Brain-Computer Interface Movement Decoding

Phil Chen
NetID: mc702
ECE 580 Spring 2021

table of contents

introduction

What is Brain-Computer Interface?

Brain-computer interfacing is a technology that connects our natural brain with man-made devices, providing a new output channel for brain signals to communicate or control external devices

A BCI recognizes the intent of the user through electrophysiological or other signals from the brain, decodes the ongoing neural activity, and translates it into output commands that accomplish the user's goal.

BCI can potentially assist in restoring lost or impaired functions of people with severe disability by various devastating neuromuscular disorders or spinal cord damage, and in enhancing or supplementing functions in healthy individuals.

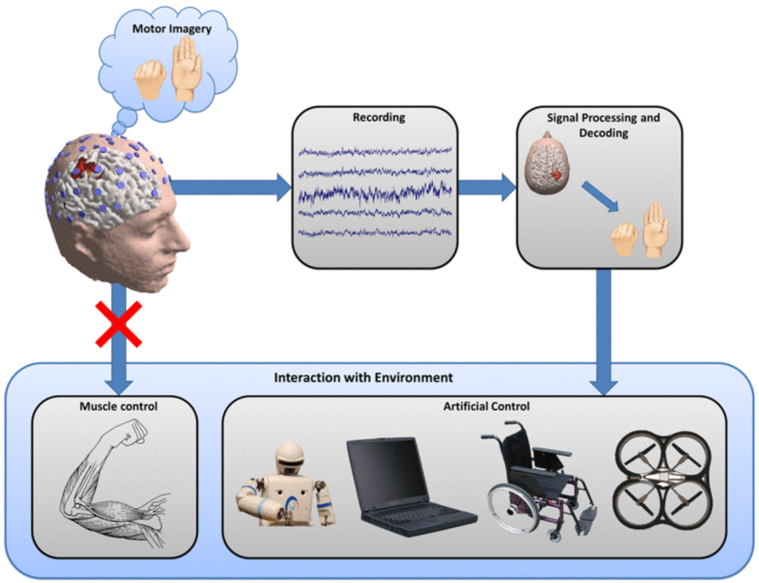

Fig. 1. General concept diagram behind a motor imagery-based brain-computer interface (BCI). EEG signals data is collected through user's imagery movement, filtered, and decoded to determine the user's intent.

Project objectives…

Provided with two sets of EEG signals data, including one from motor imagery and one from overt motion, we wish to construct an SVM classifier that distinguishes left movement (class 1) from right (class 0) with high accuracy.

In this project setup, we trained models for imagery movement data and overt movement data respectively. Each dataset contains 120 trials, and for each trial is composed of 204 features (channels). We not only hope to obtain a accurate classifier, but also wish to spot the dominating channels on the scalp that determine left/right movement.

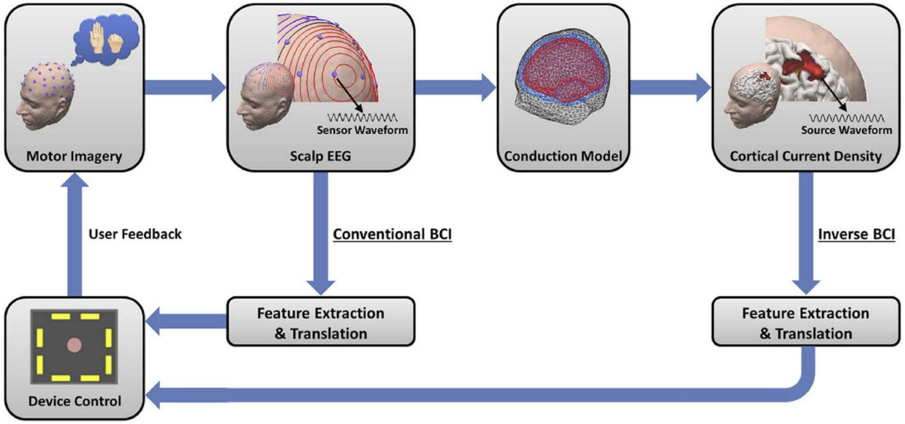

Fig. 2. Concept diagram of EEG imagery-based BCI, in conjunction with the head conduction model to produce EEG source signals that are used in classification.

introduction

Key findings...

We have several key findings with regards to this experiments setup.

1) SVM can perform well for both imagery movement and overt movement dataset with accuracy higher than 85%

2) The dominating channels of classifying such human movement show strong locality under both cases.

3) The selection of very small regularization parameter (C = 8.7e-7) implies two relatively distinguishable classes.

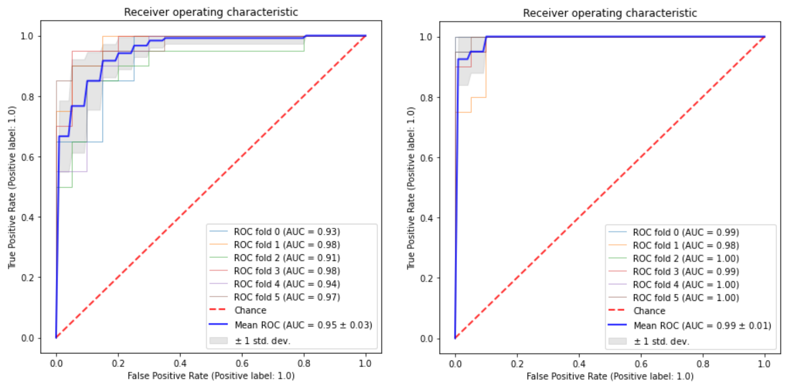

Fig. 3. Left image is the ROC curve for imagery movement data; right is that for overt movement data.

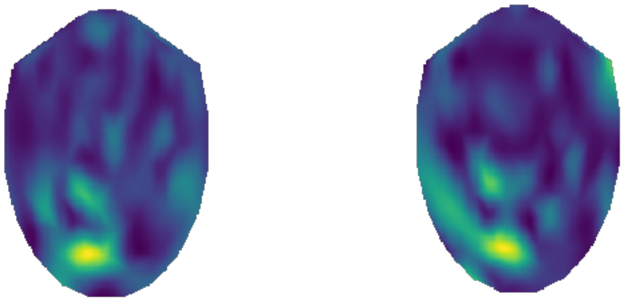

Fig. 4. Left image is the channel weights display of imagery movement; On the right is that of overt movement.

section 01

Mathematical formulation

How does support vector machine work?

By its nature, Support vector machine (SVM) solves a convex optimization problem to determine a set of support vectors that are responsible for classifications.

$$L(w,b,a,\mu) =$$

$$\frac{1}{2}\,\|w\|^2 - \sum_{n=1}^{N} a_n\{t_n(w^T x_n + b) - 1 + \varepsilon_n\} - \sum_{n=1}^{N} \mu_n \varepsilon_n + C\sum_{n=1}^{N} \varepsilon_n$$

Is the Lagrangian equation to maximize the margin of linearly inseparable data. The last two terms are for penalty of misclassification and regularization, respectively. The convex optimization problem becomes the one on the right:

$$\begin{aligned} \min_{W,C,\xi} \quad & \sum \xi_i + \lambda \cdot W^T W \\ \text{S.T.} \quad & y_i \cdot \left(W^T X_i + C\right) \geq 1 - \xi_i \\ & \xi_i \geq 0 \\ & \left(i = 1,2,\cdots,N\right) \end{aligned}$$

When $C = 0$, the constraint imposes no penalty for misclassification; Whereas when $C = \infty$, penalty for misclassification dominates and forces a boundary with few even none classification errors (highly biased).

We will later begin to explore the effect of regularization coefficient C through cross validation.

Mathematical formulation

How does SVM classify EEG signals?

Based on the Lagrangian equation provided in the previous slide, the optimization process is to solve quadratic programming problems under the constraint:

$t_n y(x_n) \geq 1 - \varepsilon_n$, where t_n is a sign multiplier, $y(x)$is the decision statistics, and ε_n is the slack variable

Lagrangian multipliers are then solved and give rise to the decision statistics by:

$y(x) = w^T x + b = \sum_{n=1}^{N} a_n t_n x_n{}^T x_n + b$

For any given observation x_n, there is a corresponding a_n that indicates which region that specific data point falls into:

- $a_n = 0 \rightarrow x_n$ does not contribute to model (falls outside of margin)
- $a_n > 0 \rightarrow x_n$ is the support vector that is necessary to the model, and in this case if:
  - $a_n < C \rightarrow x_n$ lies on the margin
  - $a_n = C \rightarrow x_n$ lies inside the margin

After training the model, we can only keep the support vectors in our model and obtain a hyper plane boundary that separates two classes. Any new observation will be classified based on which side of the plane it falls into, just like any classification rules.

Why is SVM preferred in classifying EEG signals?

Support vector machine (SVM) has been widely used for classification of electroencephalogram (EEG) signals for the diagnosis of neurological disorders due to its convex optimization process.

In BCI research, Electroencephalography (EEG) has been widely used to decode and interpret users' intentions due to its noninvasiveness, ease of use, and low cost. Of all the types of EEG-based BCI, sensorimotor rhythms (SMRs) are chosen in this case because they are readily detectable in healthy, as well as disabled individuals with neuromuscular diseases or injuries.

More importantly, SMR signals can be modulated through motor imagery (MI) tasks, which have been shown to provide a robust paradigm for generating noninvasively detectable and usable EEG signals.

SVM shows good generalization performance for high dimensional data due to its convex optimization problem, which leads to a globally optimal solution. This is in contrast with artificial neural network that often suffers from being trapped in a local minima,

On top of that, because of the high dimensional nature of EEG signals data (204 features for each observation in our case), along with the use of previous information in constructing SVM (instance-based) that gives better generalization performance, SVM is our classifier of choice.

Two-level cross validation method…

On the right is the visual representation of two-level cross validation that is applied in this project.

For the first level (outer cross-validation), we split the data into 6 folds, which renders 100×2 data for training and 20×2 data for testing. We use accuracy (probability of correct decision) as optimization metric and try to obtain as high accuracy as possible.

For the second level (inner cross-validation), we split the training data from 1st level into 5 folds, which gives 80×2 data for training and 20×2 data for testing. Here we want to find the optimized regularization parameter:

$$\lambda = \frac{1}{C}$$

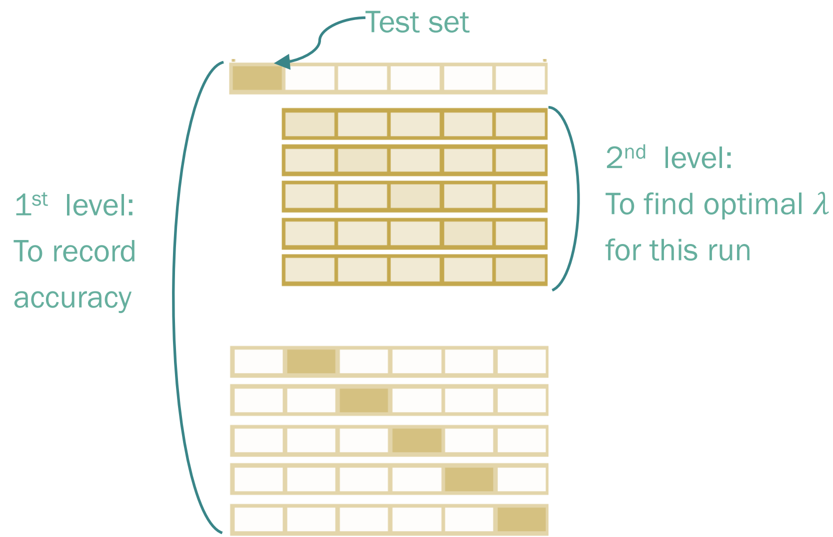

Fig. 5. Visual representation of two-level cross validation. For each run (1st level split), an optimized λ is found via 2nd level cross validation and applied back to the outer level to calculate accuracy.

2. Tibshirani, Robert. "Regression Shrinkage and Selection via the Lasso." Journal of the Royal Statistical Society. Series B (Methodological) 58, no. 1 (1996): 267-88. Accessed March 25, 2021. http://www.jstor.org/stable/2346178.

Experimental results

Imagery movement exploration

<u>*Regularization param C*</u> to cross validate

Search range: 0.01 to 10000

Optimal param for fold#1: C = 0.01

Visualizations for channel weights on 204 specific locations. Larger amplitude indicates the channel carries strong directional (left/right) information.

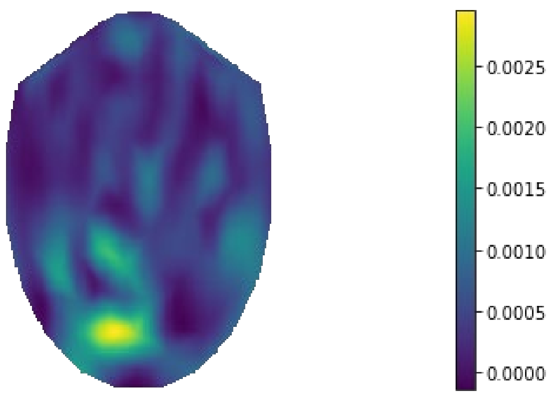

Fig. 6. Representation of channel weights on specific locations on a scalp. The coordinates are defined in accordance with the real locations that collect EEG data.

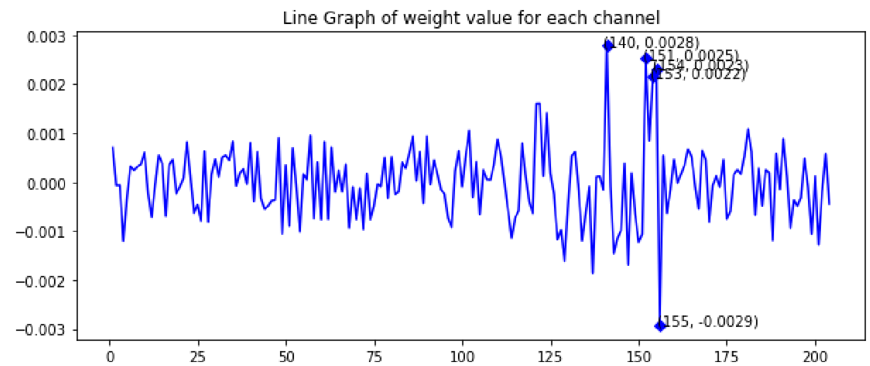

Fig. 7. Weights for 204 channels in line graph, with top 5 dominating channels marked.

Dominant channel #	Channel Weight
155	-0.0029
140	0.0028
151	0.0025
154	0.0023
153	0.0022

Table. 1. Top 5 dominating channels and corresponding weights.

section 03

Imagery movement exploration

Regularization param C to cross validate
Search range: 0.01 to 10000

Performance results after two-level cross validation. This includes accuracy of classification and the ROC curve for all 6 runs, and a total performance metric of this model as a result of cross-validation.

Run #	Accuracy	Optimal C
1	0.825	0.01
2	0.925	0.01
3	0.850	0.01
4	0.900	0.01
5	0.850	0.01
6	0.900	0.01
Total	**0.875**	**(std: 0.035)**

Table. 2. The accuracy metric for all 6 runs of 1^{st} level cross validation. The total accuracy is provided at the bottom.

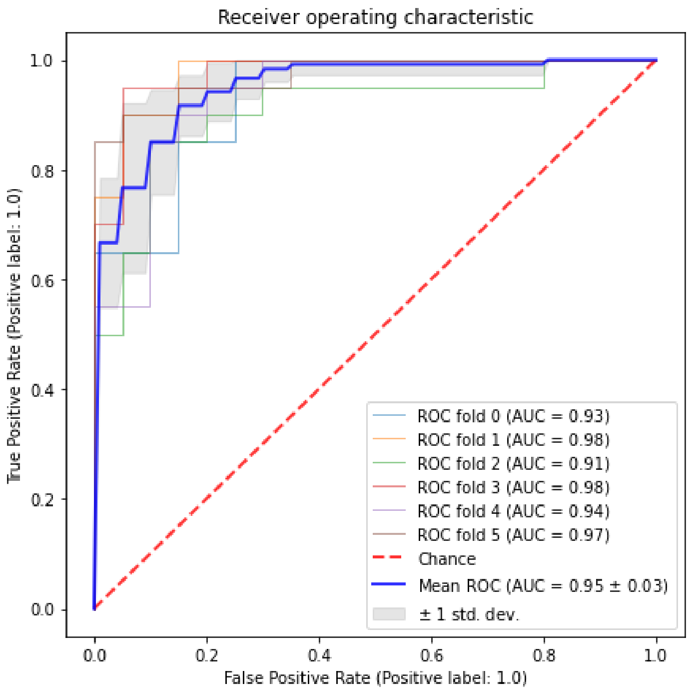

Fig. 8. ROC curves for all six runs of 1^{st} level cross validation, superimposed by the total ROC curve of the whole dataset. Any individual fold ROC is not representative enough for the total ROC, but the range of 1 standard deviation of the total ROC is bounded by the 6 ROCs.

Experimental results

Overt movement exploration

Regularization param C to cross validate

Search range: 0.01 to 10000

Optimal param for fold#1: C = 0.01

Visualizations for channel weights on 204 specific locations. Larger amplitude indicates the channel carries strong directional (left/right) information.

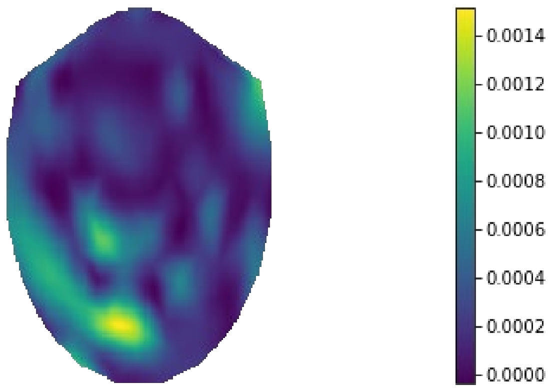

Fig. 9. Representation of channel weights on specific locations on a scalp. The coordinates are defined in accordance with the real locations that collect EEG data.

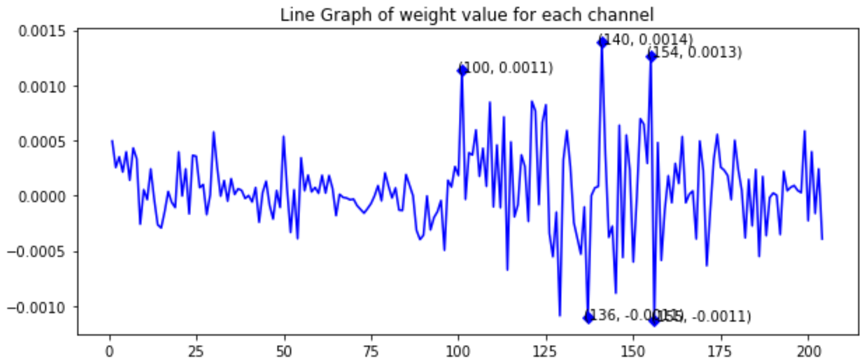

Fig. 10. Weights for 204 channels in line graph, with top 5 dominating channels marked.

Dominant channel #	Channel Weight
140	0.0014
154	0.0013
100	0.0011
155	-0.0011
136	-0.0011

Table. 3. Top 5 dominating channels and corresponding weights.

section 03

Overt movement exploration

Regularization param C to cross validate

Search range: 0.01 to 10000

Performance results after two-level cross validation. This includes accuracy of classification and the ROC curve for all 6 runs, and a total performance metric of this model as a result of cross-validation.

Run #	Accuracy	Optimal C
1	0.950	0.01
2	0.900	0.01
3	0.975	0.01
4	0.925	0.01
5	1.000	0.01
6	0.950	0.01
Total	**0.950**	**(std: 0.032)**

Table. 4. The accuracy metric for all 6 runs of 1st level cross validation. The total accuracy is provided at the bottom.

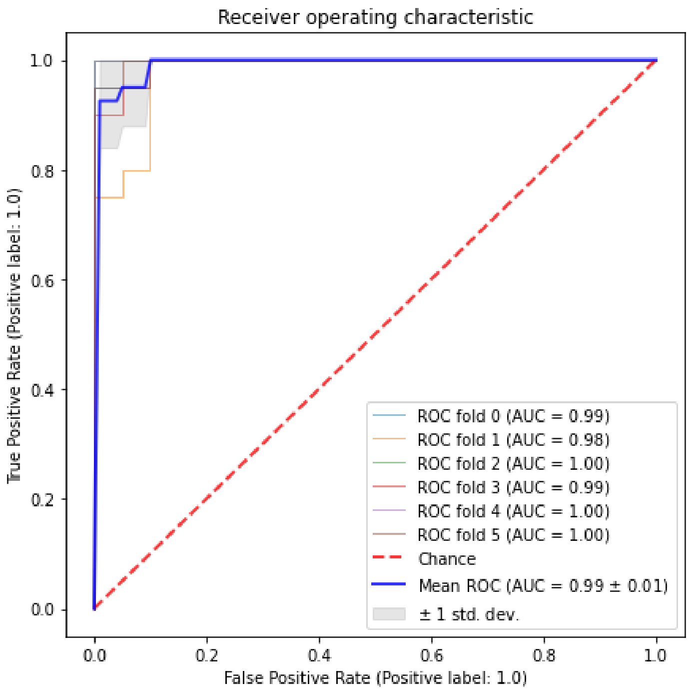

Fig. 11. ROC curves for all six runs of 1st level cross validation, superimposed by the total ROC curve of the whole dataset. Any individual fold ROC is not representative enough for the total ROC, but the range of 1 standard deviation of the total ROC is bounded by the 6 ROCs.

Imagery movement exploration

[Regularization param C](#) to cross validate

Search range: 1e-8 to 0.01

Optimal param for fold#1: C = 2.47e-7

Visualizations for channel weights on 204 specific locations. Larger amplitude indicates the channel carries strong directional (left/right) information.

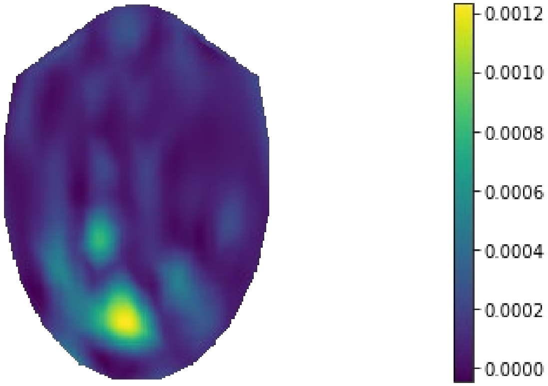

Fig. 12. Representation of channel weights on specific locations on a scalp. The coordinates are defined in accordance with the real locations that collect EEG data.

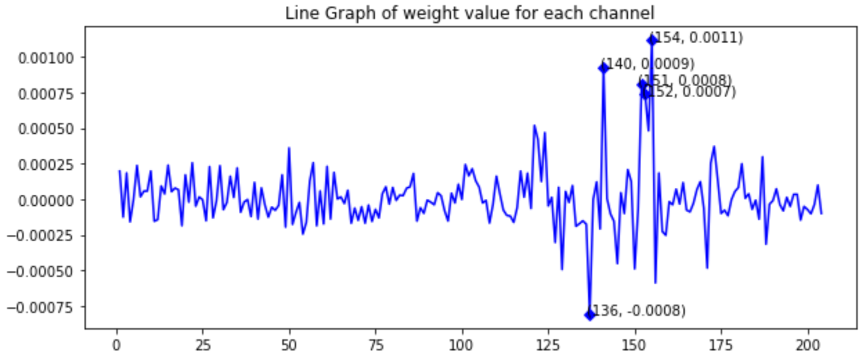

Fig. 13. Weights for 204 channels in line graph, with top 5 dominating channels marked.

Dominant channel #	Channel Weight
154	0.0011
140	0.0009
151	0.0008
136	-0.0008
152	0.0007

Table. 5. Top 5 dominating channels and corresponding weights.

Imagery movement exploration

Regularization param C to cross validate

Search range: 1e-8 to 0.01

Performance results after two-level cross validation. This includes accuracy of classification and the ROC curve for all 6 runs, and a total performance metric of this model as a result of cross-validation.

Run #	Accuracy	Optimal C
1	0.825	2.47e-7
2	0.900	7.05e-6
3	0.850	7.56e-7
4	0.900	8.70e-7
5	0.850	2.66e-6
6	0.875	1.23e-7
Total	**0.867**	**(std: 0.028)**

Table. 6. The accuracy metric for all 6 runs of 1st level cross validation. The total accuracy is provided at the bottom.

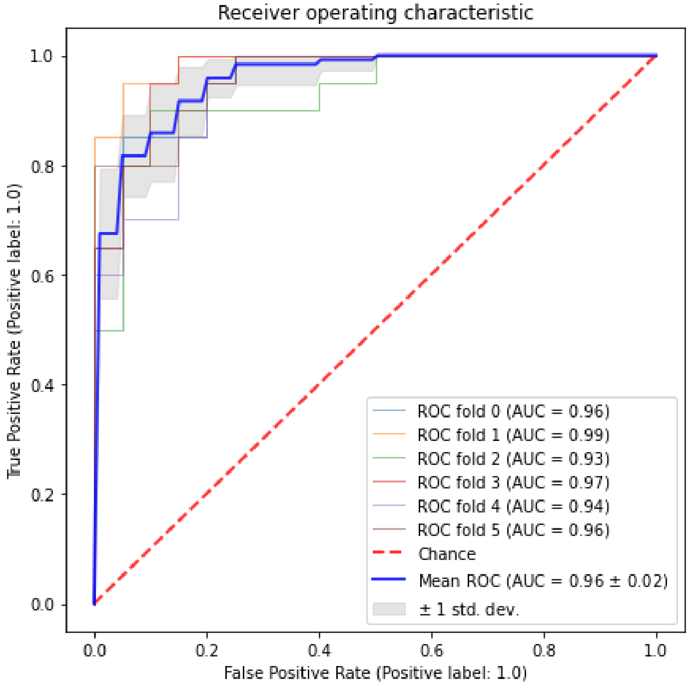

Fig. 14. ROC curves for all six runs of 1st level cross validation, superimposed by the total ROC curve of the whole dataset. Any individual fold ROC is not representative enough for the total ROC, but the range of 1 standard deviation of the total ROC is bounded by the 6 ROCs.

Overt movement exploration

Regularization param C to cross validate

Search range: 1e-8 to 0.01

Optimal param for fold#1: C = 8.70e-7

Visualizations for channel weights on 204 specific locations. Larger amplitude indicates the channel carries strong directional (left/right) information.

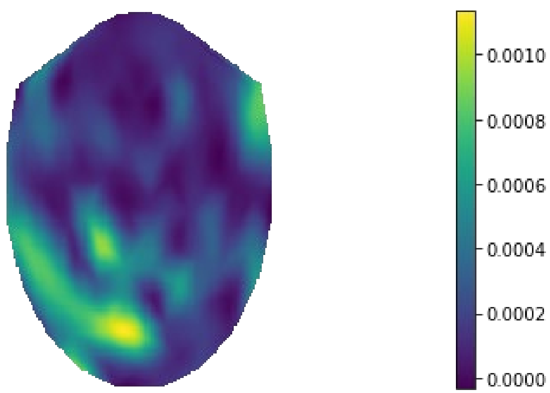

Fig. 15. Representation of channel weights on specific locations on a scalp. The coordinates are defined in accordance with the real locations that collect EEG data.

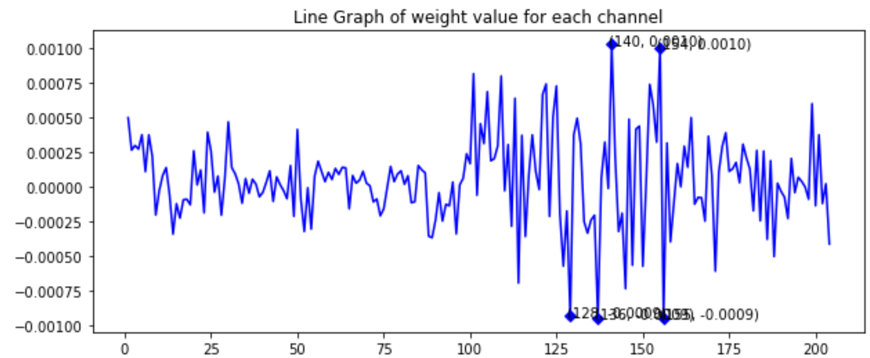

Fig. 16. Weights for 204 channels in line graph, with top 5 dominating channels marked.

Dominant channel #	Channel Weight
140	0.0010
154	0.0010
155	-0.0009
136	-0.0009
128	-0.0009

Table. 7. Top 5 dominating channels and corresponding weights.

Overt movement exploration

Regularization param C to cross validate

Search range: 1e-8 to 0.01

Performance results after two-level cross validation. This includes accuracy of classification and the ROC curve for all 6 runs, and a total performance metric of this model as a result of cross-validation.

Run #	Accuracy	Optimal C
1	0.975	8.70e-7
2	0.900	2.01e-6
3	0.975	8.70e-7
4	0.975	1.15e-6
5	1.000	8.70e-7
6	0.950	1.15e-6
Total	**0.963**	**(std: 0.031)**

Table. 8. The accuracy metric for all 6 runs of 1st level cross validation. The total accuracy is provided at the bottom.

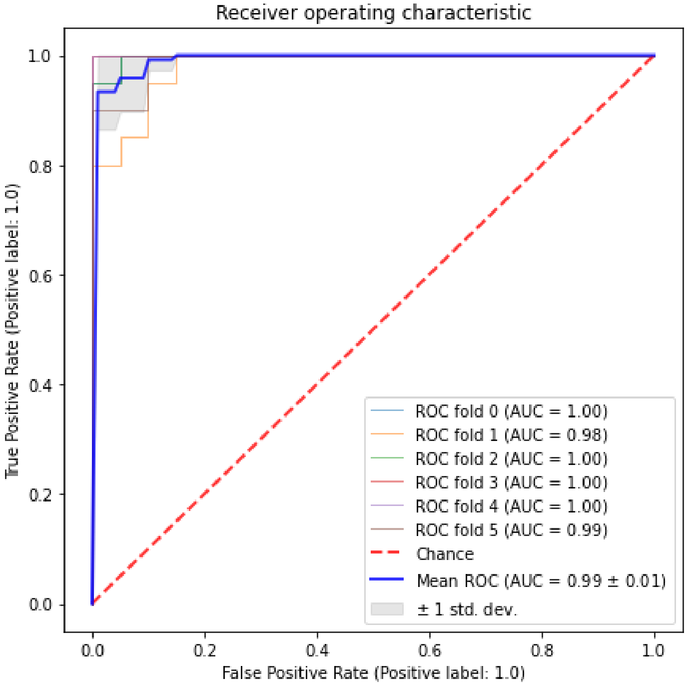

Fig. 17. ROC curves for all six runs of 1st level cross validation, superimposed by the total ROC curve of the whole dataset. Any individual fold ROC is not representative enough for the total ROC, but the range of 1 standard deviation of the total ROC is bounded by the 6 ROCs.

Comparison of regularization parameter

As I search for optimal regularization parameter C through two different types of data within the requested range, I found in both cases they converge to $C = 0.01$

This drives me to explore smaller search range of regularization parameter. The result turned out to be quite interesting – The optimal C where the model has the best performance in terms of accuracy converge to a very small number 8.7e-7 in both cases.

The accuracy under such parameter also increases, indicating this is more likely to be a better selection than 0.01.

The implication of a very small regularization parameter selection is that the dataset is relatively separable in the high dimensional space, so a smaller C will in turn improve the performance of the model.

Data Type		Imagery Movement	Overt Movement
Range 0.01~1e4	**Opt. C**	0.01	0.01
	Accuracy	0.875	0.950
Range 1e-8~0.01	**Opt. C**	8.70e-7	8.70e-7
	Accuracy	0.867	0.963

5. Ong, Frank. JPEG DCT Demo, n.d. https://inst.eecs.berkeley.edu/~ee123/sp16/Sections/JPEG_DCT_Demo.html.

Patterns for dominating channels

There are some patterns for dominating channels in both types of dataset. The first is their localities.

Based on the table on the right, we can notice that there is a 60% overlapping in the top 5 dominating channels, and they happen to locate near the bottom left of the scalp image. The percentage will rise higher if we take, for example, top 10 dominating channels.

This result tells us that for this particular experiment, no matter what the movement the tester was asked to perform, it has a high correlation in that region of human's brain, which aligns with our initial expectations for the MI task of BCI.

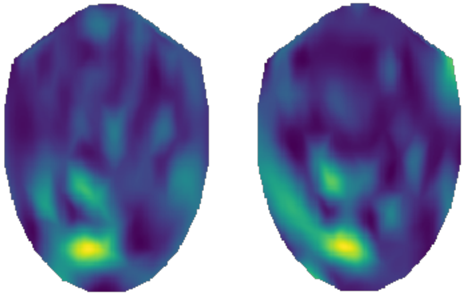

Fig. 18. Left image is the channel weights display of imagery movement; On the right is that of overt movement.

Imagery Data Dominant channel (weight)		Overt Data Dominant channel (weight)	
140	-0.0029	140	0.0014
151	0.0028	154	0.0013
154	0.0025	100	0.0011
153	0.0023	155	-0.0011
155	0.0022	136	-0.0011

5. Ong, Frank. JPEG DCT Demo, n.d. https://inst.eecs.berkeley.edu/~ee123/sp16/Sections/JPEG_DCT_Demo.html.

Difference in model accuracy

The first thing to note about the difference in model accuracy is that overt movement data, under the same regularization parameter ($C = 0.01$), has higher accuracy compared to imagery movement data.

This statement still holds true even if we select a much smaller regularization parameter ($C = 8.70e-7$). This result fits with our expectation that overt movement EEG signals, since the tester actually performed the movement, should be easier to distinguished and in turn make the model accuracy higher.

ROC results support this statement as well. The Area Under Curve (AUC) of overt movement dataset is higher than that of imagery movement dataset.

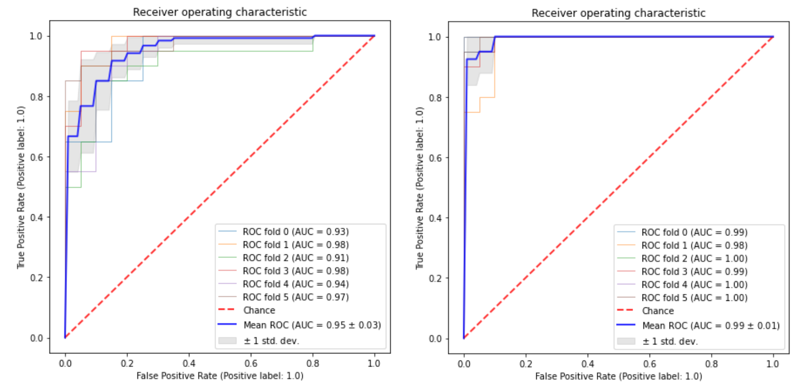

Fig. 19. Left image is the ROC curve for imagery movement data; right is that for overt movement data.

Data Type		Imagery Movement	Overt Movement
Range 0.01~1e4	**Opt. C**	0.01	0.01
	Accuracy	0.875	0.950
Range 1e-8~0.01	**Opt. C**	8.70e-7	8.70e-7
	Accuracy	0.867	0.963

section 04

5. Ong, Frank. JPEG DCT Demo, n.d. https://inst.eecs.berkeley.edu/~ee123/sp16/Sections/JPEG_DCT_Demo.html.

Thoughts beyond the results...

The datasets used in this project was well-designed and readily processed. In real-world scenario, there are several factors that might impact the results and expose the limits of this method.

One shortcoming of SVM classifier is that there is **no consideration about the distribution of data** in constructing of SVM. Therefore, the classifier is trained for in the same manner for all types of datasets without having the knowledge about the data distribution. Another factor that we need to consider is the **high sensitivity of imbalanced data** of SVM, which tends to be the case for real-world data.

Moreover, SVM can also be subject to the **presence of outliers**. As we explored it in previous assignments, outliers tend to compromise the performance of instance-based classifiers.

If we could have prior knowledge of the distribution of data, whether data is imbalanced or whether there are outliers, we could apply different modifications of SVM to deal with those problems. For example, **universum support vector machine (USVM)** provides the prior distribution information to the classifier and the hyperplane will adjust accordingly with distribution; **Twin SVM** has the advantage of being insensitive to class imbalanced data; Lastly, by selecting the universum from the interictal EEG signals, we would be able to ensure the classifier is not impacted by outliers.

References

1. He, Bin, Bryan Baxter, Bradley J. Edelman, Christopher C. Cline, and Wenjing W. Ye. “Noninvasive Brain-Computer Interfaces Based on Sensorimotor Rhythms.” *Proceedings of the IEEE* 103, no. 6 (2015): 907–25. https://doi.org/10.1109/jproc.2015.2407272.

2. Richhariya, B., and M. Tanveer. “EEG Signal Classification Using Universum Support Vector Machine.” *Expert Systems with Applications* 106 (2018): 169–82. https://doi.org/10.1016/j.eswa.2018.03.053.

3. Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke & Travis E. Oliphant. Array programming with NumPy, Nature, 585, 357–362 (2020)

4. Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010)

5. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12, 2825-2830 (2011)

6. Fernando Pérez and Brian E. Granger. IPython: A System for Interactive Scientific Computing, Computing in Science & Engineering, 9, 21-29 (2007)

References

7. D. Hunter, "Matplotlib: A 2D Graphics Environment," in Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, May-June 2007, doi: 10.1109/MCSE.2007.55.

8. Brownlee, Jason. “Nested Cross-Validation for Machine Learning with Python.” Machine Learning Mastery, January 6, 2021. https://machinelearningmastery.com/nested-cross-validation-for-machine-learning-with-python/.

Collaborations

- I discussed and shared ideas with Saksham Jain (sj305) before starting this project.
- I shared code with Saksham Jain (sj305) while working on the project
- I compared results with Saksham Jain (sj305) while working on the project.
- I received help mostly from Saksham Jain (sj305) and TAs while working on this project.

THANK YOU