider when applying machine learning based classification to their datasets.

Keywords: Classification, Electroencephalography, Cognitive Neuroscience

1 INTRODUCTION

1.1. Distinguishing Episodic Memory States Using Event-related Potential

There are various situations that even an individual who does not have any memory impairment shows episodic memory failures. For instance, one might forget what color the taxi that picked him up earlier that day was, or where he parked in the parking lot that morning.

Cognitive neuroscientists have been investigating the neural underpinnings of these kinds of memory failures, and successes, for decades. The vast majority of these studies, from both functional magnetic resonance imaging (fMRI) and electroencephalography (EEG) methods, have used an averaging approach (Paller, Kutas, & Mayes, 1987). That is, signals from many trials of a particular type (i.e., subsequently remembered, subsequently forgotten), are averaged together to increase signal to noise ratio and consequently the power to detect the ERP differences between two states.

The first EEG study to use this technique was conducted by (Paller et al., 1987) in which they designed an incidental word learning paradigm and separated ERPs according to subsequent memory performance (so-called differences based on subsequent memory or ‘Dm effects’). Dm effects can be computed by subtracting the ERPs obtained by study items later forgotten (i.e., Misses) from the ERPs obtained by study items later remembered (i.e., Hits). They found a late positive ERP extracted from words subsequently recalled or recognized was greater than the one extracted from subsequently forgotten words. In agreement with this, in another EEG study, (Paller

& Wood, 1988) designed a similar incidental word learning paradigm followed by free recall test. By investigation of ERPs, they found that ERPs elicited from later recalled words were larger than ERPs elicited from later unrecalled words from 400-800 ms after onset. This divergence demonstrates that within this time interval, stimulus features that will subsequently distinguish correctly remembered from forgotten trials have been encoded by the brain.

Besides neural activity during learning procedure, using an incidental memory paradigm, (Otten, Quayle, Akram, Ditewig, & Rugg, 2006) showed that anticipatory activity before the onset of a stimulus can contribute to following episodic memory encoding. Finally, in addition to the temporal distinctiveness between ERPs of hits and misses, despite the fact that it is difficult to localize solely based on scalp-recorded ERP signals, (Johnson Jr., 1995) advanced the hypothesis that both imaging and intracranial ERP data supports the crucial role of medial temporal lobe (MTL) structures in generating the differences between ERPs of hits and misses.

However, the ERP differences are not solely related to the difference between hits and misses. Indeed, they can be used to extract additional interesting interpretations from the memory-related neural activities as well. Initially, these findings were based on analyzing the neural activity at retrieval. For instance, (Duzel, Yonelinas, Mangun, Heinze, & Tulving, 1997) designed a word recognition task to investigate different types of conscious awareness in episodic memory. To illustrate, they recorded ERPs from healthy individuals while they specified whether they “knew” or “remembered” the words they had seen previously. The ERPs for “known” trials indicated a temporoparietal positivity in 325-600 ms after onset and a frontocentral negativity in 600-1000 ms.

On the other hand, the ERPs for “remembered” items demonstrated widespread frontal and left parietotemporal positivity during 600-1000 ms. Furthermore, (Paller et al., 1987) found larger late positive ERP from words recognized correctly with high confidence than that from words recognized correctly with low confidence. Similarly, using EEG recording, (Woodruff, Hayama, & Rugg, 2006) designed a study in which the participants performed a modified remember/know task. In more detail, the test items that were not identified as remembered were scored on a four-point scale from “definitely old” to “definitely new”. They found that the amplitude of left frontal ERPs during 300–500 ms after onset discriminates confident old responses from confident new ones and varies monotonically with the confidence of the response.

In addition to studies that used ERPs from retrieval data, there have been more recent studies that examined neural activity during encoding. For example, in an EEG study, (Duarte, Ranganath, Winward, Hayward, & Knight, 2004) inspected whether the processes of recollection and familiarity are associated with distinct neural activity during encoding by presenting the participants pictures of objects for which they later had to make remember-know memory judgments. They observed spatial and temporal differences regarding subsequent familiarity and recollection neural activities. To be more specific, subsequent familiarity-based recognition reflected a left-lateralized positivity from 300 to 450 ms at the anterior scalp, while subsequent recollection reflected a spatially distinct right-lateralized positivity from 300 to 450 ms at the anterior scalp as well as bilateral activity from 450 to 600 ms.

Collectively, these studies demonstrate that various memory states reflect the outcome of spatially and temporally distinct memory processes. In my study, I intend to concentrate on neural differences between later remembered and later forgotten items as well as scrutinizing brain activity of distinct confidence levels.

1.2 Detection of Episodic Memory States in Real Time

While the differences between subsequently remembered and forgotten trials can be distinguished using an averaging approach, it does not allow us to assess episodic memory differences that might vary from moment to moment. To be more specific, not every item will be learned in the exact same way and by using averaging strategy, these differences between events will be lost.

Exploration of brain states associated with single events using real-time signals recorded from the scalp holds great potential for real world applications. For example, if we can detect the preparedness of the brain with high accuracy, one could build a practical intervention system that could support everyday learning. For instance, consider a student who wants to study for an exam. When his memory is working optimally, the system gives the student positive feedback that he is learning the material. However, once the system detects a decline in memory encoding, it could provide a warning to the student that, perhaps, it is time to take a break or study that material again, depending on the parameters that best facilitate learning. Such a system could be beneficial for teachers who could decide when to give students a break or start new material. As a result, this hypothetical brain computer interface (BCI) system could prove very helpful in customizing learning and memory for many individuals.

However, this hypothetical system does not exist yet and before such a system can be implemented, it is first necessary to determine an individual's episodic memory preparedness in real time. There have been a handful studies investigating this issue. For instance, in an EEG study, (Salari & Rose, 2016) designed a surprise recognition task where participants were first presented with a series of images. Significant increases in theta- and beta-oscillatory power were observed for subsequently remembered than forgotten events before the presentation of each stimulus. The researchers then implemented a brain-computer interface method to monitor the real-time oscillatory activity during learning and presented to-be-encoded stimuli when beta or theta power was high, indicating a potential good memory state, or low, indicating a potential poor memory state. They found an increase in memory accuracy for images that were presented in the high oscillatory power brain states. This study provides evidence that it is possible to use real-time neural signals to predict "good" and "poor" encoding brain states and also provide an intervention to improve learning in real time. I will talk about these two topics in more detail in the following sections.

1.3 Classification Procedures Using Machine Learning

While it is possible to separate distinct episodic memory states using basic statistical methods, a much more powerful tool is applying machine learning algorithms to real-time signals. Before delving into previous studies that used machine learning in classifying memory states, it is worth describing briefly the general methodology that is used in machine learning. The vast majority of classification problems focus on separating the data points of two classes and even multiclass classification problems are usually solved by generalizing binary classification's approaches. In any binary classification problem, the standard procedure that is applied includes feature extraction, feature selection or dimensionality reduction, and finally training a classifier (Lotte et al., 2017).

For feature extraction, information that may prove useful in distinguishing the classes is obtained from the data. This step is essential since the data cannot be helpful in separating 2 classes by itself. Some of the most common types of features that are used in EEG classification include voltage amplitude (Kaper, Meinicke, Grossekathoefer, Lingner, & Ritter, 2004), powers of different frequency bands (Pfurtscheller, Neuper, Flotzinger, & Pregenzer, 1997), phase of different frequency bands (Wei, Wang, Gao, & Gao, 2007), and Time-frequency features (T. Wang, Deng, & He, 2004). Statistical features including the mean and variance of the signal (the voltage), and correlation between signals of two channels are also sometimes used.

Besides statistical features, model-based features such as Autoregressive Model (Paranjape, Mahovsky, Benedicenti, & Koles', 2001), entropy-based features (Song & Liò, 2010), and common spatial pattern (CSP)-based features (Ramoser, Muller-Gerking, & Pfurtscheller, 2000) are frequently used in classification studies. To describe each briefly, in Autoregressive Model, a regression on the voltage of the current time sample and some number of previous time samples is performed, and the regression's coefficients are used as features. Entropy-based features reflect the degree of randomness in the signal. Moreover, CSP-based features are obtained by applying spatial filters that aim to maximize the variance difference between the trials of two classes.

Regarding feature selection, initially, a large number of features are extracted where most of them are not beneficial for separating the trials of the two classes. As a result, a criterion must be provided to eliminate nonessential features. One feature selection approach is to select features of the same type, another option is to select the best features from all types of features. To be more specific, various kinds of features are extracted, and each is assigned a score based on how well it can separate the trials from the two classes. Ultimately, the top features among all of the extracted features are selected for training the classifier (Phan & Cichocki, 2010). Another approach is to use sequential forward selection (SFS) to investigate the efficiency of combined sets of features instead of individual features (Mitchell, 1997). Specifically, a combination of top individual features might not be the best input for the classifier, while a combination of features that are not individually highly scored might best separate the two classes. SFS begins from an empty set of features, and in each step creates new subsets by adding a feature that is selected by an evaluation measure such as five-fold cross validation accuracy (Dash & Liu, 1997).

For training the classifier, there are various powerful classifiers that are commonly used in real-time brain signal classification in the literature. Support vector machine (SVM) tries to separate the data representing the different classes using a linear hyperplane (Burges, 1998). Similar to SVM, Linear Discriminant Analysis (LDA) aims to use hyperplanes and the projections of the data points on them to separate the trials of two classes (Fukunaga, 1990). Bayesian classification aims at assigning a set of gaussian probability distributions that describes the probability that a data point belongs to each class and selects the class in which the data point belongs to with the largest probability (Fukunaga, 1990). Moreover, Artificial Neural Networks uses an assembly of artificial neurons that enables the generation of nonlinear decision boundaries (Bishop, 1995). Decision Trees uses a sequence of binary decisions to select the appropriate class label for a data point (A.K. Jain, R.P.W. Duin, & J. Mao, 2000). Last but not least, Logistic Regression is a specialized form of linear regression that assumes the output of regression can have only two values (Ng & Jordan, 2002).

Note that for each step of classification including feature extraction, feature selection, and training a classifier, there are numerous other techniques that have been used in the literature, but it is not possible to investigate every single approach in this study. Therefore, I intend to perform my analyses based on the most common methods at each stage. In this study, I plan to examine the efficiency of each algorithm to see which set of procedures (e.g., oscillatory power, SWS, Bayesian classification) leads to the highest classification accuracy.

1.4 Distinguishing Episodic Memory States in Real Time Using Machine Learning

There have been several attempts to separate single trial neural responses for events a person is likely to remember from those he is likely to forget using machine learning strategies. For instance, (Noh, Herzmann, Curran, & de Sa, 2014) found that it is possible to predict memory performance successfully based on single-trial EEG either during or even before item presentation during encoding. They designed a visual memory task where the participant had to determine the certainty of his decision about whether an item was old. Subsequently, they extracted CSP features from each frequency band (alpha, beta, etc) for training SVM classifiers. They reached 59.64% average accuracy across 18 participants, where the chance level was 50%. The authors concluded that their findings suggest that this method could provide an affordable and non-invasive way to track learning preparedness to optimally specify the time to present a stimulus.

(Höhne, Jahanbekam, Bauckhage, Axmacher, & Fell, 2016) designed a word recognition task using intracranial EEG recorded from epilepsy patients. They found that oscillatory phase signals and differences between these signals from rhinal and hippocampal cortex could discriminate the good from poor memory performance with an average accuracy of 69.2% using an SVM classifier. In another intracranial EEG study of epilepsy patients, (Ezzyat et al., 2018) designed a delayed free recall memory task. For each trial, they extracted averaged spectral power for 8 different frequency ranges across all the electrodes to train penalized logistic regression classifiers. Using this strategy, they achieved 61% accuracy to successfully predict whether the individual was going to later recall the word. Using this method, they were able to then apply a closed-loop training system that improved learning ability across individuals.

In summary, the literature strongly supports that there are various techniques that can be used in the real-time recording to successfully detect at which times the brain is functioning well or poorly to learn the stimulus.

1.5 Limitations of Previous Studies

Although there have been several efforts to discriminate remembered trials from forgotten ones using real-time signals, the progress in reaching high accuracy is still in primary stages and there is a lot of room for improvement. For instance, (Noh et al., 2014) reached 59.64% and (Ezzyat et al., 2018) reached 61% average accuracy. However, these performance values are not high enough to move forward in designing a reliable, practical system that can successfully predict preparedness for memory encoding to improve memory performance. Moreover (Höhne et al., 2016) reached an average accuracy of 69.2% which is higher than studies of (Noh et al., 2014) and (Ezzyat et al., 2018).However, it should be noted that in this study, the data used for classification was intracranial EEG which has much less noise than scalp EEG hence reaching higher accuracy is not surprising. In addition, the experiment was conducted using only epilepsy patients hence the results might not be generalized to healthy people.

The studies mentioned above have some similar limitations that are worth mentioning. First and foremost, all of them used only a small number of data features. To be more specific, (Noh et al., 2014) used only CSP features across different frequencies, (Höhne et al., 2016) performed classification using solely phase features, and (Ezzyat et al., 2018) extracted only power values. Extracting many types of features and selecting those that best differentiate the classes of interest is potentially important for improving classification given the complexity of EEG signals and difficulty separating largely similar cognitive brain states. That is, events that are later remembered with high or low confidence likely share many perceptual and conceptual features and the differences between them may be relatively subtle. Furthermore, (Noh et al., 2014) and (Höhne et al., 2016) used SVM, and (Ezzyat et al., 2018) used Logistic Regression as their classifiers, but there is no guarantee that SVM or Logistic Regression is the optimal classifier. In other words, based on the distribution of the hit and miss trials on the selected features' space, the trials might be more separable using a classifier other than SVM or Logistic Regression, such as LDA, Bayesian classifier, etc. For instance, in **Figure 1**, there is a simple two-dimensional problem where SVM cannot perform well while artificial neural network can separate two classes perfectly.

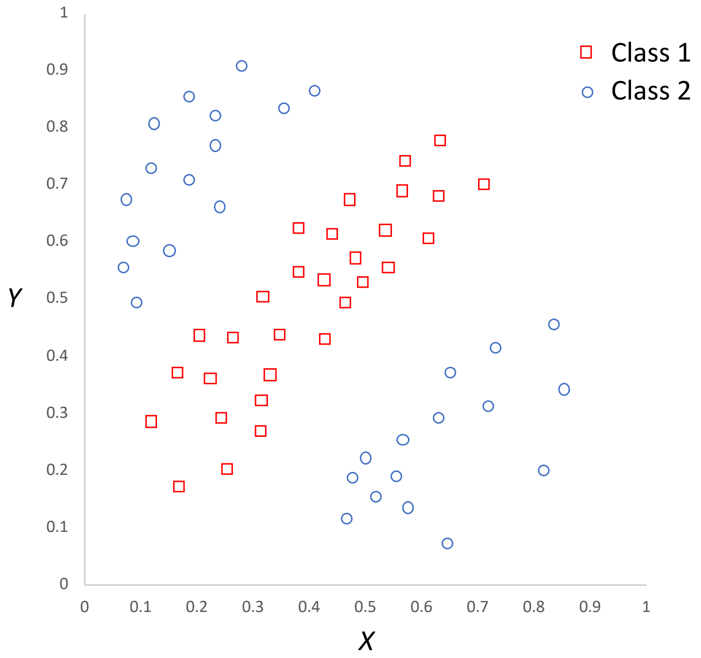

Figure 1- A Simple Example in which SVM fails and Neural Network Succeeds at Classification

Moreover, while the aforementioned studies performed machine learning to perform classification in real time, there have been other studies that performed classification analyses on BCI problems offline (Blankertz et al., 2004; Kaper et al., 2004; Sardouie & Shamsollahi, 2012) although the problems of interest were not related to memory. The goals of these studies were to compare different approaches at a particular stage of classification instead of giving a feedback instantaneously, and this is similar to what I did in this study since my analyses were done offline, not in real time.

In summary, while the progress in predicting subsequent memory performance has been noteworthy, there are still several modifications that can be applied to current routine algorithms to improve classification accuracy.

1.6 Present study

In this study, I directly compared feature extraction methods, feature selection methods, and classifiers for the same dataset in order to determine the combination of procedures that yields the best classification performance. Such information will prove useful for future researchers aiming to apply machine learning classification methods to their cognitive neuroscience questions. I used a previously recorded dataset that was collected in the lab and published (James, Strunk, Arndt, & Duarte, 2016; Powell, Strunk, James, Polyn, & Duarte, 2018; Strunk, James, Arndt, & Duarte, 2017) . I used this dataset for two reasons. First, this dataset consisted of data collected from young, middle aged, and old adults which allows our results since to be generalized across different ages. Furthermore, the rich study allowed us to assess multiple kinds of classification problems: item recognition, context recognition, color perception, and attention. If a particular classification algorithm or set of features performs well for multiple problems, it would have greater generalizability.

In this episodic memory task, attentional demands were manipulated by having participants attend to the relationship between an object and either a color or scene while ignoring the other context feature during encoding. At retrieval, participants were asked whether they saw a specific object during encoding and whether each scene and color context matched the one they had previously encoded. Finally, they made decisions about how confident they were regarding their decisions about the two contexts.

Although EEG data was recorded during both the encoding and retrieval phase of the experiment, I performed the following analyses for the encoding period only. The reason being that the real-time classification studies discussed here all examined memory encoding and optimal brain states for learning, not for retrieving. Future applications that might benefit from the results of this project could be those that aim to use real-time classification to improve learning. Furthermore, the goal of this study was not to classify encoding and retrieval brain states per se, but to compare machine learning based methods for the classification of cognitive states. The main goal of this project was to define a specific, robust procedure that will perform at a high level not only for episodic memory problems but also for other cognitive questions, including those related to perception and attention.

To this end, in the first analysis, I distinguished the trials where the object was later remembered from those where the object was later forgotten. In the second analysis, I was interested in separating trials where both the object and the associated color/scene context were correctly remembered from those where the color/scene was forgotten. Third, I performed a four-class classification between correct and incorrect context memory trials associated with high or low confidence decisions. Although these are unique memory states, phenomenologically, I did not expect accuracy to be very high as the classes overlap in their decision confidence or accuracy. Fourth, I performed three-class classification between perception of the 3 different colors or 3 different scenes. For instance, I classified the trials with green color vs the trials with brown color vs the trials with red color context to examine if it was actually possible to successfully distinguish the trials based on the color perceived during encoding stage. Finally, once I determined the optimal classification procedure for the above-mentioned analyses, I applied it to another dataset to assess its generalizability to other cognitive problems.

2 Methodology

2.1 Feature Extraction

As discussed in the previous chapter, EEG signals measure voltage fluctuations across different electrodes that are put on the scalp and measured at a specific sampling rate. As a result, for each electrode, there is a time series of voltages for each trial. Numerically, for each trial there is a matrix that has the same number of rows as the number of electrodes (N) and same number of columns as the number of voltage samples across the electrodes (T).

I used four types of features in this study. These features are extracted from voltage oscillations and also power of different frequency bands including theta (3-7 Hz), alpha (8-12 Hz), beta (13-30 Hz), and gamma (35-80 Hz). For describing the types of features that were extracted in this study, I will use the term "voltage values" for illustration, but the same rules apply to power representation of the signal. The extracted features in my study are described in the following sections.

2.1.1 Statistical Features

These features include statistical mean, variance, and correlation between signals. To be more specific, as I said before, each trial has a matrix of $N \times T$ for N electrodes and T voltage samples. As a result, for each electrode, the trial has T voltage samples across the recording period. In this study, we divided the T voltage samples into 5 time-intervals (i.e., [0 400], [400 800], …, [1600 200] ms) since the recording period was relatively long for each trial (i.e., 2 seconds). In addition, while EEG signal has a non-stationary nature (i.e., its statistical characteristics change over time), it behaves closer to stationary in the short time intervals and dividing the recording duration into 5 shorter time intervals helps to extract more meaningful features. In the next step, for each electrode and time interval, the mean and the variance of its voltage samples were calculated and used as features, which each led to $5 \times N$ features across the N electrodes and 5 time-intervals. Furthermore, for correlation, we divided electrodes into 4 regions of electrodes, namely frontal right, frontal left, posterior right, and posterior left electrodes. Across time, we took averages from the voltage samples across the electrodes of each region to end up with an average set of voltage samples across time for each of the 4 regions. Pearson correlation between average voltage samples of each pair of distinct regions of electrodes were computed, leading to 30 $(5 \times \binom{4}{2})$ features.

2.1.2 Features based on Entropy

Generally, entropy is a measurement of signal's uncertainty. To illustrate, if the voltage of a signal is always $1\ \mu v$, there is no uncertainty in this signal. On the other hand, if the voltage of a signal changes at every time samples non-periodically, there is some amount of uncertainty in this signal since the voltage of next time sample is not certainly known. There are several choices for definition of entropy, but I used Shannon entropy which in the discrete form is defined as (Shannon, 1948):

$$H(x) = -\sum_{x} p(x) \log_2\big(p(x)\big)$$

Where $p(x)$ denotes the probability density that x occurs. In this case, the entropy is maximum when the probability distribution is uniform.

To obtain the probability distribution of the signal, the distribution can be estimated based on the occurrence of different voltage values in each time sample of the signal for each electrode.

2.1.3 Model-based Features

There are some situations where the EEG signal can be modelled in a specific form, the parameters of the model are estimated and used as a set of features. There are several models that have been used in the literature, but in this study, I used the most common one which is Autoregressive model (AR) (Lotte et al., 2017). To give an insight about the interpretation of AR coefficients, I should note that AR model represents the predictability of brain activity at the current time sample, something we measure using voltages of EEG signals, based on the neural activity of most recent previous time samples. The predictability of a participant's neural activity might differ based on whether he is going to later remember or forget an information, which justifies using the AR parameters as features.

In the AR model, which is applied to EEG signals, the voltage of the signal at each time is considered as a linear combination of the voltages of the signal at p previous times, in addition to a white noise:

$$x(n) = \sum_{i=1}^{p} \alpha_i x[n-i] + u[n]$$

In this case, the α_is are the parameters of the model and can be used as a set of features. I used $p = 4$ which is the most common value used in the literature (Sardouie & Shamsollahi, 2012).

2.1.4 Features based on Common Spatial Pattern (CSP)

CSP filter is a spatial filter that tries to separate the trials of the two classes in the best possible way according to their updated variances which will be used as features (Ramoser et al., 2000)In other words, when CSP applies its filters to current voltage series of electrodes, the outcome is a transformed voltage series for each electrode. CSP filter is designed in such a way that when applied to data, for a specific electrode, if the variances of trials that belong to first class are high, the variances of the trials that belong to second class will be low and vice versa.

To be more specific, initially, the normalized covariance matrix for each trial $X^{(i)}$ is calculated using the following formula:

$$C^{(i)} = \frac{X^{(i)}X^{(i)^T}}{trace\left(X^{(i)}X^{(i)^T}\right)}$$

$X^{(i)^T}$ denotes the transpose of $X^{(i)}$ where the rows of $X^{(i)^T}$ are the columns of $X^{(i)}$ and the columns of $X^{(i)^T}$ are the rows of $X^{(i)}$. Moreover, the trace of a matrix is the sum of its diagonal elements. In the second step, the $C^{(i)}$s that belong to the first class are averaged and denoted by $\overline{C_1}$. $\overline{C_2}$ is calculated similarly. Afterwards, the following optimization problem should be solved:

$$w = \arg\max_{a} \frac{w^T\overline{C_1}w}{w^T\overline{C_2}w}$$

To reach this goal, first, matrix P should be found so that:

$$P(\overline{C_1} + \overline{C_2})P^T = I$$

Then S_1 and S_2 are defined as follows:

$$S_1 = P\overline{C_1}P^T , S_2 = P\overline{C_2}P^T$$

By this transition, S_1 is a diagonal matrix with the eigen values sorted from highest to lowest (λ_i) while S_2 is a diagonal matrix with the eigen values sorted from lowest to highest ($1 - \lambda_i$). The eigen values of S_1 and S_2 are directly related to the variances of the trials of the first class and second class respectively for the new transformed electrodes.

From generalized eigenvalue decomposition, we have:

$$\exists R, D: S_1 = RDR^T, S_2 = R(I - D)R^T$$

Finally, the proposed filter can be calculated as:

$$W = R^T P$$

In the next step, the initial matrix of the experiment $X^{(i)}$ is transformed by the projection matrix to form the matrix $Z^{(i)}$:

$$Z^{(i)} = WX^{(i)}$$

The rows of W are the common spatial patterns and they apply a spatial filter to the voltages of all electrodes. In other words, for each trial, a spatial filter computes a linear combination of voltage series of all electrodes to create a transformed voltage series for each electrode. Generally, only some rows from the beginning of the matrix (where the eigen values for the first class are high and the eigen values for the second class are low) and the same number of rows from the end of the matrix (where the eigen values for the first class are high and the eigen values for the second class are low) are used. The diagonal elements of covariance matrix $Z^{(i)}$ or their logarithms are used as features to discriminate between the two classes. Note that the diagonal elements of covariance matrix $Z^{(i)}$ are the updated variances.

2.2 Feature Selection

After statistical, entropy-based, model-based, and CSP-based features have been extracted from every single channel, a subset needs to be selected that are useful for classification. This step is often skipped in cognitive studies since the researchers rely on a priori knowledge to extract only the features that they believe are informative based on the literature. However, it is possible that some very useful information will be missed, and performance will be suboptimal. On the other hand, by extracting many features, there could be an added time cost to the analysis. Moreover, the machine learning literature strongly supports the essence of feature selection for improving classification performance and avoiding overfitting (Dias, Jacinto, Mendes, & Correia, 2009; Koprinska, 2009; Lotte et al., 2017). Overfitting is a situation that occurs when the classifier is designed too specifically for the amount of data that is available. In other words, the classifier is trying to be so perfect that it is learning even the noise of the data and might be unable to classify new data points (Burnham & Anderson, 2003). On the other hand, underfitting is the case where the classifier is designed too generally for the current data points and it does not capture some fundamental properties of the dataset. For the purpose of illustration, **Figure 2** explains the concepts of overfitting and underfitting.

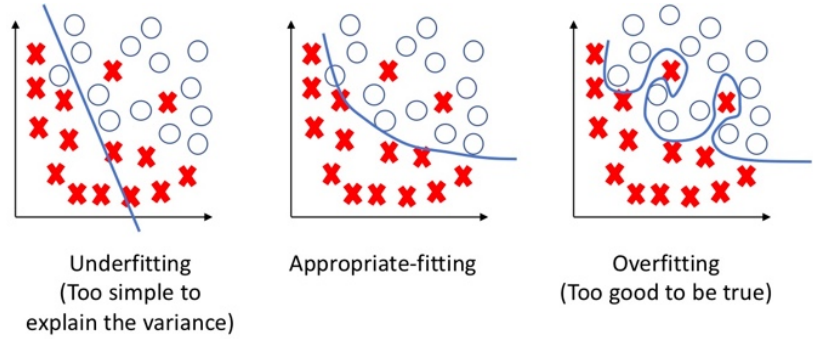

Figure 2- concepts of overfitting and underfitting

As a result, it's essential to choose only the features that are beneficial for classification. There are a few approaches to select the best features, but I'm going to focus on two of these strategies: filter and wrapper methods.

2.2.1 Filter Methods

In filter methods, evaluation of features is independent of the classification algorithm. The individual features will be evaluated based on their information content, such as interclass distance, information-theoretic measures, and etc. There are many evaluation metrics, but we focus on Fisher's criterion since it is one of the most common approaches (Gu, Li, & Han, 2011). Suppose that there are n_1 trials that belong to the first class and n_2 trials that belong to the second class. For each trial, all kinds of features stated above have been extracted for each channel. As a result, for a specific feature f there are $n_1 + n_2$ values. The average and variance are computed from the values of f for the n_1 trials and are denoted by μ_1 and σ_1 respectively. The same is performed for the values of f for the trials that belong to second class. Furthermore, the mean and variance of values of f across all $n_1 + n_2$ trials are calculated and denoted by μ and σ respectively. The Fisher score for the feature f can be computed using the following formula:

$$score(f) = \frac{(\mu_1 - \mu)^2 + (\mu_2 - \mu)^2}{\sigma_1^2 + \sigma_2^2}$$

To give an intuition about this criterion, a feature is useful if the values of the first class are largely different from the values of the second class and the variance for those values is low for each class. To illustrate, in **Figure 3** there is a problem where there are only 2 features X and Y. By Fisher's criterion it can be seen that Y is a more useful feature. This makes sense since the Y values for the red class are all between [0.57 0.79] while the Y values for the blue class are all between [0.21 0.50]. Thus, if a classifier is told that a Y value for new anonymous trial is 0.65, it can tell with high confidence that this trial belongs to the red class. However, the X values for the red and blue class have substantial overlap in the range of 0.27 to 0.60 in the figure and if the classifier is told that an X value for a new anonymous trial is 0.5, it will perform at chance level and cannot really tell whether this trial belongs to the blue or the red class.

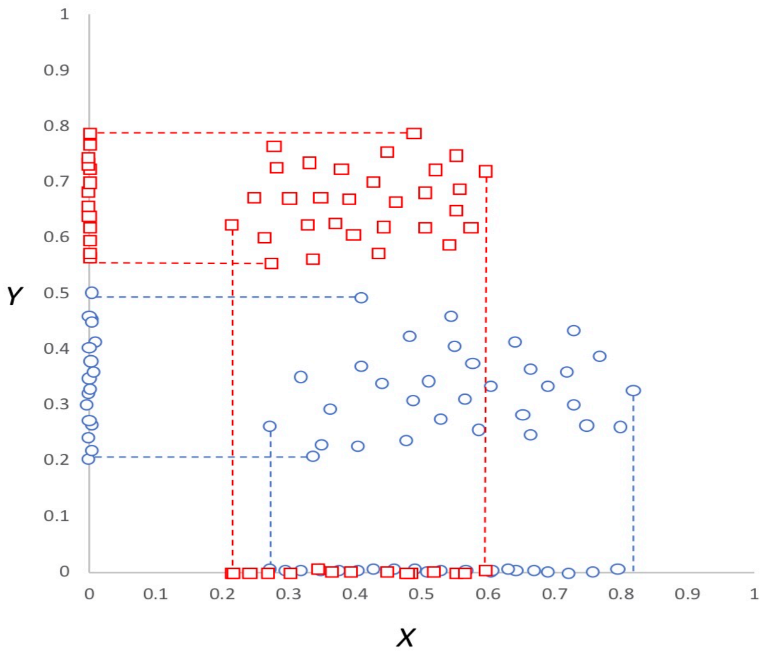

Figure 3- An Illustration for Fisher's criterion

After the Fisher score is computed for each feature, the features are sorted by their scores. Those features with highest scores are selected. The number of features that should be used for training the classifier depends on the domain knowledge and there is not a determined number to use for every analysis. A rule of thumb suggests selecting about the squared root of the number of available trials (Hua, Xiong, Lowey, Suh, & Dougherty, 2004). For instance, if there are 200 trials, less than 20 features should be selected for training the classifier. However, in this study, I delved into the details about how the performance of the classifier is related to the number of selected features.

2.2.2 Wrapper Methods

While the evaluation of individual features is independent of the classifier for filter methods, wrapper methods use criteria related to the classification algorithm to assess the usefulness of the extracted features. In other words, these methods use a pattern classifier that appraises feature subsets by their predictive accuracy (rate of recognition on test data) using cross-validation or statistical resampling.

In order to search for the optimal subset of features, one approach is to search through all subsets of features exhaustively. Specifically, if there are n extracted features, this exhaustive strategy examines all of the $\binom{n}{m}$ possible subsets that include m features. Subsequently, once all of the subsets of all sizes ($m = 1, 2, \ldots, n$) are assessed, the subset that performs the best according to the criterion function will be selected as the optimal subset. However, the number of possible subsets increase at a combinatorial rate by increasing the number of extracted features which makes exhaustive search computationally expensive and impractical. As a result, heuristics are used in practice to perform the searching process much faster even though they cannot guarantee optimality. While there are various heuristic search techniques that are used in the literature, we used the most common strategy which is known as sequential forward selection (SFS) (Mitchell, 1997).

Initially, the sequential forward selection chooses the best single feature according to a specific criterion function (five-fold cross validation performance in this study). In the next step, pairs of features are formed by using one of the remaining features and this already chosen best feature, and the best pair is determined. Subsequently, the triplets of features are formed using one of the remaining features and the two already chosen best features, and the best triplet is specified. This process continues until a subset of predefined number of features are chosen.

2.2.3 Combination of Filter and Wrapper Methods

Regarding the wrapper methods, I described how it is expensive to exhaustively search through the best subset of features and I also mentioned that heuristic search can be used to decrease the running time to find a relatively good subset of features to use for classification. However, while heuristic search methods such as SFS are faster algorithms compared to exhaustive searching, it can still be impractical to utilize them if there are many features to search through.

As a result, a much faster approach is to evaluate the features initially using the filter method and then search through only the features that have received high scores using the filter method. For instance, if there are 10000 extracted features in an analysis, running SFS on the whole set of features is not practical and an efficient way is to evaluate the features using filter methods and then search through only the top 500 features. This method saves a lot of running time and it probably performs as well as searching through all features. To be more specific, if a feature is not among the top 500 features based on the statistical criterion, it probably cannot be an effective feature for classification, and disregarding it will not affect the performance dramatically (Mitchell, 1997). In this study, I have used the filter approach as well as the combination of filter and wrapper methods.

2.3 Training a Classifier

After the best features are selected, the training data (the EEG data for which the related labels are known) will be projected to the space of the selected features and then they will be used to conduct a learning process. For example, in the classification problem associated with **Figure 4**, only two features are selected, and the EEG data of each training trial will be projected into that two-dimensional space in order to train the classifier. In the next step, the trials that belong to test data (the EEG data that the labels are unknown) will also get projected in the same two-dimensional space and should be classified according to the obtained classifier at training stage.

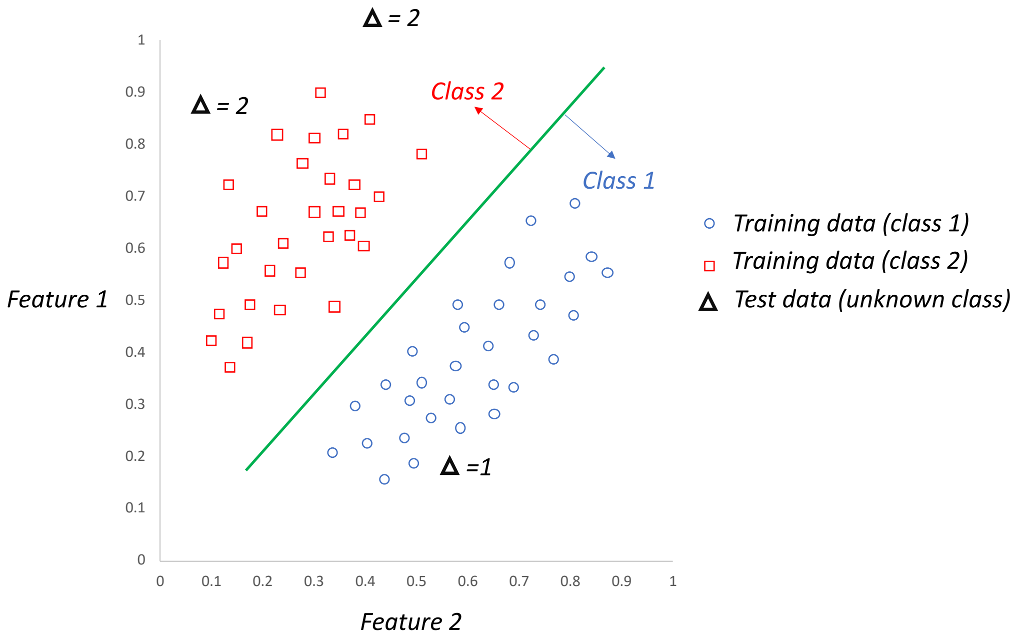

Figure 4- Training a classifier and testing it on the test data. In the first step, the trials will be projected to the space of selected features which is a two-dimensional space in this case. Subsequently, the classifier will be trained to separate the trials of the two classes in the best way possible. It learns that if it uses the green line as the threshold (i.e., all the trials on the right side of the line (which exceed the threshold) are associated to one class while the trials on the other side of the line are associated to the other class), the training trials are maximally separable based on the class they belong to. In the next step, once the classifier receives the test trials with unknown labels, it first projects their EEG data to the same selected features space and then predicts

the labels based on the threshold that it previously had learned. If a test trial exceeds the threshold, it will be labeled as a member of class 1, and if not, it will be labeled as a member of class2.

2.3.1 Binary Classification

In most cases, the problem has only two different classes. For instance, in my project the two classes might be the brain states where the individual remembers or forgets the previously presented item. Most of the classifiers are specialized for binary classification. There are various classifiers that are different in their computation time based on their methodology. In the following section, μ_i and Σ_i are the mean and the covariance matrix of the i'th class respectively:

2.3.1.1 Support Vector Machine (SVM)

SVM tries to separate the trials of each class from the trials of the other class using a linear hyperplane (Burges, 1998). The linear hyperplane should be specified in a way that maximizes the distance from the hyperplane to the closest trial from each class. This property of the hyperplane leads to higher separability of the trials in two classes in the test data which results in higher classification accuracy.

Let's assume that the training trials are represented by:

$$\Gamma = \{(x_1, c_1), (x_2, c_2), \dots, (x_n, c_n)\}$$

Where $c_i \in \{-1,1\}$ determines the class of each trial. Suppose the equation for optimal hyper plane is written as:

$$w^T x + b = 0$$

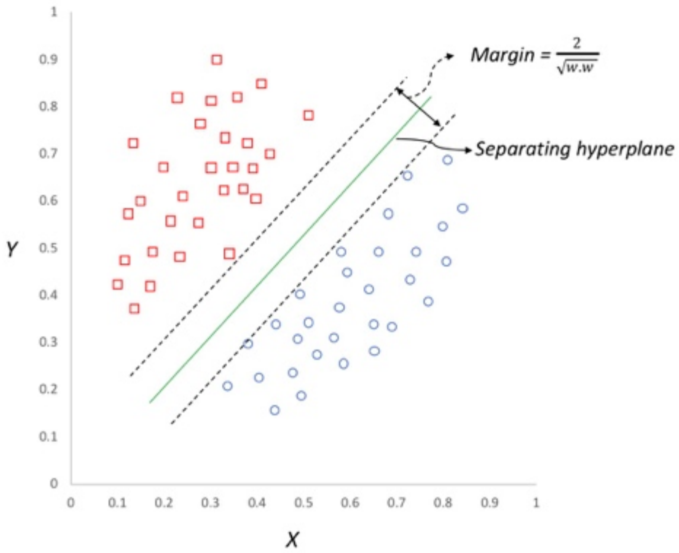

Figure 5- An Illustration about Support Vector Machine Classifier

It can be shown that the margin width is equal to:

$$M = \frac{2}{\sqrt{w.w}} = \frac{2}{w}$$

In order to increase the separability of trials of the two classes using this hyperplane, the margin width should be maximized which leads to minimizing w. This can be done by solving a set of quadratic equations that are subject to some set of inequality restrictions.

However, in most cases it is not possible to find a linear hyperplane that can separate the trials of the two classes perfectly. As a result, in order to solve the optimization problem, one should define an error function in classification which somehow represents the sum of the distances of the hyperplane to the trials that are classified wrongly. Thus, in this situation the optimization problem tries to maximize the margin width while it is minimizing the error function.

Furthermore, beside the label that the classifier gives to each trial, it also gives a score for each trial that denotes the level of certainty about the associated label. For instance, if a trial is close to the separator hyperplane, the classifier is not really certain about the label of the trial, compared to a trial that is far away from the separator hyperplane.

2.3.1.2 LASSO

LASSO is a type of regression that tries to define the output (the class) based on a linear combination of the inputs by solving the following optimization problem (Tibshirani, 1996):

$$\min_{\beta} ||y - X\beta||^2 + \lambda ||\beta||_1$$

Where y is the output class and the classes are treated as numbers 1 and 2. It can be shown that regardless of which numbers each class is associated with, the final result will be the same. X is the input of the regression which is the vector of the selected features. In addition, $||\beta||_1$denotes the nonzero coefficients for the linear regression. As a result, not only the goal is to minimize the mean squared error of the output estimation, it is also desirable to reduce the number of features that will be used for the output estimation.

Eventually, the output value of the regression will be real numbers for each trial and by setting the threshold equal to 1.5, if the output value is higher than the threshold, the trial will get the label of class 2 and if the output value is less than the threshold, the trial will get the label of class 1.

2.3.1.3 Logistic Regression

Logistic regression calculates the probabilities for classification problems that have two possible outcomes. It's a modified version of the linear regression model that is used for binary classification problems. One might wonder why it's not always appropriate to use linear regression for classification and treat the two classes as numbers (such as 0 and 1). One issue is that the output of the linear regression does not represent the actual probabilities that a trial belongs to each class. Instead, it fits the best linear hyperplane that minimizes the distances between the hyperplane and the points. As a result, linear regression simply interpolates between the points, and it's not possible to interpret the outcome values as probabilities which indicates there is not a meaningful threshold that separates one class from the other. Furthermore, using linear regression, it's possible to receive outcomes that are below 0 or above 1. In order to address these shortcomings, logistic regression is used for classification. Rather than fitting a linear hyperplane, logistic regression squeezes the output of a linear equation between 0 and 1 by using the logistic function which is defined as:

$$logistic(x) = \frac{1}{1 + e^{-x}}$$

It is straightforward to switch to logistic regression once the linear regression model is obtained. In the linear regression model, the relationship between outcome and features are modelled using a linear equation:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m$$

For classification, the probabilities should be between 0 and 1, so the right side of the equation will be wrapped into the logistic function so that the output can only vary from 0 to 1:

$$p\left(\hat{y}^{(i)} = 1\right) = \frac{1}{1 + \mathrm{e}^{-(\beta_0 + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \cdots + \beta_m x_m^{(i)})}}$$

In the end, if the probability that a trial belongs to one class is more than 0.5, that class will be chosen as the label for that trial and otherwise, the other class will be chosen as the label for that trial.

2.3.1.4 Bayesian Classifier

This classifier performs based on the conditional probability of occurrence in each class (Fukunaga, 1990).In mathematical terms:

$$\begin{cases} if\ P(x|c_1) > P(x|c_2) \quad then \quad x \in c_1 \\ if\ P(x|c_1) < P(x|c_2) \quad then \quad x \in c_2 \end{cases}$$

Where x is a data point, c_1 and c_2 represent the classes 1 and 2, and P indicates the probability. In order to find the class that each trial belongs to, an estimation of the probability distribution of each class is needed. However, since no information about the true probability distribution and its parameters is known, it is assumed that it follows a normal distribution.

By assuming s normal distribution, the following parameters for class i are defined:

$$A_i = -0.5 \times \Sigma_i^{-1}$$

$$b_i = \Sigma_i^{-1} \times \mu_i$$

$$c_i = -0.5 \times \mu_i^T \times \Sigma_i^{-1} \times \mu_i - 0.5 \times \log|\Sigma_i| + \log(P_i)$$

$$d_i(x) = x^T \times A_i \times x + b_i^T \times x + c_i$$

Where Σ_i and μ_i are the covariance matrix and the mean vector of the data points that belong to class i and $\Sigma_i^{-1}$is the inverse matrix of Σ_i. By the above definitions, it can be inferred that:

$$\begin{cases} if\ d_1(x) > d_2(x) \quad then \quad x \in c_1 \\ if\ d_1(x) < d_2(x) \quad then \quad x \in c_2 \end{cases}$$

2.3.2 Multiclass Classification

While there are many situations in which a problem has two classes (e.g., remembered vs. forgotten; move left vs. move right), there also exist various cases where it is desirable to distinguish multiple brain states. For the purpose of illustration, in this study, each individual is presented with a colored square that might be either green, red, or brown. If we want to understand how well subjects can attend to and encode color, we face a three-class classification problem. There are many techniques for solving multiclass classification problem and I plan to use two of these methods:

2.3.2.1 Classification Based on the Voting Method

This strategy is a generalization of binary classification to multiclass classification problem. As stated before, in binary classification including SVM and LDA, the classifier gives a score and label for each trial indicating how certain the classifier is about the associated label. The score is a value between 0 to 1 and can be interpreted as a probability that the trial belongs to the selected class. It can be easily inferred that the associated class has a score p_i which is larger than 0.5 and the other class has a score $1 - p_i$ which is less than 0.5.

In the voting method, one against others approach is used. To be more specific, consider a three-class problem. In the first classification, the trials of the second and third class are combined to create a new merged class and now, the problem has turned to a binary classification problem, where the classes consist of the first class (1) and the new merged class (2 and 3). The same thing is done for class 2 against classes 1,3 combined and class 3 against classes 1,2 combined. After these three classifications, each trial has obtained three labels and scores. In the next step, the results are gathered in the following table:

Table 1- Voting Method to Generalize the Binary Problem to Multiclass Problem

	Class 1	Class 2	Class 3
Class 1 vs others	p_1	$1 - p_1$	$1 - p_1$
Class 2 vs others	$1 - p_2$	p_2	$1 - p_2$
Class 3 vs others	$1 - p_3$	$1 - p_3$	p_3
Total scores	$1 + p_1 - p_2 - p_3$	$1 - p_1 + p_2 - p_3$	$1 - p_1 - p_2 + p_3$

In this stage, each class has a score which means how much is it possible that the trial belongs to that class. In order to associate a class to the trial, the class with highest score compared to other two classes will be picked. Note that although my example was for the three-class case, this technique works regardless of the number of classes in the multiclass problem (Bishop, 1995).

2.3.2.2 Classification Using Decision Trees

In this method, a decision tree is used and at each node of the tree, we face a binary classification problem. At the root of the tree, the classes are partitioned into two groups and the classification is performed between two groups of classes. At each new node, the groups (if they consist of more than one class) are partitioned into two subgroups and the classification is performed again. The same process is repeated until each leaf of the tree represents a single class. For illustration, consider the four-class case:

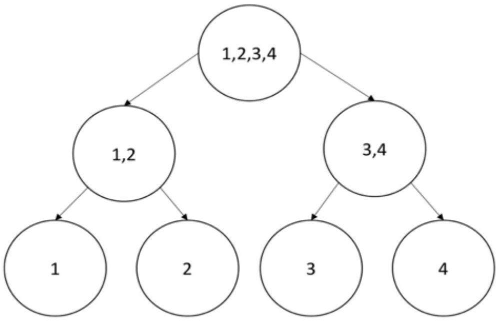

Figure 6- Binary Decision Tree

In the first step, the five-fold cross validation accuracy between each pair of single classes i and j is computed and the accuracy is denoted by μ_{ij} which represents the separability of the two classes. In the next step, if the classes at each node (M) are divided into two partitions of $\{M_1, M_2\}$, the score of the partition shows how much this partition can separate each class from other classes eventually. The score of a partition can be computed using the following equation:

$$Score_{\{M_1,M_2\}} = \frac{\frac{\sum_{i\in M_1, j\in M_2}\mu_{ij}}{|M_1||M_2|}}{\frac{\sum_{i,j\in M_1, i\neq j}\mu_{ij} + \sum_{i,j\in M_2, i\neq j}\mu_{ij}}{\binom{|M_1|}{2} + \binom{|M_2|}{2}}}$$

As a result, at each node with more than one class, the classes are divided into two partitions using the partition with highest score (Mirjalili, Sardouie, & Samiee, 2019).

2.4 Measures Used to Assess a Classifier's Performance

Various measurements are used to evaluate a classifier's performance. Before explaining different criteria, it is important to note that the decision to assign the positive and negative labels to the conditions of interest is up to the researcher. Here, I call the remembered trials as positives and forgotten ones as positive.

The most common criterion to assess a classifier's performance, is its accuracy, which is defined as below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

True positive (TP) reflects the number of correctly labeled positive trials (e.g., remembered trial labeled "remembered"). True positive (TN) reflects the number of correctly labeled negative trials (e.g., forgotten trial labeled "forgotten"). False positive (FP) reflects the number of incorrectly labeled negative trials (e.g., forgotten trial labeled "remembered"). False negative (FN) reflects the number of incorrectly labeled negative trials (e.g., remembered trial labeled "forgotten"). These counts can be used to calculate sensitivity or "true positive rate" and specificity or "true negative rate"

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

Specificity and Sensitivity can be combined into a single measurement known as the geometric mean or G-Mean:

$$G - Mean = \sqrt{TPR \times TNR}$$

While the accuracy of a classifier is the most common way to describe the classifier's performance, it is not perfect. For instance, imagine a classification problem with 90 positives and 10 negatives. If the classifier labels all of the trials as positives, accuracy will be 90%, but specificity is low, as no negative trials are detected. It is essential that the classifier has a relatively high specificity *and* sensitivity. Depending on the problem of interest, we might require very high specificity. For the purpose of illustration, suppose there is a very rare cancer that 1 out of 10000 may have. A highly sensitive test is needed in order to identify and treat the individuals with that cancer. In this case, when a classifier is trained, it learns that in order to minimize its error, the best approach is to label all of the trials as hits and this way the classifier accuracy will be 99%. But the problem is that the cancer cases cannot be detected using this method. Although in theory, it is ideal to have equal numbers of positive and negative trials in order to reduce biased classification, in actuality, this is seldom case, as illustrated in this cancer example.

There are two different solutions for this issue. Firstly, a similar approach to bootstrapping can be performed (Efron & Tibshirani, 1993). To illustrate it with an example, 100 classifiers will be trained where each of them uses an equal number of positive and negative trials. For instance, if a classification problem consists of 20 hits and 180 misses in the training set, 20 misses will be sampled randomly for 100 times and each time, a classifier will be trained using the 20 hits and the 20 sampled misses. Subsequently, each of these 100 trained classifiers will be used on each test trial to predict its label. A vote will then be taken and the label that was assigned by the majority of these 100 trained classifiers (more than 50 of them) to this test trial will be selected as its final predicted label.

While the bootstrapping approach can be very useful, it is computationally expensive since it requires a large number of classification analyses. An alternative strategy that can be used in combination with the wrapper method is to define the wrapper's criterion function based on average cross-validation g-means instead of the typical average accuracy because of the good performance of G-Mean in case of imbalance between the trials of each class (Abdulrauf Sharifai & Zainol, 2020; Mosley, 2013).This ensures the selection of only those features that lead to relatively high TPR and TNR simultaneously. For the present study, the number of recognized events is often greater than the number of forgotten ones. Consequently, my goal was to have the highest possible G-Mean instead of accuracy in order to ensure high levels of both specificity and sensitivity.

3. Experiments

3.1 Participants

The data included here was obtained from participants that had participated in three previously published EEG studies (James et al., 2016; Powell et al., 2018; Strunk et al., 2017). The participants consisted of 22 young (18–35), 15 middle-aged (35-60), and 21 older (60– 80) healthy, right-handed adults. All subjects were native English speakers and had normal or corrected vision. Participants were compensated with course credit or $10/hour and were recruited from the Georgia Institute of Technology and surrounding community. None of the participants reported any neurological or psychiatric disorders, vascular disease, or using any medications that impacts the central nervous system. Participants completed a standardized neurological battery and were excluded if their scores were above or below two standard deviations of the group mean. All participants signed consent forms that are approved by the Georgia Institute of Technology Institutional Review Board. Three older participants were excluded in this study since EEG recordings from one or more of the encoding blocks were not available because of computer malfunction.

3.2 Materials:

Four hundred thirty-two grayscale images of objects were selected from the Hemera Technologies Photo-Object DVDs and Google images. At encoding, 288 of these objects were presented, in half of them the attention was directed to a color and in the other half directed to a scene. Each grayscale object was presented on the center of the screen with white background. In the left and right of the object there was a color square and scene. The locations of the context features (e.g., color or scene) were counter-balanced across blocks so that they were shown an equal number of times the on the right and left-hand side of the object in the center. For each encoding trial, participants were instructed to focus on either the colored square or the scene, which served as the target context for that trial. The potential scenes included a studio apartment, cityscape, or island. The potential colored squares consisted of green, brown, or red. Each of the 432 context and object pictures spanned a maximum vertical and horizontal visual angle of approximately 3°. During retrieval, all 288 objects were included in the memory test in addition to 144 new object images that were not presented during encoding. Study and test items were counterbalanced across subjects.

3.3 Procedure

Figure 7 illustrates the procedure used during the study and test stages. Before the beginning of each phase, participants were provided instructions and given 10 trials for practicing. For the study stage, participants were asked to make a subjective yes/no assessment about the relationship between the object and either the colored square (i.e., is this color likely for this object?) or the scene (i.e., is this object likely to appear in this scene?). Instructions for the task specified that on any specific trial the participant should pay attention to one context and ignore the other context.

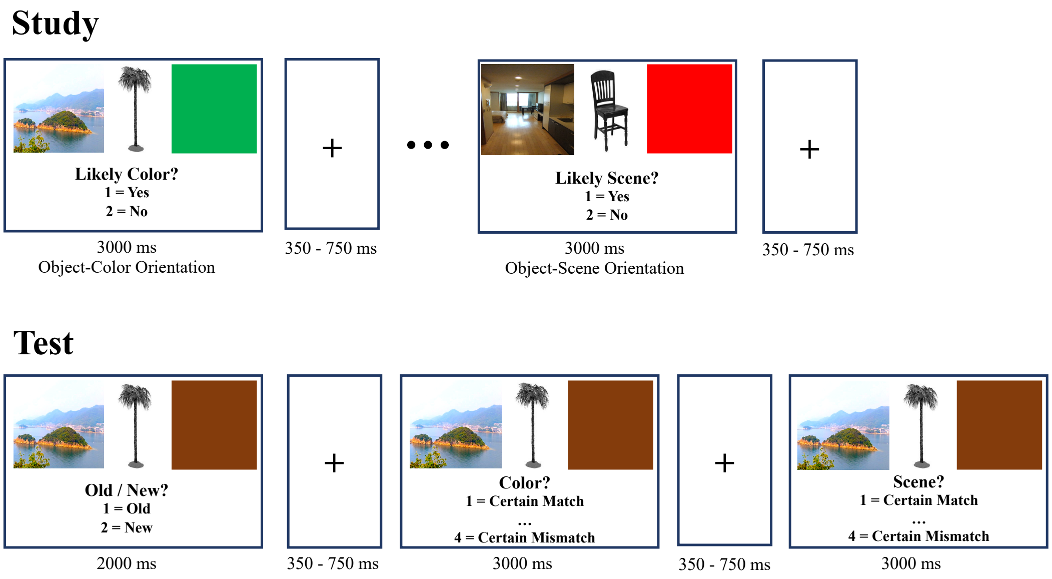

Figure 7- Task Design for Study and Test Phase

Within the study phase there were four blocks where each block consisted of four mini-blocks and each of them included 18 trials, as can be seen in **Figure 8** (Powell et al., 2018). In advance of beginning each mini-block, participants were provided a prompt (e.g., "Now you will assess how likely the color is for the object" or "Now you will assess how likely the scene is for the object"). Since prior evidence has suggested that memory performance in older adults is more disrupted when they have to switch between two distinct kinds of tasks (Kray & Lindenberger, 2000), mini-blocks were used to orient the participant to which context they should pay attention to in the upcoming trials. Moreover, it decreases the task demands of having to switch from judging one context (e.g., color) to judging the other every time (e.g., scene). Additionally, each trial in a mini block had a reminder prompt presented underneath the pictures during study trials (see **Figure 7**).

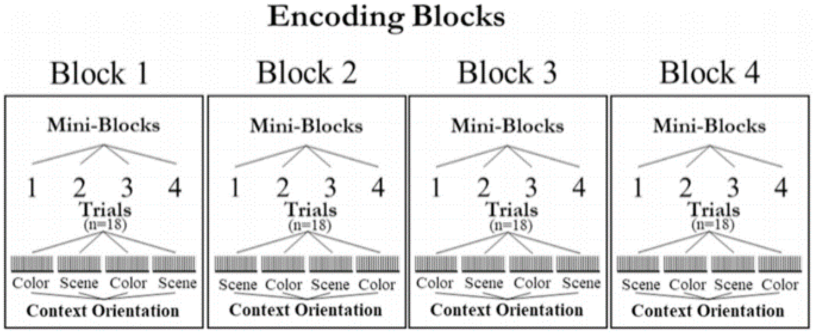

Figure 8- Mini-blocks Used in Cross-validation (n-1). Four Mini-blocks per Block, 18 trials per Mini-block.

At test stage, participants were presented with both old and new objects. Similar to the study phase, each object was shown by both a scene and a colored square. For each object, the participant initially decided whether it was an old or a new image. If the participant detected the object as a new one, the next trial began after 2000 ms. If participants stated that it was old, then they were asked to make two additional assessment about each context feature and describe their certainty about their judgment (i.e., one about the colored square and another about the scene). The order of the second and third questions was counterbalanced across participants. For old items, the pairing was set so that an equal number of old objects were presented with: (1) both context images matching those presented at encoding stage, (2) only the color matching, (3) only the scene matching, and (4) neither context images matching. Responses to the context questions were made on a scale from 1 (certain match) to 4 (certain mismatch). Totally, there were four study and four test blocks. Young adults finished all four study blocks before the four test blocks. For older adults, to better equate item memory performance with young adults, the memory load was halved so that they finished a two- block study-test cycle twice (two study, two test, two study, two test). Both younger and older adults finished a short practice of both the study and test blocks in advance of beginning the first study block. As a result, both younger and older adults knew of the following memory test.

3.4 EEG recording

Continuous scalp-recorded EEG data was recorded from 32 Ag-AgCl electrodes using an ActiveTwo amplifier system (BioSemi, Amsterdam, Netherlands). Electrode position is based on the extended 10–20 system (Nuwer et al., 1998). Electrode positions consisted of: AF3, AF4, FC1, FC2, FC5, FC6, FP1, FP2, F7, F3, Fz, F4, F8, C3, Cz, C4, CP1, CP2, CP5, CP6, P7, PO3, PO4, P3, Pz, P4, P8, T7, T8, O1, Oz, and O2. External left and right mastoid electrodes were used for referencing offline. Two additional electrodes recorded horizontal electrooculogram (HEOG) at the lateral canthi of the left and right eyes and two electrodes placed superior and inferior to the right eye recorded vertical electrooculogram (VEOG). The sampling rate of EEG was 1024 Hz with 24-bit resolution without high or low pass filtering.

3.5 EEG preprocessing

Offline analysis of the EEG data was performed in MATLAB 2015b using the EEGLAB (Delorme & Makeig, 2004), ERPLAB (Lopez-Calderon & Luck, 2014), and FIELDTRIP (Oostenveld, Fries, Maris, & Schoffelen, 2011) toolboxes. The continuous data was down sampled to 256 Hz, referenced to the average of the left and right mastoid electrodes, and band pass filtered between .5 Hz and 125 Hz. The data was then epoched from –1000 ms prior to stimulus onset to 3000 ms. The time range of interest was begun from – 300 ms to 2000 ms, but a longer time interval is required to account for signal loss at both ends of the epoch during wavelet transformation. Each epoch was baseline corrected to the average of the whole epoch, and an automatic rejection process deleted epochs in which a blink occurred during stimulus onset or epochs with extreme voltage shifts that spanned across two or more electrodes. The automated rejection processes identified epochs with the following parameters in the raw data: 1) The voltage range was greater than 99th percentile of all epoch voltage ranges within a 400 ms time interval (shifting in 100 ms intervals across each epoch). 2) The linear trend slope was higher than the 95th percentile of all epoch ranges with a minimum R2 value of 0.3. 3) The voltage range was larger than 95th percentile of all epoch voltage ranges within a 100 ms time interval (shifting in 25 ms intervals across each epoch), between –150 and 150 ms from stimulus onset for frontal and eye electrodes only. Then an independent component analysis (ICA) was run on all head electrodes for identifying additional artifacts highlighted by the components. The following parameters were used on the components to reject epochs: 1) The voltage range exceeded 99th percentile of all epoch voltage ranges within a 400 ms time interval (shifting in 100 ms intervals across each epoch). 2) The kurtosis or joint probability was greater than 15 standard deviations within the component or 23 standard deviations of all components for the epoch. To identify activity related to ocular artifacts (i.e., blinks and horizontal eye movements), ICA was run on the first 20 principle components of the head electrodes for the accepted epochs. Components related to ocular artifacts were omitted from the data by visually inspecting the topographic component maps and component time course with the ocular electrodes (Bell & Sejnowski, 1995; Delorme, Sejnowski, & Makeig, 2007; Hoffmann & Falkenstein, 2008). Each epoch was re-baselined to the –300 to –100 ms time period before stimulus onset since the epochs were no longer baselined to a specific time period after deleting components related to ocular activity. This was done solely for the purposes of visual inspection and detection of additional artifacts in each epoch (e.g., amplifier saturation, spiking, extreme values, uncorrected ocular activity), and does not impact the frequency decomposition. If a dataset had a noisy electrode (e.g., larger than 30% of the data required to be rejected), it was deleted from the processing stream and interpolated using the nearby channels to estimate the activity within the bad channel before running the time frequency procedure (Delorme & Makeig, 2004). After all processing stages, about 13% (SD = 8%) of the epochs were removed.

3.6 Frequency decomposition

Each epoch was transformed into a time frequency representation by Morlet wavelets (Percival, Walden, & others, 1993) with 78 linearly spaced frequencies from 3 to 80 Hz, at 5 cycles. During the wavelet transformation, each epoch was decreased to the time interval of interest and down sampled to 50.25 Hz (Cohen, 2014). For the following MVPA analyses, I examined item hit events (i.e., the old objects that the participant identified correctly as old) across both context features (i.e., attend color and attend scene), including all levels of confidence. The average number of trials for younger, middle-aged, and older adults are as follows: Younger (M = 190.50, SD = 41.01); Middle-aged (M = 187.31, SD = 40.24) Older (M = 177.06, SD = 38.56).

3.7 Summary of Classification Methods and Cognitive Problems that were Investigated in this Study

Here is a summary of different cognitive problems that I performed classifications on:

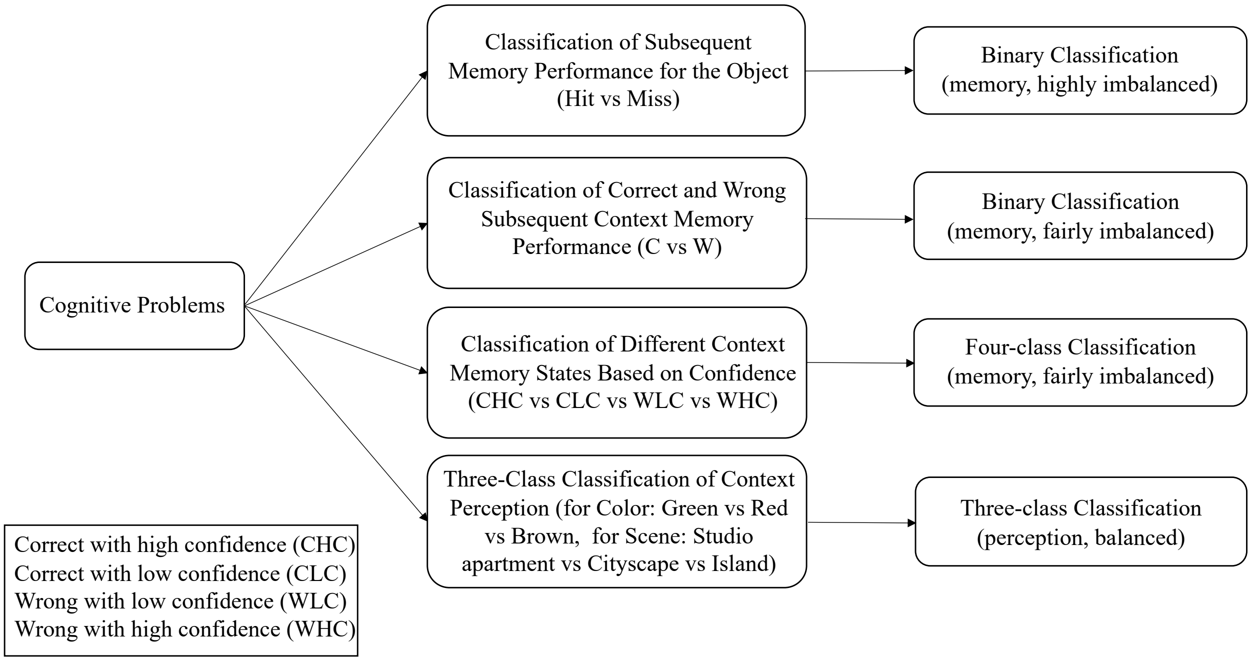
Cognitive Problems
Classification of Subsequent Memory Performance for the Object (Hit vs Miss)
Classification of Correct and Wrong Subsequent Context Memory Performance (C vs W)
Classification of Different Context Memory States Based on Confidence (CHC vs CLC vs WLC vs WHC)
Three-Class Classification of Context Perception (for Color: Green vs Red vs Brown, for Scene: Studio apartment vs Cityscape vs Island)
Binary Classification (memory, highly imbalanced)
Binary Classification (memory, fairly imbalanced)
Four-class Classification (memory, fairly imbalanced)
Three-class Classification (perception, balanced)
Correct with high confidence (CHC)
Correct with low confidence (CLC)
Wrong with low confidence (WLC)
Wrong with high confidence (WHC)

Figure 9- Summary of Cognitive Problems We Intend to Perform Classification on

The average number of hits was 188 (range: [101-244]) while the average number of misses was 63 (range: [27-170]) across participants. Moreover, for color context memory, the average number of high-confidence corrects was 47 (range: [1-146]), average number of low confidence corrects was 52 (range: [1-110]), average number of low confidence incorrects was 48 (range: [2-100]), and average number of high confidence incorrects was 36 (range: [2-72]) across participants. Finally, for the trials where the item was correctly identified as old and the color context was correctly identified as a match/mismatch (regardless of the confidence), on average, the context of 34 (range: [11-47]) of them were red, 37 (range: [13-53]) of them were green, and 36 (range: [10-55]) of them were brown. Importantly, I only performed classification analysis if all of the classes on that problem contained at least 20 trials for that participant.

Furthermore, a summary of the methods that were used in each stage of classification can be found in **Figure 10**:

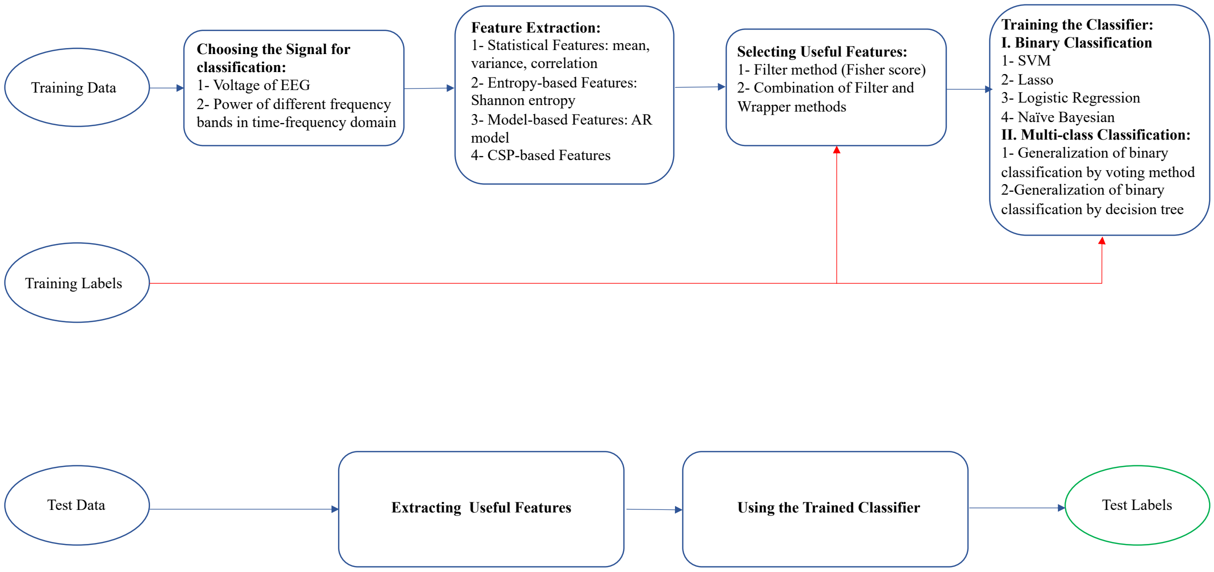
Training Data
Choosing the Signal for classification:
1- Voltage of EEG
2- Power of different frequency bands in time-frequency domain
Feature Extraction:
1- Statistical Features: mean, variance, correlation
2- Entropy-based Features: Shannon entropy
3- Model-based Features: AR model
4- CSP-based Features
Selecting Useful Features:
1- Filter method (Fisher score)
2- Combination of Filter and Wrapper methods
Training the Classifier:
I. Binary Classification
1- SVM
2- Lasso
3- Logistic Regression
4- Naïve Bayesian
II. Multi-class Classification:
1- Generalization of binary classification by voting method
2-Generalization of binary classification by decision tree
Training Labels
Test Data
Extracting Useful Features
Using the Trained Classifier
Test Labels

Figure 10- A Summary of the Proposed Methodology

4. Results

As stated in the Methods chapter, each classification problem consists of three main stages: feature extraction, feature selection, and training the classifier using the selected features. I used both the voltage of EEG signal and the power of 4 different frequency bands (theta, alpha, beta, and gamma) in the time-frequency representation. Since the signals represented a relatively long period of recording (2 seconds), I broke the signals into 5 segments, each segment representing 400ms of EEG recording. For each of the 5 time bins, I then extracted several types of features including the mean, variance, entropy of the signal of each electrode (32*3 features), the correlation between the signals of different electrode regions (6 features), the parameters of the autoregressive model ($p = 4$) obtained from the signal of each electrode (4*32 features), as well as the features extracted by applying the CSP filters to the signal (32 features). As a result, there were 262 features for each time segment resulting in 1310 features for each of the 4 frequency bands and the voltage leading to 6550 features. Subsequently, I evaluated each of these 6550 features using the Fisher's criterion. For the filter method, I selected the 10 features with the highest Fisher scores to use for classifier training. As an alternative approach to the filter method, I used a combination of Filter and wrapper methods in which the wrapper searched for the best 10 features among the 50 features that had the highest Fisher scores. The evaluation function for the wrapper was to select the features that lead to the highest average 5-fold cross validation G-Mean. Importantly, using both approaches, I made sure no false positive feature was being selected for a particular participant i.e., a feature that had a very high score for a subject while its score was low on average among participants. I did not consider this feature since it might have received a high score by chance, and I decided to not rely on these kinds of features that have accidentally received high scores for a participant. Mathematically, I took an average of the Fisher scores of each feature across participants and sorted the features based on their average Fisher scores. If a feature was ranked high for a participant while on the average scores ranking, that feature was not among the top 500, I would not consider that feature for classification.

Moreover, since there was an imbalance between the number of trials in some analyses (e.g., item hits vs item misses), I decided to handle the issue based on the feature selection strategy. Specifically, if I was using the Filter method for feature selection, I would use the bootstrapping approach in which I resampled the same number of hits and misses to train the classifier and I would repeat this process for 50 times and for labeling each test trial, I would pick the class which was selected for the majority of the times (i.e., more than 25 times) for that specific trial. On the other hand, if I was using the combination of filter and wrapper methods, I would not handle the imbalance issue by bootstrapping because of two reasons. Firstly, I had set the highest five-fold cross validation G-Mean as the evaluation criterion for selecting effective features and G-Means takes into account both specificity and sensitivity are high for the feature that is going to get selected. Moreover, the wrapper method is already a time-consuming algorithm since it performs many classification analyses and applying a bootstrapping approach to it makes it impractical.

Lastly, once the best features were selected, I compared the performance of 4 different classifiers: SVM, LASSO, Bayesian classifier, and Logistic Regression. For the classification problems that consisted of more than two classes, I applied both binary decision tree and the voting method to generalize the binary problem into a multiclass one.

In order to evaluate the classification performance, I used average five-fold cross validation G-Mean for the binary classification problems since the number of trials in different classes were imbalanced for the two binary problems in this study. Moreover, for multi-class classification, I used average five-fold cross validation accuracy since it would not be possible to define G-Mean for a multi-class problem, and the number of trials of different classes were relatively balanced in the multi-class problems in this study (see **4.3.2** for more detail). Furthermore, in order to investigate whether a classifier has performed above the chance level, I used permutation tests (Nichols & Holmes, 2002) by repeating the classification analysis to reach an empirical null distribution for the classifier performance. Specifically, I carried out the same five-fold cross validation classification procedure but used labels that were randomly shuffled at each repetition. This process was conducted 500 times per analysis with random label assignment on each repetition. This established an empirical null distribution of classification performance G-mean scores. Subsequently, I set the G-Mean, which was higher than 95% of the G-Mean values in the null distribution, as the threshold for determining the significance of a classifier's performance.

4.1 Investigating the feature types that were selected the most during all of the analyses

While it is highly recommended to extract many types of features and get as much information as possible from the signals (al-Qerem, Kharbat, Nashwan, Ashraf, & blaou, 2020), one should also consider the amount of time that it takes to extract many types of features. In this study, I investigated the ratio of the number of times that each feature type was selected as a good feature for training a classifier across all the analyses and participants. The results are shown in **Figure 11.**

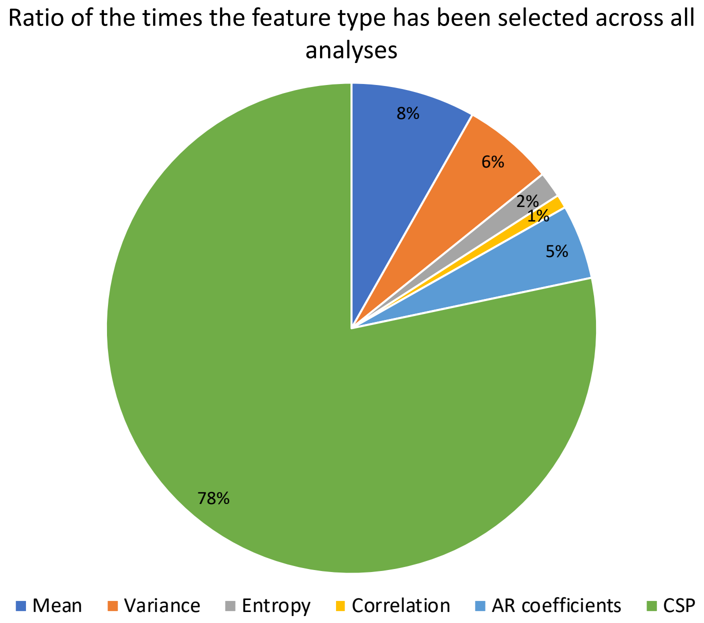

Figure 11- The percentage of the time each feature type was selected across all of the analyses

I found that features that were based on CSP were selected for 78% of the time and this suggests the superiority of this feature type for future analyses. A reason for their efficiency is that they are the output of an optimization problem since CSP filter is designed to maximize the difference between the variance of the trials of two classes.

4.2 Investigating the performance of feature selection methods across different classifiers

For each classification problem, once I extracted different types of features for different time segments and frequency bands, I used two feature selection methods in order to choose the effective features for training the classifier. I compared these two methods across different binary classification problems using different classifiers.

4.2.1 Binary Classification of Subsequent Item Memory Performance

In this problem, I was interested in classifying the trials at encoding based on the subsequent memory for the object i.e., item hits vs item misses. The summary of the results can be found in **Figures 12** and **13.**

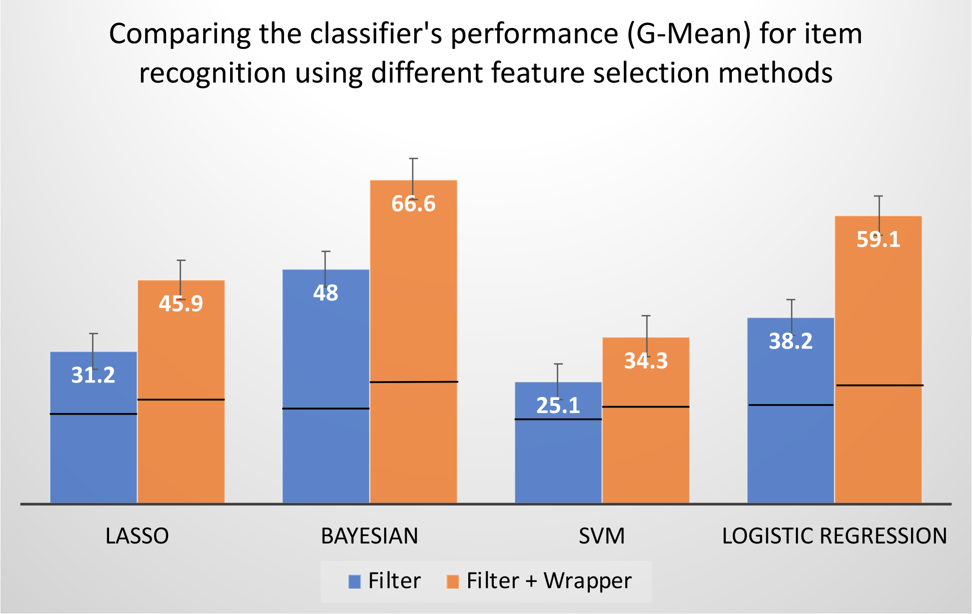

Figure 12- Comparing the filter method and the combination of filter and wrapper methods using different classifiers for item recognition across participants. The horizontal black lines indicate the 95% percentile of the empirical null distribution and the vertical lines indicate the 95% confidence interval of the perfromance across participants.

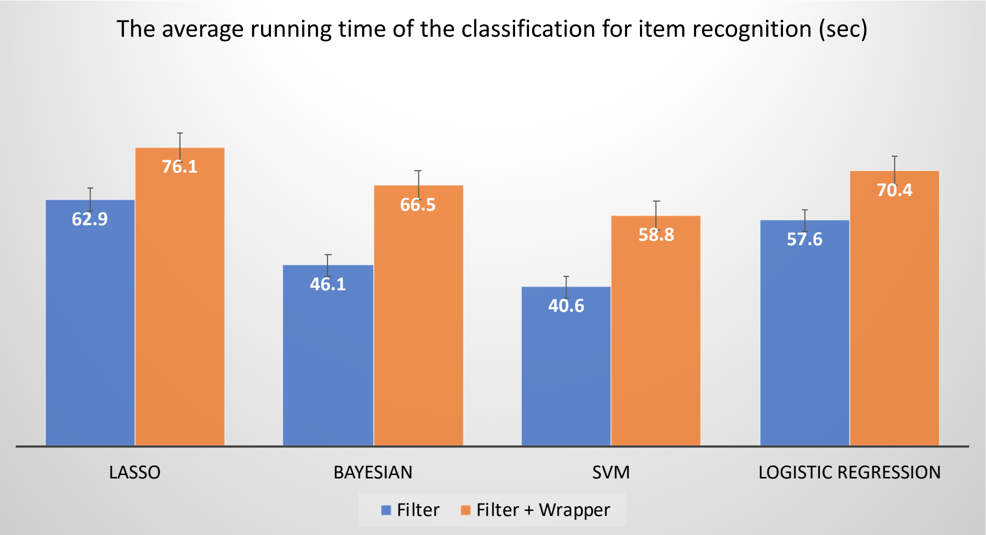

Figure 13- The average running time of different methods for classifying item recognition for each participant in seconds

In order to statistically compare the filter method and the combination of filter and wrapper methods, I ran two-way ANOVA to investigate if there is an effect of feature selection and choice of classifier for the performance and running time. Regarding both the performance and running time, the ANOVA results showed that effects of feature selection and choice of classifier were significant, while the interactions were non-significant [Performance: feature selection: $F(1,180) = 16.21$, $p < 0.001$, choice of classifier: $F(1,180) = 6.85$, $p = .010$, interaction: $F(1,180) = 0.31$, $p = .578$], [Running time: feature selection: $F(1,180) = 15.37$, $p < 0.001$, choice of classifier: $F(1,180) = 7.28$, $p = .007$, interaction: $F(1,180) = 0.38$, $p = .538$]. Since the effect of feature selection is significant for both performance and running time, as can be seen in **Figure 12**, The combination of filter and wrapper outperforms the filter method, while in terms of running time, as can be seen in **Figure 13**, the filter method is the faster approach.

4.2.2 Binary Classification of Subsequent Context Memory Performance (correct vs incorrect regardless of confidence)

In this problem, I was interested in classifying the trials at encoding based on the subsequent context memory for the trial. Specifically, I was interested in classifying context correct and incorrect trials (i.e., whether the participant has correctly identified the context as a match/mismatch). Since each item consisted of two contexts (color and scene), I performed the classification analyses on both contexts but since the results were very similar, I decided to show only the results associated with color context memory. It is also important to note that the trials were collapsed across the confidence levels in order to have enough (at least 20) context correct and context incorrect trials for every participant. The summary of the results can be found in **Figures 14** and **15**

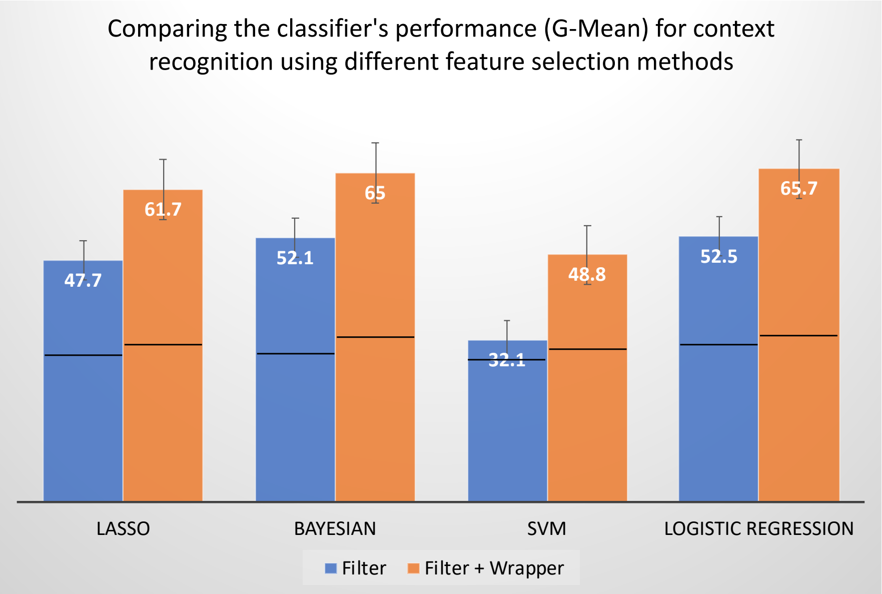

Figure 14- Comparing the filter method and the combination of filter and wrapper methods using different classifiers for context recognition across participants. The horizontal black lines indicate the 95% percentile of the empirical null distribution and the vertical lines indicate the 95% confidence interval of the perfromance across participants.

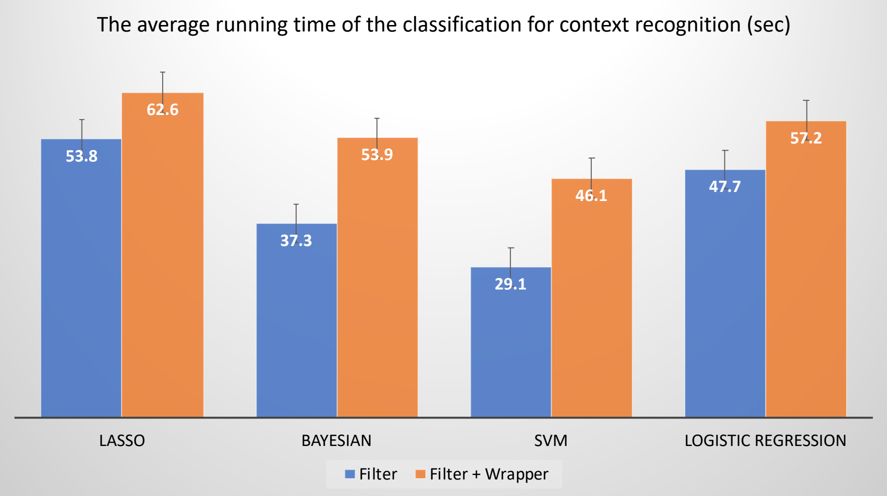

Figure 15- The average running time of different methods for classifying context recognition for each participant in seconds

Again, I statistically compared the filter method and the combination of filter and wrapper methods using two-way ANOVA. Regarding both the performance and running time, the ANOVA results indicated that the effects of feature selection and choice of classifier were significant, while the interactions were not significant [Performance: feature selection: $F(1,180) = 17.17$, $p < 0.001$, choice of classifier: $F(1,180) = 6.62$, $p = .011$, interaction: $F(1,180) = 0.26$, $p = .611$], [Running time: feature selection: $F(1,180) = 18.92$, $p < 0.001$, choice of classifier: $F(1,180) = 7.69$, $p = .006$, interaction: $F(1,180) = 0.52$, $p = .472$]. Since the effect of feature selection is significant for both performance and running time, as can be seen in **Figure 14**, The combination of filter and wrapper outperforms the filter method, while in terms of running time, as can be seen in **Figure 15**, the filter method is the faster approach.

4.3 Investigating the performance of each classifier in different analyses

The prior analyses made clear that the combination of the filter and wrapper methods outperforms the filter method alone. As a result, in this next section, I have compared different types of classifiers after they were trained by the effective features selected using the combination of filter and wrapper methods. Moreover, for multiclass classification, I have compared different types of classifiers, as well as different methods of generalization of binary problem into a multiclass one (since these two comparisons were not mutually exclusive).

4.3.1 Binary classification

4.3.1.1 Binary Classification of Subsequent Item Memory Performance

In section 4.2.1, I ran two-way ANOVA to examine if there is an effect of feature selection and choice of classifier for the performance and running time and the ANOVA showed that the effects of feature selection and the choice of classifier are significant in terms of both performance and running time. Since I found that the combination of wrapper and filter outperforms the filter method, I compared the classifiers only based on the feature selection using that approach. For the performance, since Bayesian had the best performance (as can be seen in **Figure 12**), I compared its performance with the other three classifiers' performances across participants and the t-tests indicated that Bayesian significantly outperformed the other three classifiers [Bayesian vs LASSO: $t(45) = 17.304$, $p < 0.001$; Bayesian vs SVM: $t(45) = 26.818$, $p < 0.001$; Bayesian vs Logistic Regression: $t(45) = 5.884$, $p < 0.001$]. However, for the running time, as can be seen in **Figure 13,** SVM was the fastest classifier. While it was significantly faster than Logistic Regression and LASSO, it was not significantly faster than Bayesian [SVM vs LASSO: $t(45) = 2.219$, $p = 0.016$; SVM vs Bayesian: $t(45) = 1.061$, $p = 0.15$; SVM vs Logistic Regression: $t(45) = 2.103$, $p = 0.021$].

4.3.1.2 Binary Classification of Subsequent Context Memory Performance (correct vs incorrect regardless of confidence)

Again, the conducted ANOVA in 4.2.2 showed that the effects of feature selection and the choice of classifier are significant in terms of both performance and running time. Similar to the previous section, since I found that the combination of wrapper and filter outperforms the filter method, I compared the classifiers only based on the feature selection using that approach. For the performance, since Logistic Regression had the best performance (as can be seen in **Figure 14**), I compared its performance with the other three classifiers' performances across participants and the t-tests indicated that Logistic Regression significantly outperformed SVM and LASSO, but not

Bayesian [Logistic Regression vs LASSO: $t(45) = 3.428$, $p < 0.001$; Logistic Regression vs SVM: $t(45) = 5.273$, $p < 0.001$; Logistic Regression vs Bayesian: $t(45) = 0.543$, $p = 0.29$]. However, for the running time, as can be seen in **Figure 15,** SVM was the fastest classifier. While it was significantly faster than Logistic Regression and LASSO, it was not significantly faster than Bayesian [SVM vs LASSO: $t(45) = 2.364$, $p = 0.011$; SVM vs Bayesian: $t(45) = 0.985$, $p = 0.16$; SVM vs Logistic Regression: $t(45) = 2.289$, $p = 0.013$].

4.3.2 Multi-class classification

As stated in the Methods chapter, I used two different approaches to generalize a binary classification problem into a multiclass problem, namely the voting method and the binary decision trees. Once I selected the effective features using the combination of filter and combination methods, I used both generalization approaches for all of the four classifiers in order to investigate which classifier and generalization method outperform the others. Moreover, I used only the accuracy as the evaluation metric of the performance since it is not possible to define sensitivity, specificity, and G-Mean in a multi-class classification problem. Instead, it is possible to make a confusion matrix (e.g., how often a trial belonged to class a, but the classifier has assigned it to class b?) and show the results in this matrix, as can be seen in **Figure 16.** However, the confusion matrices did not show obvious biases in assigning the trials to a particular class for these analyses and hence, I decided focus only on the accuracy to make the comparison more straightforward.

		Actual Values		
		Class 1	Class 2	Class 3
Predicted Values	Class 1	22%	7%	4%
	Class 2	7%	24%	6%
	Class 3	5%	4%	21%

Figure 16- The confusion matrix for a three-class classification problem. This matrix describes the ratio of the times a trial with a specific true label is assigned to a specific label by the classifier

4.3.2.1 Four-Class Classification of Subsequent Context Memory Performance

In this problem, I was interested in classifying all four different types of context memory states including correct with high confidence, correct with low confidence, incorrect with low confidence, and incorrect with high confidence. One important thing to keep in mind is that there were only 27 participants who had at least 20 trials for each of these four classes and I performed the classification only on these participants. The summary of the results can be found in **Figures 17 and 18**.

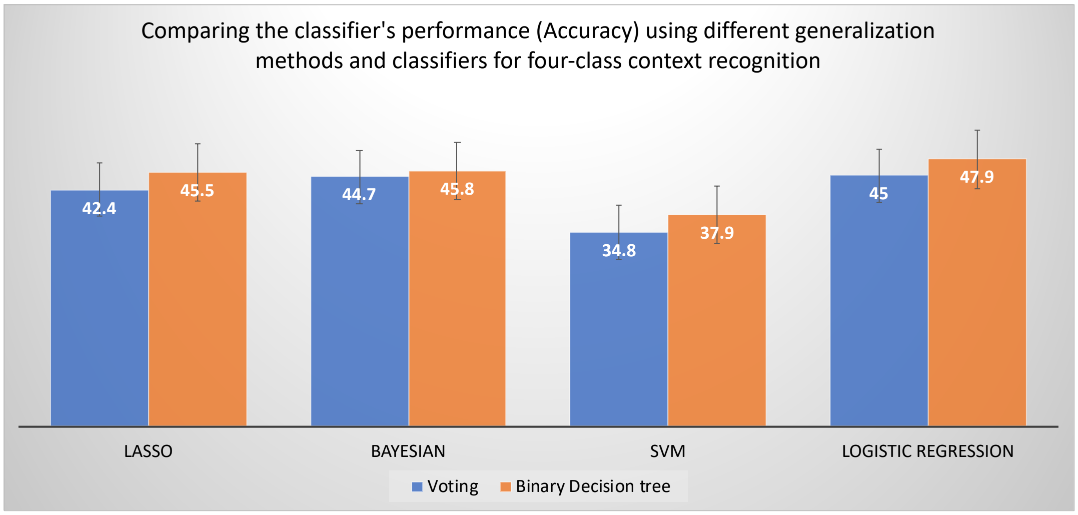

Figure 17- Comparing the performance of different classifiers across participants for four-class classification of context recognition. The horizontal black lines indicate the 95% percentile of the empirical null distribution and the vertical lines indicate the 95% confidence interval of the perfromance across participants.

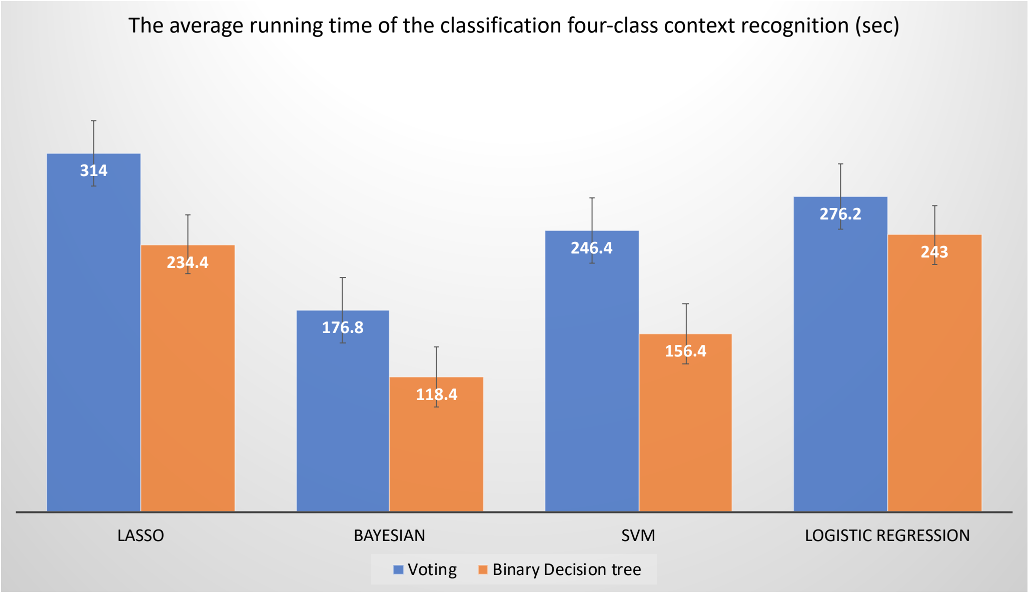

Figure 18- The average running time of different classifiers for four-class classification of context recognition for each participant in seconds

In order to statistically compare the voting method and the binary decision tree, I ran two-way ANOVA to investigate if there is an effect of generalization method and choice of classifier for the performance and running time. Regarding both the performance and running time, the ANOVA results showed that effects of feature selection and choice of classifier were significant, while the interactions were non-significant [Performance: generalization method: $F(1,104) = 18.21$, $p < 0.001$, choice of classifier: $F(1, 104) = 7.85$, $p = .006$, interaction: $F(1, 104) = 0.41$, $p = .523$], [Running time: generalization method: $F(1, 104) = 16.37$, $p < 0.001$, choice of classifier: $F(1, 104) = 7.33$, $p = .007$, interaction: $F(1, 104) = 0.39$, $p = .533$]. Since the effect of generalization method is significant for both performance and running time, as can be seen in **Figure 17** and **Figure 18**, The binary decision tree outperforms and is faster than the voting method. The fact that it takes less time to run is not surprising since in this case, the voting method performs 4 binary classifications while the binary decision tree performs 3 binary classifications. Subsequently, since I found that the binary decision tree outperforms the voting method, I compared the classifiers only based on the generalization using that approach. For the performance, since Logistic Regression had the best performance (as can be seen in **Figure 17**), I compared its performance with the other three classifiers' performances across participants and the t-tests indicated that Logistic Regression significantly outperformed SVM, but not LASSO and Bayesian classifier [Logistic Regression vs LASSO: $t(26) = 1.473$, $p = 0.074$; Logistic Regression vs SVM: $t(26) = 4.688$, $p < 0.001$; Logistic Regression vs Bayesian: $t(26) = 1.561$, $p = 0.063$]. However, for the running time, as can be seen in **Figure 18,** Bayesian was the fastest classifier. While it was significantly faster than Logistic Regression and LASSO, it was not significantly faster than SVM

[Bayesian vs LASSO: *t*(26) = 4.276, *p* < 0.001; Bayesian vs SVM: *t*(26) = 1.379, *p* = 0.087; Bayesian vs Logistic Regression: *t*(26) = 4.703, *p* < 0.001].

4.3.2.2 Three-Class Classification of Context Decoding

In this problem, I was interested in classifying the trials based on the context that the participant perceived during encoding. I used only the trials where the object was correctly identified as old and the context of interest was correctly identified as a match/mismatch compared to the one shown during encoding. The reason being that for trials the participant later forgot the object or context, he may not have been attending to the context and performance for this analysis might be contaminated by these "error" trials. Since there were three possible colors/scenes, this was a three-class classification problem, and the procedure was similar to the previous problem. Again, since the results for color and scene decoding were fairly similar, I have shown only the results for color decoding in this section. The results are shown in **Figures 19** and **20.**

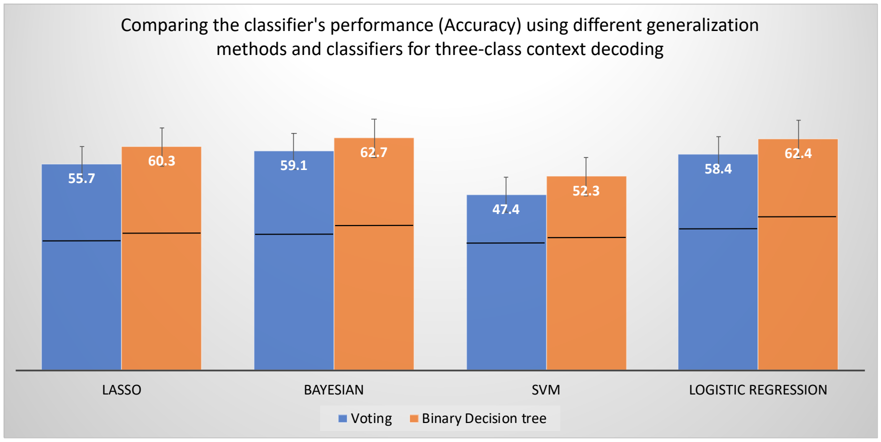

Figure 19- Comparing the performance of different classifiers across participants for three-class classification of context decoding. The horizontal black lines indicate the 95% percentile of the empirical null distribution and the vertical lines indicate the 95% confidence interval of the perfromance across participants.

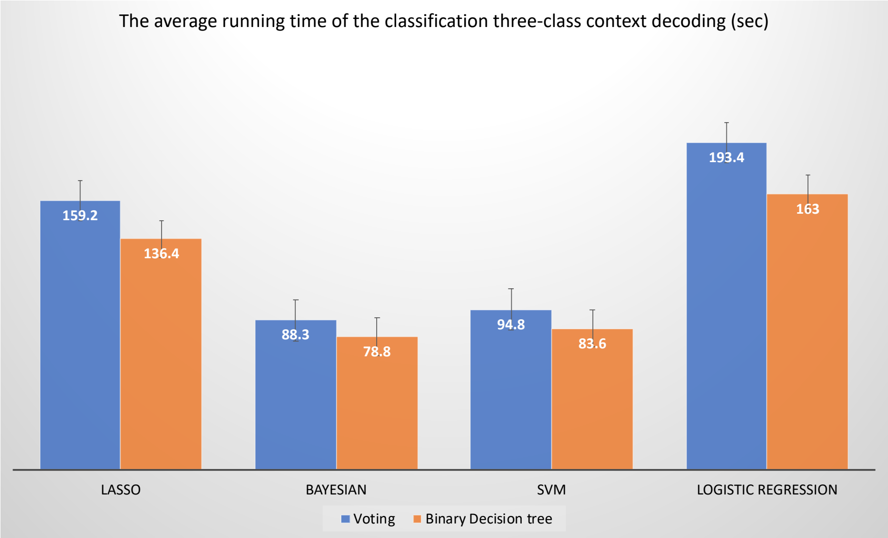

Figure 20- The average running time of different classifiers for three-class classification of context decoding for each participant in seconds

Again, in order to statistically compare the voting method and the binary decision tree, I ran two-way ANOVA to investigate if there is an effect of generalization method and choice of classifier for the performance and running time. Regarding both the performance and running time, the ANOVA results showed that effects of feature selection and choice of classifier were significant, while the interactions were non-significant [Performance: generalization method: $F(1,180) = 19.31$, $p < 0.001$, choice of classifier: $F(1,180) = 10.71$, $p = .001$, interaction: $F(1,180) = 0.66$, $p = .418$], [Running time: generalization method: $F(1,180) = 21.67$, $p < 0.001$, choice of classifier: $F(1,180) = 8.63$, $p = .003$, interaction: $F(1,180) = 0.29$, $p = .591$]. Since the effect of generalization method is significant for both performance and running time, as can be seen in **Figure 19** and **Figure 20**, The binary decision tree outperforms and is faster than the voting method. In the next step, since I found that the binary decision tree outperforms the voting method, I compared the classifiers only based on the generalization using that approach. For the performance, since Bayesian had the best performance (as can be seen in **Figure 19**), I compared its performance with the other three classifiers' performances across participants, and the t-tests showed that Bayesian classifier significantly outperformed SVM, but not LASSO and Logistic Regression [Bayesian vs LASSO: $t(45) = 1.407$, $p = 0.083$; Bayesian vs SVM: $t(45) = 4.792$, $p < 0.001$; Bayesian vs Logistic Regression: $t(45) = 0.542$, $p = 0.295$]. However, for the running time, as can be seen in **Figure 20,** Bayesian was the fastest classifier. While it was significantly faster than Logistic Regression and LASSO, it was not significantly faster than SVM [Bayesian vs LASSO: $t(45) = 4.677$, $p < 0.001$; Bayesian vs SVM: $t(45) = 1.340$, $p = 0.093$; Bayesian vs Logistic Regression: $t(45) = 4.703$, $p < 0.001$].

4.4 Investigating the impact of the number of features selected on classifier performance

While the combination of Filter and Wrapper methods led to the best results, a few questions remain unanswered. First, it is not clear how many features should be passed through the filter method so that the wrapper can search for the most effective ones. It is certainly useful to keep as many features as possible since there might be features that will not receive very high Fisher scores although they could be effective for classification. As a result, the wrapper might miss these types of features if it searches through only a small number of features. However, searching through large number of features can become very time-consuming and it is essential to determine a reasonable number of filtered features that the wrapper will search from. Here, I performed several classification analyses based on the number of features the wrapper searched through to select the best features for the classification of item hits vs misses. In order to be consistent, the wrapper always selected the best 10 features among the features through which it was searching. I performed these analyses for the four types of classifiers (SVM, Bayesian, LASSO, and Logistic Regression) for 10 participants. I limited this analysis to just 10 participants because of its time-consuming running time. The average performance and running time based on the number of filtered features for the Bayesian classifier are shown in **Figures 21** and **22.** The average patterns for the other classifiers were very similar (not shown).

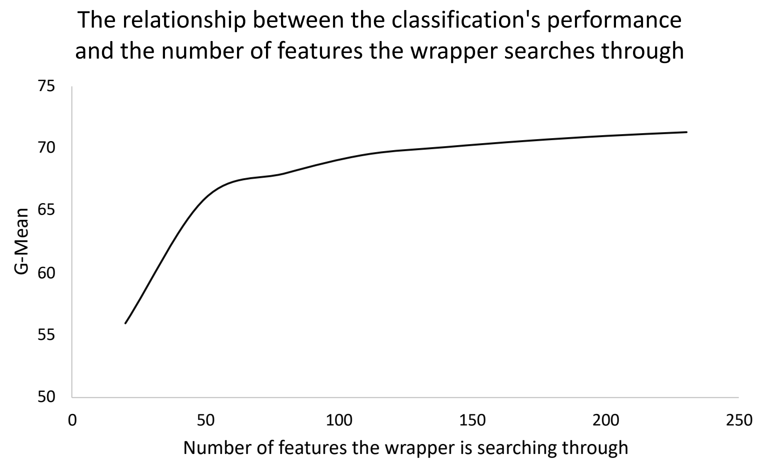

Figure 21- The relationship between the classification's performance and the number of features the wrapper searches through

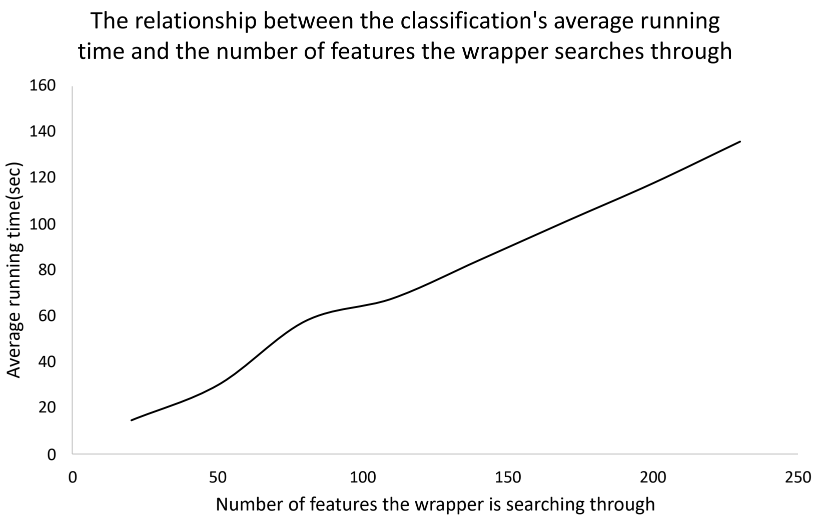

Figure 22- The relationship between the classification's average running time and the number of features the wrapper searches through

Based on these two plots, for problems of this study, it can be interpreted that filtering the top 50-100 features for the wrapper to search through them seems to be an efficient choice for this dataset since by filtering more than 100 features, time will be sacrificed considerably for only a slight increase in performance.

The second question that needs to be answered is how many features should be selected by the wrapper? Again, the answer depends on the problem and the researcher's preference, but I selected between 1 and 30 from the 100 filtered features for 10 participants. The average performance and running time for the Bayesian classifier are shown in **Figures 23** and **24**. The patterns for other classifiers were very similar.

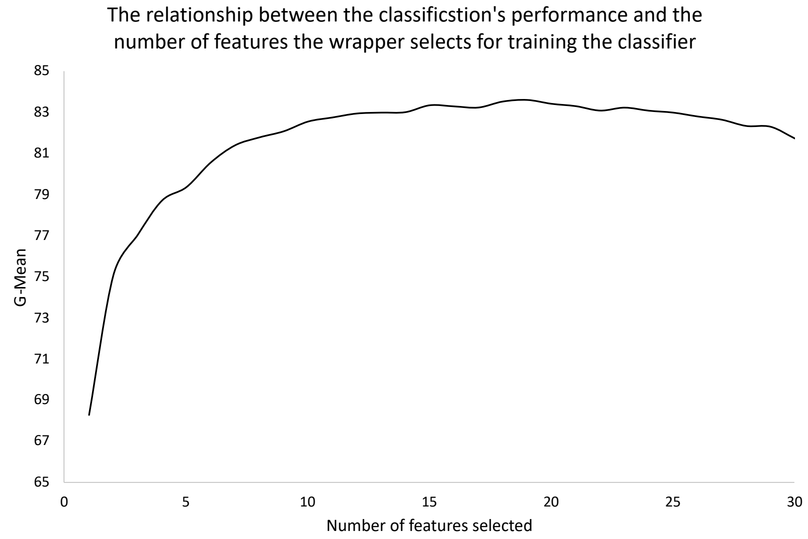

Figure 23- The relationship between the classification's performance and the number of features the wrapper selects

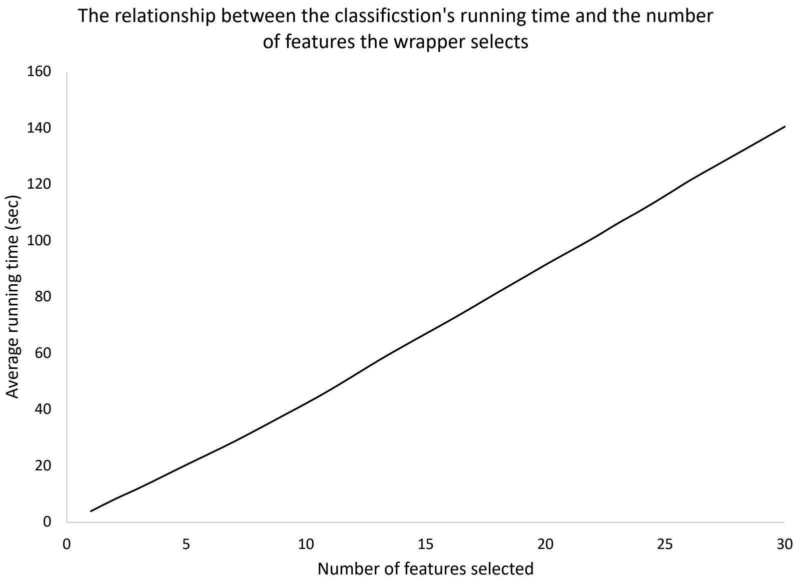

Figure 24- The relationship between the classification's average running time and the number of features the wrapper selects

Based on these two plots, it can be interpreted that selecting approximately 10 features is a reasonable decision since performance did not increase considerably past this point. However, as can be seen in **Figure 2**4, running time did continue to increase. Moreover, as can be seen, the performance starts to drop after 15 features which indicates the model is overfitting the data, and cross validation performance decreases. In other words, by increasing the number of features, the classification model becomes more complicated (see **Figure 2**) and it will train the classifier in a way to improve the performance on the training set (since it is provided with more information regarding the training data compared to before). However, the model becomes too specific to the training set and it won't generalize well to the test set which leads to a drop in performance on cross-validation.

4.5 Testing the generalizability of the methodology

The above analyses showed that optimal classification performance and running time was obtained by using a sequence of extracting CSP-based features, filtering a specific number of features and then passing them to wrapper for feature selection, and eventually training a Bayesian classifier. However, it is essential to verify that this approach works well for other classification problems and datasets in order to establish its generalizability. To this end, I performed two additional analyses. First, I trained a classifier using the data from one participant in this study and tested this classifier on another participant of this study. While the performance will likely drop, if it remains strong, the classification approach has generalizability (Arevalillo-Herráez, Cobos, Roger, & García-Pineda, 2019). Second, I performed the same analysis approach on another dataset to make sure the methodology can be used for other classification problems as well.

4.5.1 Training a classifier on one participant and testing it on another participant

In this analysis, for classifying item hits and misses, I extracted all types of features that I had already used in this study from the data of one of the participants in this study, evaluated each feature by Fisher's criterion, filtered the top 50 features for the wrapper to select the best 10 features by using Bayesian classifier. Once the classifier was trained, I extracted the selected features from the data of another participant in this study and used the previously trained classifier to classify the trials of that individual. I repeated this process for 10 pairs of participants. I found that the average G-Mean was 37% (range: [26-46]) while the chance level was 29% on average (range: [25-33]) and in 9 out of 10 analyses the classification performed above the chance level.

For example, as can be seen in **Figure 12**, performance dropped from 66.6% to 37%. However, the fact that most of these analyses performed above the chance level confirms the generalizability of the recommended methodology obtained in this study.

4.5.2 Testing the methodology on another dataset

In order to investigate whether the optimal methodology works well on another dataset, I performed the classification analysis on a free-recall associate memory task. Specifically, this task consisted of three sessions in which the first session consisted of a study block of English category-exemplar pairs (e.g., TREE and PINE), followed by four test blocks cued recall. Session 2 consisted of alternating study and test blocks for Swahili-English word pairs while Session 3 was a simple cued recall test on the items studied in Session 2, with the Swahili words presented as cues. (Rafidi, Hulbert, Pacheco, & Norman, 2018). EEG data were collected during the first two sessions. I was interested in classifying the items correctly recalled vs the items not correctly recalled. Again, I extracted all types of features that I had already used in this study (including CSP-based features, entropy, mean, variance, correlation, and features based on the AR model), evaluated each feature by Fisher's criterion, filtered the top 50 features for the wrapper to select the best 10 features by using Bayesian classifier. I performed this classification procedure for 10 participants. I found that the average G-Mean was 71% (range: [57-84]) while the chance level was 42% on average (range: [38-46]).

5. Discussion

In everyday life, it is very common that even a person with no clinically significant memory impairment shows episodic memory failures such as forgetting where he parked his car earlier in the morning. Using fMRI and EEG, cognitive neuroscientists have been examining the neural foundations of these kinds of memory failures, and successes, with various approaches. While it is common to use ERPs or average BOLD signals to discriminate neural activity associated with successful vs. unsuccessful memory performance, averaging approaches do not allow us to explore single events. Classification of brain states associated with single events using real-time signals recorded from the scalp offers the potential for the development of real-time interventions to support everyday learning. Although some studies have performed single trial classification of different memory states, performance has arguably been insufficient for the purpose of developing an effective intervention system. Moreover, these previous studies used different methods (i.e., classifiers, feature selection, etc.) without explaining the reasons behind their choices. As a result, in this study, I systematically compared different methods for the same dataset collected from an adult lifespan sample in order to examine which methods work the best in order to give some recommendations for future researchers who are interested in performing classification analyses on cognitive problems. I found that the CSP-based features can distinguish the trials of different classes better than other types of features, and the combination of the wrapper and filter methods outperforms using only the filter method to select the effective features. Moreover, the Bayesian classifier was the best classifier, especially when there was an imbalance between the number of trials of each class. Lastly, for multi-class classification, the best strategy to generalize the binary classification is to use binary decision trees.

5.1. Recommendations for Future Studies

Researchers who want to optimize their classification's performance in the future can apply different methods to their datasets and compare the results to choose the best method since each problem may have its particular optimal solution. However, the current results can provide a shortcut for researchers. Here, I have provided the answers to a list of likely questions that a researcher will need to answer before performing classification analyses. I hope that by following these recommendations, future researchers can perform optimal classification analyses. It is worth mentioning that while I make these recommendations for EEG/MEG datasets, the same principles may apply to other kinds of data, including fMRI.

5.1.1 What type(s) of features should I extract?

Regarding the features that one should extract, it is highly recommended to extract as many types of feature as possible. However, if running time is a concern, I suggest the use of CSP-based features (Blankertz et al., 2004; Blankertz, Tomioka, Lemm, Kawanabe, & Muller, 2008; Guger, Ramoser, & Pfurtscheller, 2000; Koles, Lazar, & Zhou, 1990; Noh et al., 2014; Y. Wang, Gao, & Gao, 2005). While CSP-based features have been used frequently in the literature, in this study, I also found that they outperform the other type of features that are commonly used in the literature including entropy, mean and variance of signal, correlation, and features obtained using AR model.

Specifically, while I extracted all of the aforementioned features for each analysis, the CSP-based features were selected for 78% of the time while the other five types of features were selected 22% of the time altogether, as can be seen in **Figure 11**. One reason for the superiority of the CSP-based features is the fact that they are extracted after solving an optimization problem. Specifically, the CSP filter is designed in such a way as to maximize the variance difference between the trials of two classes. Thus, it is not surprising that it outperforms the variance of the signals of the original electrodes (Guger et al., 2000). While the CSP-based features lead to high classification performance, one might wonder why there are studies that used other types of features instead of CSP-based features. One of the reasons for choosing other types of features is related to the purpose of the research. Specifically, if the primary concern for a classification problem is the performance, CSP-based features are the best option. However, one might be interested in features that are an inherent property of the brain, rather than mathematically derived ones, so the findings can be more directly related to this property. For example, (Höhne et al., 2016) used both phase and power information and compared their associated performances and found that classification using phase features outperformed the one using power features. Based on this finding, they could confirm the functional relevance of phase for long-term memory operations and recommended that phase information might be utilized for memory enhancement applications that use deep brain stimulation. Thus, the choice of features to extract is dependent upon the purpose of the researcher's classification problem.

5.1.2 How should I select the features that I want to use to train the classifier?

While it is crucial to get as much information as possible from each trial by extracting several types of features across time, frequency bands and electrodes, not all of the extracted features will necessarily be useful for classification. There are two commonly used techniques for finding the effective features for classification, namely filtering and wrapper methods. Filtering methods are fast since they involve a non-iterative computation on the dataset which execute faster than training a classifier. Moreover, since the filter methods evaluate the intrinsic characteristics of the features, rather than their interaction with a specific classifier, their results show more generality and the selected features will perform "well" for a larger family of classifiers. By contrast, wrapper methods generally achieve higher performance than filter methods, but the solution will lack generality since they are tuned to the particular interactions between the classifier and the dataset. Specifically, if a set of features are selected using the wrapper method while a Bayesian classifier was used in the process, the same set of features will not necessarily lead to high performance on another classifier, such as SVM. This is because the choice of classifier matters in selecting the effective features since different sets of features are evaluated by directly training that particular classifier. Moreover, the fact that the wrapper method evaluates several different sets of features by directly training the classifier using those features lead to its slower execution while the filter method is faster since it can quickly evaluate each individual feature using a statistical metric, instead of training any classifier (Mitchell, 1997). If the number of extracted features is high (e.g., more than 200), it will be impractical to pass all of the features to the wrapper and it would be wise to apply wrapper methods on a smaller set of features by first using filter methods to score individual features, and then picking those with the highest scores to pass to the wrapper (usually top 50, top 100, or top 200 depending on the researcher's preference for the running time and performance). With all of these issues in mind, for the classification problems for which accuracy is the most important factor, such as those related to brain-computer interfaces, wrapper methods such as sequential forward selection should be used to select the most effective features (Dias, Kamrunnahar, Mendes, Schiff, & Correia, 2010; Kirar & Agrawal, 2018; Zhang, Gan, & Wang, 2015). In the current analyses, on average, selecting the effective features using the combination of filter and wrapper methods resulted in 13.8% G-Mean improvement and took 17.1 seconds longer compared to the filter method but only for the training stage (see **Figure 10**). For the current problems of interest involving offline data analyses of memory-related brain states, this time cost was of no consequence.

5.1.3 Which classifier should I use?

First of all, if there is an imbalance between the number of trials in each condition, it's highly suggested that the researcher uses Bayesian classifier due to its ability to handle the imbalance issue better than other commonly used classifiers. One of the reasons that Bayesian classifiers can handle the imbalance issue better than the other commonly used classifiers is because it does not solve an optimization problem to reduce the cost/error function. Specifically, when a classifier is adjusting its parameter to reduce the cost, it might find it optimal to label all of the trials as the class with majority of the trials to minimize the associated error function. However, since Bayesian classifiers do not do this, they are not very sensitive to the imbalance (Ali, Shamsuddin, & Ralescu, 2015; Daskalaki, Kopanas, & Avouris, 2006). However, if there is no imbalance between conditions, although Bayesian classifiers don't outperform Logistic Regression and LASSO, it is still the recommended classifier due to its fast training (as can be seen in **Figure 19** and **Figure 20**). Specifically, the parameters of the Bayesian classifier model including a priori and conditional probabilities are learned using a deterministic set of steps. These steps involve only counting and dividing which are trivial operations. Moreover, as I mentioned above, the Bayesian classifier does not perform an optimization of a cost equation involved in training the model and it does not solve any matrix equations which are procedures that can be computationally costly (Fukunaga, 1990). Moreover, in this study, I found that Bayesian classifier outperformed the second choice of classifier (i.e., Logistic Regression) by 1.3% G-Mean improvement while it was trained 54 second faster than the second choice of classifier.

5.1.4 What if the number of trials is not the same across different classes?

In many classification problems, there is an imbalance between the number of trials for each class and this can lead to a bias for the classifier to label the trials as the class with majority of the trials. In order to handle this issue, one technique is to use under-sampling/bootstrapping i.e., resampling the same number of trials from each class and repeating this process for many times, but there are two disadvantages with this approach: it will never use all the available trials and it is slow to execute since it performs classification multiple times. The suggested approach is to change the evaluation metric when the sequential forward selection is selecting the effective set of features. Specifically, while accuracy is a good measurement of a classifier's performance when the classes are balanced, it can be misleading for the imbalanced situation. As a result, it is recommended to use all the trials for classification, but simply change the evaluation metric to the average k-fold cross-validation G-Mean. In this study, handling the imbalance issue by changing the evaluation metric of the wrapper method to the average k-fold cross-validation G-Mean resulted in 13.8% G-Mean improvement and took 17.1 seconds longer compared to the second approach i.e., under-sampling/bootstrapping while selecting the effective features using the filter method. Another solution would be to apply the under-sampling/bootstrapping approach while selecting the features using the wrapper method. However, since the wrapper method is already a time-consuming method, applying the under-sampling/bootstrapping technique on that case would be impractical and I did not perform that analysis.

5.1.5 How many features should I select?

An important parameter that one needs to determine is the number of effective features that one should select in order to properly train the classifier and it depends on the problem, how many trials are available, and tolerance to the running time. It is recommended to plot a figure like **Figure 21** to understand how much performance would change by selecting an additional feature. While the general pattern in the diagram will be the same as **Figure 21** (i.e., the performance will start to increase by increasing the number of features, then remain at a stable level, and then start to decrease due to overfitting), the cut-off point will change based on the problem and researcher's preference. Moreover, at some point, there will be a trade-off between performance and the running time so that increasing the performance just a little bit might not be worth it if the running time is important for the researcher. Thus, there is not a unique answer for this question and it's up to the researcher to decide the optimal number.

5.1.6 Do I have enough trials to obtain a reliable and non-inflated estimation of performance for my classifier?

While k-fold cross-validation is the most common approach to evaluate a classifier's performance, performance might be inflated if there are not enough trials to train the classifier with as it cannot sufficiently control overfitting (Vabalas, Gowen, Poliakoff, & Casson, 2019). As a result, it is suggested to use nested cross-validation to predict the classifier's performance on future trials as it is a robust performance estimator regardless of the number of trials. In nested cross-validation, a part of the data will be completely left out and the remaining part will be used for k-fold cross validation. This process is repeated so that each trial will be left out for once. The average performance on the left-out sets will be used as the performance estimate. In this study, because some participants had very good memory performance, there were relatively few trials in the incorrect/forgotten memory conditions making it impossible to use nested cross-validation (i.e., reserving some trials for testing) for every participant. However, I performed nested cross-validation analyses on 10 of the participants who had enough incorrect memory conditions that I could perform classification for item memory, and the performance did drop a few percent compared to 5-fold cross validation (as can be seen in **Figure 25**). Importantly, however, the pattern of performance across different methods (i.e., classifiers, features extracted) was the same between 5-fold cross validation and nested cross-validation.

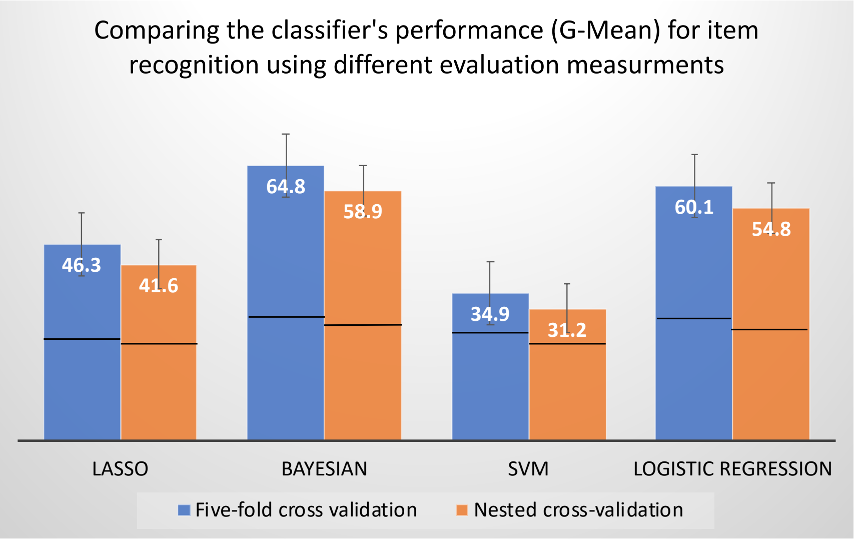

Figure 25- Comparing the classifier's performance according to the way it estimates the performance for the unseen data. In this analysis, the classifier separated item memory conditions and selected the effective features using the combination of filter and wrapper methods. As can be seen, the five-fold cross validation is slightly inflated due to the insufficient number of trials. However, the pattern of performance is preserved, and Bayesian classifier is the superior classifier according to nested cross-validation as well.

Although data collection involving human participants is expensive and recording neural activity for a long period can be exhausting for participants, it is highly recommended that the researcher collects as much data as possible. In order to determine if there are enough trials for the data the researcher has, a data-driven and practical approach is to plot a learning curve, like the one in **Figure 26** which is obtained by performing item memory classification for one participant after breaking each trial artificially into 10 trials (the new trials represented 200 ms of EEG recording instead of 2 seconds) in order to have enough trials to show the learning curve.

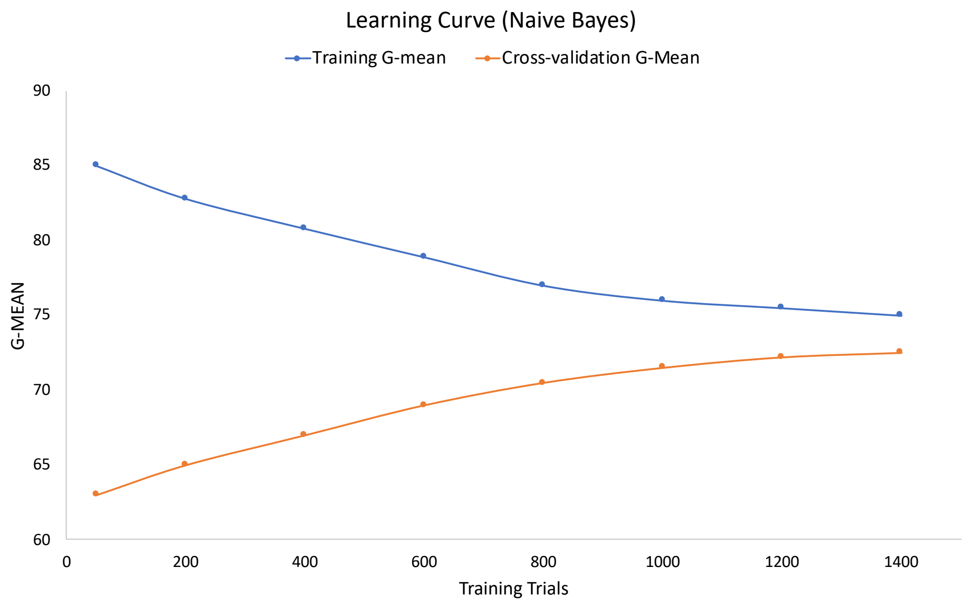

Figure 26- Learning curve to determine if there is enough training data for classification

The learning curve shows the evolution of the training and test errors as the size of training set increases. The training error increases by increasing the number of trials since it becomes more difficult to fit a model that accounts for the increasing variability/complexity of the training set. On the other hand, the test error decreases by having more trials since the model can generalize better using a higher amount of information. As can be sees on the right side of the plot, the two lines in the plot tend to get closet and asymptote. Consequently, there will be eventually a point in which having more trials will not impact the trained model. It is also worth mentioning that the difference between the test error and training error asymptotes is an indication of the model's overfitting. But more importantly, this plot is implementing whether it is necessary to have more data. Specifically, if the researcher plots the learning curve and the training and test lines do not seem to be reaching an asymptote, it is essential to obtain more data. As can be seen in **Figure 26**, in this study, having around 200 trials is not enough to have a non-inflated estimate of the classification performance since the training and test lines do not reach an asymptote at that point. However, I artificially increased the trials to show that the lines will reach an asymptote at some point and this indicates that it was necessary to collect more data in this particular study.

5.1.7 What if I am interested in classifying more than two cognitive conditions?

In order to classify more than 2 classes, a researcher should break the problem into multiple binary classification problems. While there are several approaches to do so, the efficient way, in terms of both running time and performance, is to use binary decision trees to generalize the binary problem into a multi-class one. While binary decision trees have been used frequently in the literature to generalize to a multiclass problem (Fei & Liu, 2006; Freeman, Kuli, & Basir, 2013; Mao, Zhou, Pi, Sun, & Wong, 2005), in this study, I also found that they outperform the voting method which is also commonly used in the literature. Specifically, on average, generalizing a binary classification problem into a multiclass one by binary decision tree resulted in 3.4% improvement in accuracy and it took 41.9 second less for the classification training compared to generalizing using voting method as can be seen in **Figures 17-20.** One of the reasons that it performs well is its simplicity and the fact that it does not have any hyperparameter. Moreover, since it performs less classification analyses compared to the voting method, when the results of the binary problems are combined to produce the final label, binary decision tree will have a better performance as the errors of the binary problems will be accumulated to some degree to generate the final labels (Fei & Liu, 2006).

5.2. Conclusion

All in all, in this study, using recorded EEG during episodic memory tasks, I systematically compared different methods of feature extraction, feature selection, and choice of classifier in the same study to examine which methods work the best for various classification problems. I found that the CSP-based features could discriminate the classes better than other types of features, and the combination of filtering and sequential forward selection was the optimal method to select the effective features. Furthermore, Bayesian classification outperformed other common options. Moreover, I tested these methods on another dataset, and they outperformed alternative methods, supporting their generalizability.

References

A.K. Jain, R.P.W. Duin, & J. Mao. (2000). Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 4–37.

Abdulrauf Sharifai, G., & Zainol, Z. (2020). Feature Selection for High-Dimensional and Imbalanced Biomedical Data Based on Robust Correlation Based Redundancy and Binary Grasshopper Optimization Algorithm. *Genes*, *11*(7). https://doi.org/10.3390/genes11070717

al-Qerem, A., Kharbat, F., Nashwan, S., Ashraf, S., & blaou, khairi. (2020). General model for best feature extraction of EEG using discrete wavelet transform wavelet family and differential evolution. *International Journal of Distributed Sensor Networks*, *16*, 155014772091100. https://doi.org/10.1177/1550147720911009

Ali, A., Shamsuddin, S. M., & Ralescu, A. (2015). *Classification with class imbalance problem : a review*. (August 2016).

Arevalillo-Herráez, M., Cobos, M., Roger, S., & García-Pineda, M. (2019). Combining Inter-Subject Modeling with a Subject-Based Data Transformation to Improve Affect Recognition from EEG Signals. *Sensors (Basel, Switzerland)*, *19*(13). https://doi.org/10.3390/s19132999

Bell, A. J., & Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*. https://doi.org/10.1162/neco.1995.7.6.1129

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition CLARENDON PRESS • OXFORD 1995*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.679.1104&rep=rep1&type=pdf

Blankertz, B., Muller, K.-., Curio, G., Vaughan, T. M., Schalk, G., Wolpaw, J. R., … Birbaumer, N. (2004). The BCI competition 2003: progress and perspectives in detection and discrimination of EEG single trials. *IEEE Transactions on Biomedical Engineering*, *51*(6), 1044–1051. https://doi.org/10.1109/TBME.2004.826692

Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Muller, K. (2008). Optimizing Spatial filters for Robust EEG Single-Trial Analysis. *IEEE Signal Processing Magazine*, *25*(1), 41–56. https://doi.org/10.1109/MSP.2008.4408441

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*. https://doi.org/10.1023/A:1009715923555

Burnham, K., & Anderson, D. R. (2003). Model selection and multimodel inference : a practical information-theoretic approach. *Journal of Wildlife Management*, *67*, 655.

Cohen, M. X. (2014). Analyzing Neural Time Series Data: Theory and Practice. *MIT Press*. https://doi.org/10.1017/CBO9781107415324.004

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, *1*(1), 131–156. https://doi.org/https://doi.org/10.1016/S1088-467X(97)00008-5

Daskalaki, S., Kopanas, I., & Avouris, N. (2006). Evaluation of Classifiers for an Uneven Class Distribution Problem. *Applied Artificial Intelligence*, *20*, 381–417. https://doi.org/10.1080/08839510500313653

Delorme, A., & Makeig, S. (2004). EEGLAB: An open source toolbox for analysis of single-trial

EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, *134*(1), 9–21. https://doi.org/10.1016/j.jneumeth.2003.10.009

Delorme, A., Sejnowski, T., & Makeig, S. (2007). *Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis*. *34*, 1443–1449. https://doi.org/10.1016/j.neuroimage.2006.11.004

Dias, N. S., Jacinto, L. R., Mendes, P. M., & Correia, J. H. (2009). Feature down-selection in brain-computer interfaces dimensionality reduction and discrimination power. *2009 4th International IEEE/EMBS Conference on Neural Engineering, NER '09*, 323–326. https://doi.org/10.1109/NER.2009.5109298

Dias, N. S., Kamrunnahar, M., Mendes, P. M., Schiff, S. J., & Correia, J. H. (2010). Feature selection on movement imagery discrimination and attention detection. *Medical & Biological Engineering & Computing*, *48*(4), 331–341. https://doi.org/10.1007/s11517-010-0578-1

Duarte, A., Ranganath, C., Winward, L., Hayward, D., & Knight, R. T. (2004). Dissociable neural correlates for familiarity and recollection during the encoding and retrieval of pictures. *Cognitive Brain Research*, *18*(3), 255–272. https://doi.org/10.1016/j.cogbrainres.2003.10.010

Duzel, E., Yonelinas, A. P., Mangun, G. R., Heinze, H.-J., & Tulving, E. (1997). Event-related brain potential correlates of two states of conscious awareness in memory. *Proceedings of the National Academy of Sciences*, *94*(11), 5973–5978.

https://doi.org/10.1073/pnas.94.11.5973

Efron, B., & Tibshirani, R. J. (1993). An introduction to the bootstrap. Vol. 57. In *Refrigeration And Air Conditioning*. https://doi.org/10.1111/1467-9639.00050

Ezzyat, Y., Wanda, P. A., Levy, D. F., Kadel, A., Aka, A., Pedisich, I., … Kahana, M. J. (2018). Closed-loop stimulation of temporal cortex rescues functional networks and improves memory. *Nature Communications*, *9*(1). https://doi.org/10.1038/s41467-017-02753-0

Fei, B., & Liu, J. (2006). Binary tree of SVM: a new fast multiclass training and classification algorithm. *IEEE Transactions on Neural Networks*, *17*(3), 696–704. https://doi.org/10.1109/TNN.2006.872343

Freeman, C., Kuli, D., & Basir, O. (2013). Feature-selected tree-based classification. *IEEE Transactions on Cybernetics*, *43*(6), 1990–2004. https://doi.org/10.1109/TSMCB.2012.2237394

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition (2nd Ed.)*. USA: Academic Press Professional, Inc.

Gu, Q., Li, Z., & Han, J. (2011). Generalized Fisher Score for Feature Selection. *UAI*.

Guger, C., Ramoser, H., & Pfurtscheller, G. (2000). Real-time EEG analysis with subject-specific spatial patterns for a brain-computer interface (BCI). *IEEE Transactions on Rehabilitation Engineering : A Publication of the IEEE Engineering in Medicine and Biology Society*, *8*(4), 447–456. https://doi.org/10.1109/86.895947

Hoffmann, S., & Falkenstein, M. (2008). The correction of eye blink artefacts in the EEG: A comparison of two prominent methods. *PLoS ONE*, *3*(8). https://doi.org/10.1371/journal.pone.0003004

Höhne, M., Jahanbekam, A., Bauckhage, C., Axmacher, N., & Fell, J. (2016). Prediction of successful memory encoding based on single-trial rhinal and hippocampal phase information. *NeuroImage*, *139*, 127–135. https://doi.org/10.1016/j.neuroimage.2016.06.021

Hua, J., Xiong, Z., Lowey, J., Suh, E., & Dougherty, E. R. (2004). Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, *21*(8), 1509–1515. https://doi.org/10.1093/bioinformatics/bti171

James, T., Strunk, J., Arndt, J., & Duarte, A. (2016). Neuropsychologia Age-related de fi cits in selective attention during encoding increase demands on episodic reconstruction during context retrieval : An ERP study. *Neuropsychologia*, *86*, 66–79. https://doi.org/10.1016/j.neuropsychologia.2016.04.009

Johnson Jr., R. (1995). Event-related potential insights into the neurobiology of memory systems. *Handbook of Neuropsychology*, *10*(January 1995), 135–163.

Kaper, M., Meinicke, P., Grossekathoefer, U., Lingner, T., & Ritter, H. (2004). BCI Competition 2003--Data set IIb: support vector machines for the P300 speller paradigm. *IEEE Transactions on Bio-Medical Engineering*, *51*(6), 1073–1076. https://doi.org/10.1109/TBME.2004.826698

Kirar, J. S., & Agrawal, R. K. (2018). Relevant Frequency Band Selection using Sequential Forward Feature Selection for Motor Imagery Brain Computer Interfaces. *2018 IEEE*

Symposium Series on Computational Intelligence (SSCI), 52–59. https://doi.org/10.1109/SSCI.2018.8628719

Koles, Z. J., Lazar, M. S., & Zhou, S. Z. (1990). Spatial patterns underlying population differences in the background EEG. *Brain Topography*, *2*(4), 275–284. https://doi.org/10.1007/BF01129656

Koprinska, I. (2009). Feature Selection for Brain-Computer Interfaces. *Proceedings of the 13th Pacific-Asia International Conference on Knowledge Discovery and Data Mining: New Frontiers in Applied Data Mining*, 106–117. Berlin, Heidelberg: Springer-Verlag.

Kray, J., & Lindenberger, U. (2000). *Psychology and Aging Adult Age Differences in Task Switching*. *15*(1), 126–147. https://doi.org/10.1037//0882-7974.15.1.126

Lopez-Calderon, J., & Luck, S. J. (2014). ERPLAB: an open-source toolbox for the analysis of event-related potentials. *Frontiers in Human Neuroscience*, *8*(April), 1–14. https://doi.org/10.3389/fnhum.2014.00213

Lotte, F., Bougrain, L., Cichocki, A., Cierc, M., Congedo, M., Rakotomamonjy, A., & Yger, F. (2017). A review of classification algorithms for EEG-based BCI: A 10 year update. *Journal of Neural Engineering*, *4*, R1–R13. https://doi.org/10.1088/1741-2560/4/2/R01

Mao, Y., Zhou, X., Pi, D., Sun, Y., & Wong, S. T. C. (2005). Multiclass cancer classification by using fuzzy support vector machine and binary decision tree with gene selection. *Journal of Biomedicine & Biotechnology*, *2005*(2), 160–171. https://doi.org/10.1155/JBB.2005.160

Mirjalili, S., Sardouie, S. H., & Samiee, N. (2019). A Novel Algorithm Based on Decision Trees in Multiclass Classification. *2018 25th Iranian Conference on Biomedical Engineering and 2018 3rd International Iranian Conference on Biomedical Engineering, ICBME 2018*, 1–6. https://doi.org/10.1109/ICBME.2018.8703580

Mitchell, T. M. (1997). *Machine Learning*. Retrieved from https://books.google.com/books?id=EoYBngEACAAJ

Mosley, L. (2013). *A balanced approach to the multi-class imbalance problem*.

Ng, A., & Jordan, M. (2002). On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. *Adv. Neural Inf. Process. Sys*, *2*.

Noh, E., Herzmann, G., Curran, T., & de Sa, V. R. (2014). Using single-trial EEG to predict and analyze subsequent memory. *NeuroImage*, *84*, 712–723. https://doi.org/10.1016/j.neuroimage.2013.09.028

Nuwer, M. R., Comi, G., Emerson, R., Fuglsang-Frederiksen, A., Guerit, M., Hinrichs, H., … Rappelsburger, P. (1998). IFCN standards for digital recording of clinical EEG. *Electroencephalography and Clinical Neurophysiology*, *106*(3), 259–261. https://doi.org/10.1016/S0013-4694(97)00106-5

Oostenveld, R., Fries, P., Maris, E., & Schoffelen, J. M. (2011). FieldTrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Computational Intelligence and Neuroscience*, *2011*. https://doi.org/10.1155/2011/156869

Otten, L. J., Quayle, A. H., Akram, S., Ditewig, T. A., & Rugg, M. D. (2006). Brain activity before an event predicts later recollection. *Nature Neuroscience*, *9*(4), 489–491.

https://doi.org/10.1038/nn1663

Paller, K. A., Kutas, M., & Mayes, A. R. (1987). Neural correlates of encoding in an incidental learning paradigm. *Electroencephalography and Clinical Neurophysiology*, *67*(4), 360–371. https://doi.org/10.1016/0013-4694(87)90124-6

Paller, K. A., & Wood, C. (1988). Erps Predictive of Subsequent. *Biological Psychology*, *26*(1–3), 269–276. https://doi.org/10.1063/1.3583461

Paranjape, R. B., Mahovsky, J., Benedicenti, L., & Koles', Z. (2001). The electroencephalogram as a biometric. *Canadian Conference on Electrical and Computer Engineering 2001. Conference Proceedings (Cat. No.01TH8555)*, *2*, 1363–1366 vol.2. https://doi.org/10.1109/CCECE.2001.933649

Percival, D. B., Walden, A. T., & others. (1993). *Spectral analysis for physical applications*. cambridge university press.

Pfurtscheller, G., Neuper, C., Flotzinger, D., & Pregenzer, M. (1997). EEG-based discrimination between imagination of right and left hand movement. *Electroencephalography and Clinical Neurophysiology*, *103*(6), 642–651. https://doi.org/https://doi.org/10.1016/S0013-4694(97)00080-1

Phan, A. H., & Cichocki, A. (2010). Tensor decompositions for feature extraction and classification of high dimensional datasets. *Nonlinear Theory and Its Applications, IEICE*. https://doi.org/10.1587/nolta.1.37

Powell, P. S., Strunk, J., James, T., Polyn, S. M., & Duarte, A. (2018). Decoding selective attention to context memory: An aging study. *NeuroImage*, *181*(June), 95–107. https://doi.org/10.1016/j.neuroimage.2018.06.085

Rafidi, N. S., Hulbert, J. C., Pacheco, P., & Norman, K. A. (2018). Reductions in Retrieval Competition Predict the Benefit of Repeated Testing. *BioRxiv*. https://doi.org/10.1101/292557

Ramoser, H., Muller-Gerking, J., & Pfurtscheller, G. (2000). Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering*, *8*(4), 441–446. https://doi.org/10.1109/86.895946

Salari, N., & Rose, M. (2016). Dissociation of the functional relevance of different pre-stimulus oscillatory activity for memory formation. *NeuroImage*, *125*, 1013–1021. https://doi.org/10.1016/j.neuroimage.2015.10.037

Sardouie, S. H., & Shamsollahi, M. B. (2012). Selection of efficient features for discrimination of hand movements from MEG using a BCI competition IV data set. *Frontiers in Neuroscience*, *6*(APR), 1–7. https://doi.org/10.3389/fnins.2012.00042

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Song, Y., & Liò, P. (2010). A new approach for epileptic seizure detection: sample entropy based feature extraction and extreme learning machine. *Journal of Biomedical Science and Engineering*, *03*(06), 556–567. https://doi.org/10.4236/jbise.2010.36078

Strunk, J., James, T., Arndt, J., & Duarte, A. (2017). ScienceDirect Age-related changes in neural oscillations supporting context memory retrieval. *CORTEX*, *91*, 40–55. https://doi.org/10.1016/j.cortex.2017.01.020

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267–288. Retrieved from http://www.jstor.org/stable/2346178

Vabalas, A., Gowen, E., Poliakoff, E., & Casson, A. J. (2019). Machine learning algorithm validation with a limited sample size. *PloS One*, *14*(11), e0224365. https://doi.org/10.1371/journal.pone.0224365

Wang, T., Deng, J., & He, B. (2004). Classifying EEG-based motor imagery tasks by means of time-frequency synthesized spatial patterns. *Clinical Neurophysiology*, *115*(12), 2744–2753. https://doi.org/10.1016/j.clinph.2004.06.022

Wang, Y., Gao, S., & Gao, X. (2005). Common Spatial Pattern Method for Channel Selelction in Motor Imagery Based Brain-computer Interface. *Conference Proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, *2005*, 5392–5395. https://doi.org/10.1109/IEMBS.2005.1615701

Wei, Q., Wang, Y., Gao, X., & Gao, S. (2007). Amplitude and phase coupling measures for feature extraction in an EEG-based brain-computer interface. *Journal of Neural Engineering*, *4*(2), 120–129. https://doi.org/10.1088/1741-2560/4/2/012

Woodruff, C. C., Hayama, H. R., & Rugg, M. D. (2006). Electrophysiological dissociation of the neural correlates of recollection and familiarity. *Brain Research*, *1100*(1), 125–135. https://doi.org/10.1016/j.brainres.2006.05.019

Zhang, L., Gan, J. Q., & Wang, H. (2015). Localization of neural efficiency of the mathematically gifted brain through a feature subset selection method. *Cognitive Neurodynamics*, *9*(5), 495–508. https://doi.org/10.1007/s11571-015-9345-1