Tuned Differential Entropy Feature and EEG-based Emotion Classifer

Lee Chanyoung

Abstract—This paper outlines the different combinations of data scaling, classification models, feature selection, and dimensional reduction algorithms that we take to tune a classifying model to achieve an accurate emotion classification. After experimenting with different combinations, the results show that the combination of a Standard Scaler, SGD classifier, mRMR(MIQ) feature selection, and an LDA dimension reduction is the optimal method to achieve maximum accuracy and minimum fold difference on a 15 fold cross-validation test. The model resulted in high precision of 99.48%, with a minimum accuracy difference between each fold of 2.74%.

Index Terms—EEG emotion classification, Differential Entropy, Machine Learning, Classification.

1 Introduction

The purpose of a machine-learning emotion classifier is to enable a machine to recognize, understand and represent a user's emotional state. This is a hot topic in emotional AI research, which contains multiple fields of study such as neural science, psychology, data science. The EEG-based emotion classifier is a representative problem for classification. Unlike the traditional data - image and sound - EEG data is unstable due to noise and its discrete characteristic, thus a challenging research topic. This paper extends on the proposal of a new EEG feature named differential entropy (DE), which claims to be a more stable and accurate EEG feature than traditional features such as the energy spectrum.[1] We will be attempting to create a three-class classifier that takes in the DE data and outputs either a "positive," "neutral," or "negative" response.

This paper will outline the steps taken and discoveries while attempting to improve the accuracy of the model. A detailed comparison of which combinations of preprocessing tools and classifiers are used to achieve our goal will be presented.

1.1 Input Data Description

The data used in this research origins from the SJTU emotion EEG dataset, SEED. We will be using a subset of the dataset that contains 15 samples of the participant's data. The EEG data is recorded while the participants watch 5 separate films that evoke each of the three emotions: positive, neutral, negative. The DE feature has 62 outputs which were read every second. The feature has a dimension of (62,5), therefore contains 310 features to feed the model. Throughout this research, testing will be conducted with 15-fold cross-validation.

2 Base Model

For the base model, the original data will not be preprocessed. We will feed the raw 15 samples of data to the prelearned classifier, Sklearn's SVC (RBFkernel), and train it.

310 features are a lot of data to train. This causes the base model to take an extended amount of time to train.

The mean accuracy of our base model is 61.22% and Figure 1 shows the results for the 15 fold cross-validation. This score is not desirable, so we shall experiment with preprocessing our data to gain a higher accuracy.

Fold	1	2	3	4	5
Accuracy	32.52%	58.22%	49.79%	79.08%	51.59%
Fold	6	7	8	9	10
Accuracy	61.10%	68.62%	62.72%	53.59%	42.86%
Fold	11	12	13	14	15
Accuracy	82.17%	50.61%	91.30%	61.40%	72.74%

Figure 1. Base model 15-fold accuracy

3 Preprocessing Choices

Analyzing data that has not been carefully screened for specific problems can produce misleading results. Thus, the representation and quality of data are first and foremost before running any analysis.[2]

The steps below will explain what we learned of the DE data and how we decided to tune it to suit our needs.

Also, to shorten the training time, we will select a subset of features using the minimal-redundancy-maximal-relevance algorithm. Using the subset as our input, we will compare the accuracy of using different dimension reduction methods and scalings and decide which combination best suits the task.

3.1 Scaler Choice

For the model, there are three different scaler options: Min-Max, Standard, and Normalizer.

Since the range of raw data values varies widely, objective functions will not work properly without scaling in some machine learning algorithms. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be scaled so that each feature contributes approximately proportionately to the final distance.

Another reason why feature scaling is applied is that gradient descent converges much faster with feature scaling than without it.[3]

Scaling the feature with the MinMaxScaler by itself raised the accuracy of the base model to 74%. Combining the scaling, feature selection, and dimension reduction can drastically increase the model's accuracy and training speed.

The effect of each scaler on the accuracy of the model will be shown in the experiment section.

3.2 mRMR Feature Selection

Feature selection identifies subsets of data relevant to the parameters used and is usually called Maximum Relevance.

Features can be selected in many different ways. One scheme is to choose features that correlate strongest to the classification variable. This is called the maximum-relevance selection. The mRMR is a maximum-relevance selection heuristic that also selects features that are far away from each other while still having a "high" correlation to the classification variable.

The mRMR has two options: MID, MIQ. Mutual Information Difference and Quotient scheme, respectively. They define the relevance and redundancy when using Mutual Information.[4]

mRMR(MID) and mRMR(MIQ) selects subsets with subtle differences, and their comparison will be made in our experiment section.

3.3 Dimension Reduction Algorithms

3.3.1 *Principal Component Analysis (PCA)*

PCA is a technique for reducing the dimensionality of a dataset that is hard to interpret. It aims to increase interpretability but at the same time minimizes information loss. It does so by creating new uncorrelated variables that maximize variance, called Principal Components of the data. [5]

Figure 2 shows the plotted data after a 2D, and 3D PCA dimension reduction algorithm has been applied. It is clear to see that neither can be linearly or polynomially separable.

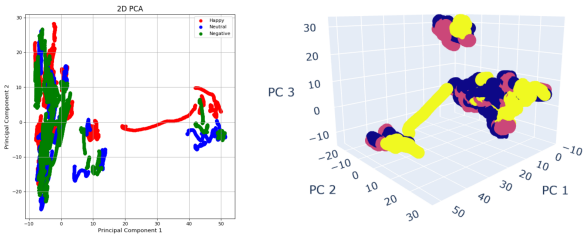


Figure 2. 2D and 3D PCA Plot

3.3.2 *Independent Component Analysis (ICA)*

ICA also reduces a large dataset into a smaller dimension that can be relatively easily understood. Unlike PCA, which assumes that the components are uncorrelated in both spatial and temporal domains, ICA components are maximally statistically independent in only one domain.[6]

PCA aims to minimize information loss, while ICA aims to maximize each feature's independence. Suppose x,y are two random variables, and their distribution functions are given by Px, Py respectively. If we receive some information about x and that doesn't change whatever knowledge we have about y, then we can safely say that x, y are independent variables.

The plotted data of a 2D and 3D ICA graph can be observed in Figure 3.

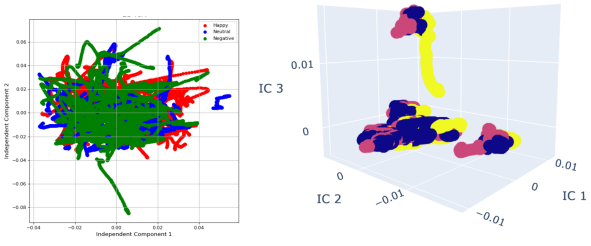


Figure 3. 2D and 3DICA Plot

Similar to PCA, the ICA dimension reduction does not achieve any separability. The accuracy after using ICA can also be seen in the experiment section.

3.3.3 *Linear Discriminant Analysis (LDA)*

LDA is also a dimension reduction algorithm, but LDA is developed to maximize the ratio of the between-class variance to the within-class variance, unlike the prior two methods. This guarantees maximum class separability.[7]

The LDA results with the plot shown in Figure 4. This figure shows that the plotted data is more separable than the prior two methods, which will result in better performance. The actual accuracy for each of the methods mentioned above can be seen in the experiment section.

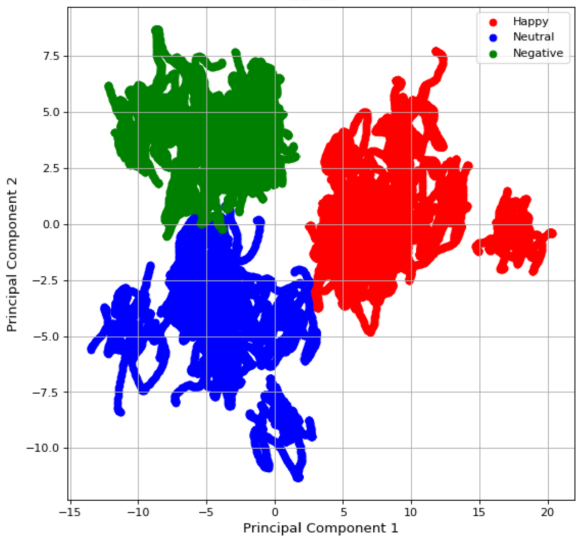


Figure 4. 2D LDA Plot

One crucial point that we discovered while organizing the data is that the LDA algorithm must be executed on each sample set individually to maximize separability.

LDA maximizes the difference between the classes. In Figure 5, the X and y data were all unified into one array and then processed through the LDA algorithm. Our resulting plot is not as separable as our previous example.

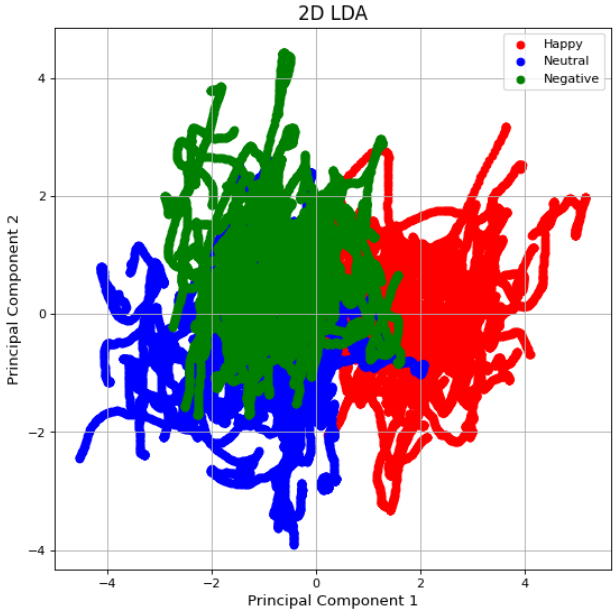


Figure 5. Unified LDA Plot

This is perhaps due to the innate difference between individuals, such as mental status, stress levels, and other physical or psychological conditions. This is a matter of neural science, which we have no authority to comment on. But with this thought, we attempted to fit each sample individually. We discovered that doing so resulted in a better performance, which abides by the theory that maximizing the difference of the three classes for each sample prior to unification will increase the clarity of the data.

4 Experiment

This section will show the experiments we ran on the model using several different transfer learning models: libsvm, sklearn linear SVM, RBF kernel SVM, K-Neighbors , SGD, and MLP. We will be using a 15 fold cross-validation on our 15 samples to test our model's accuracy. We will compare the different combinations of scalers and models that maximize the mean accuracy and minimize the fold differences.

4.1 mRMR(MIQ) Feature Selected Data Testing

mRMR feature selection has two variations. For this section, We will test the 28 features chosen by the MIQ variation. The number 28 was selected after testing a few combinations of the first 32 features chosen by mRMR.

This section was not decided by accurate testing. The number of features that we can select ranges from one to three hundred and ten. There were too many variations to test, so the number 28 was chosen without data backing its credibility.

We welcome anyone to question, research, or test our decision.

4.1.1 *PCA and Scaler Accuracy Test*

Figure 6 and 7 represents the 2D and 3D PCA results using each scaler.

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	49.26%	52.02%	57.66%

Model	K-Neighbor	SGD	MLP
AvgAccuracy	54.04%	58.06%	**59.46%**

(a) STD Scaler

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	47.73%	51.45%	**58.85%**

Model	K-Neighbor	SGD	MLP
AvgAccuracy	55.71%	55.08%	58.46%

(b) Normalizer

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	49.56%	50.73%	52.74%

Model	K-Neighbor	SGD	MLP
AvgAccuracy	49.52%	**56.09%**	50.05%

(c) MinMax Scaler

Figure 6. 2D PCA and scaler combinations

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	48.80%	49.97%	**57.37%**

Model	K-Neighbor	SGD	MLP
AvgAccuracy	52.74%	56.01%	54.77%

(a) STD Scaler

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	45.71%	50.91%	**59.42%**

Model	K-Neighbor	SGD	MLP
AvgAccuracy	53.93%	55.67%	56.83%

(b) Normalizer

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	47.80%	48.35%	50.15%

Model	K-Neighbor	SGD	MLP
AvgAccuracy	51.71%	**52.52%**	51.24%

(c) MinMax Scaler

Figure 7. 3D PCA and scaler combinations

From the figures above, it is evident that PCA does not benefit the accuracy of our model. Pondering why this may be so, we came to learn of a metric called explained variance. Explained variance is the fraction of variance explained by a principal component is the ratio between the principal component's variance and the total variance. The more variance, the more information of the original data remains.

Questioning whether a higher explained variance will result in a more accurate machine. We decided to learn how many components we would need to have.

A 2D PCA model has an explained variance of 64%. This means that a 2-dimensional reduction only captured 64% of our original data.

From figure 8, we can see that the explained variance cumilates as the number of components (dimensions) increase.

According to the graph, 13 components result in an explained variance of 97%.

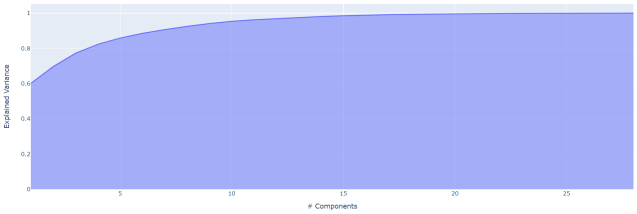


Figure 8. Explained Variance Area Curve.

To test our hypothesis, we shall run the PCA dimension reduction with 13 components and the results will be shown in Figure 9.

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	60.71%	60.08%	54.36%

Model	K-Neighbor	SGD	MLP
AvgAccuracy	51.13%	60.16%	53.77%

(a) STD Scaler

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	58.99%	59.40%	59.44%

Model	K-Neighbor	SGD	MLP
AvgAccuracy	57.07%	58.94%	55.29%

(b) Normalizer

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	56.21%	57.12%	52.91%

Model	K-Neighbor	SGD	MLP
AvgAccuracy	47.33%	56.75%	47.02%

(c) MinMax Scaler

Figure 9. 16 component PCA and scaler combinations

From all the figures above in this section, it is evident that our model's accuracy is less than our base model when the dimension is reduced by PCA. As we know, PCA maximizes the information held within the data. This apparently does not seem to be a crucial factor when reducing the dimensions of our data. Next, we will test whether the independence of that data has a higher impact on the data than PCA.

4.1.2 *ICA and Scaler Accuracy Test*

Figures 10 and 11 below shows the accuracy of the 2D and 3D ICA data scaled with different scalers.

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	34.09%	**35.11%**	31.30%

Model	K-Neighbor	SGD	MLP
AvgAccuracy	32.87%	30.41%	31.47%

(a) STD Scaler

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	12.67%	12.72%	33.01%

Model	K-Neighbor	SGD	MLP
AvgAccuracy	32.96%	25.99%	**33.39%**

(b) Normalizer

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	26.89%	26.98%	26.69%

Model	K-Neighbor	SGD	MLP
AvgAccuracy	29.22%	**31.05%**	29.37%

(c) MinMax Scaler

Figure 10. 2D ICA and scaler combinations

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	41.31%	41.36%	**44.34%**

Model	K-Neighbor	SGD	MLP
AvgAccuracy	39.17%	42.13%	42.33%

(a) STD Scaler

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	**44.71%**	44.13%	40.61%

Model	K-Neighbor	SGD	MLP
AvgAccuracy	38.97%	41.64%	40.36%

(b) Normalizer

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	34.01%	33.29%	34.24%

Model	K-Neighbor	SGD	MLP
AvgAccuracy	**38.02%**	34.14%	32.01%

(c) MinMax Scaler

Figure 11. 3D ICA and scaler combinations

The figures above show that the ICA reduction algorithm does not help but hinders the model's accuracy.

After research on ICA dimension reduction, we found that whitening is preferred before feeding in the data.

The goal of ICA is to rotate the data (unitary transform) so that each axis looks as non-Gaussian as possible. Gaussian data yet looks Gaussian after rotation. If we don't "center" (whiten) the data, all the algorithm can really do is rotate the block distribution of the data to one axis. By bringing the mean to zero (centering) and normalizing the variance in all directions (whitening), we give the algorithm freedom to rotate in all directions.

This process can be done by PCA. Therefore, to test this "preferred" method. We will run PCA on all features and then reduce the data dimension.

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	31.82%	31.59%	30.88%
Model	K-Neighbor	SGD	MLP
AvgAccuracy	34.55%	23.78%	32.27%

(a) PCA whitening and 2D ICA (STD)

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	35.40%	36.50%	30.44%
Model	K-Neighbor	SGD	MLP
AvgAccuracy	31.72%	36.74%	31.69%

(b) PCA whitening and 3D ICA (STD)

Figure 12 PCA whitening and ICA test

Even though we whitened the data, the accuracy did not seem to improve. We concluded that this method will not drastically change the accuracy just by changing the scaler.

Increased independence of the data did not seem to improve the model's accuracy. Instead, it performed worse than PCA.

4.1.3 *LDA and Scaler Accuracy Test*

The LDA dimension reduction algorithm maximizes the separation of different classes. This method has successfully modified the data to be somewhat distinct in 2 dimensions.

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	99.33%	98.32	98.02%
Model	K-Neighbor	SGD	MLP
AvgAccuracy	98.57%	**99.48%**	98.54%

(a.1) STD Scaler Accuracy

Model	SKlearn SVM	LIBSVM	RBF
MaxFoldDiff	-3.74%	-20.80	-25.01%
Model	K-Neighbor	SGD	MLP
MaxFoldDiff	-15.26%	**-2.74%**	-17.38%

(a.2) STD Scaler Fold Difference

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	98.89%	98.82%	98.81%
Model	K-Neighbor	SGD	MLP
AvgAccuracy	98.34%	**98.96%**	98.82%

(b.1) Normalizer Accuracy

Model	SKlearn SVM	LIBSVM	RBF
MaxFoldDiff	-11.40%	-11.90%	-12.10%
Model	K-Neighbor	SGD	MLP
MaxFoldDiff	-18.06%	**-9.78%**	-12.96%

(b.2) Normalizer Fold Difference

Model	SKlearn SVM	LIBSVM	RBF
AvgAccuracy	97.71%	97.02%	97.23%
Model	K-Neighbor	SGD	MLP
AvgAccuracy	**98.09%**	98.02%	96.43%

(c.1) MinMax Scaler Accuracy

Model	SKlearn SVM	LIBSVM	RBF
MaxFoldDiff	-18.17%	-31.82%	-29.81%
Model	K-Neighbor	SGD	MLP
MaxFoldDiff	-15.49%	**-14.99%**	-31.37%

(c.2) MinMax Scaler Fold Difference

Figure 13. LDA and scaler combinations

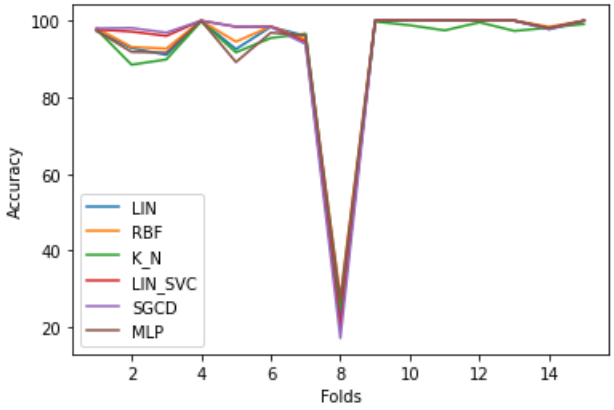
From Figure 13, we can see that the combination of an SGD classifier with the LDA results in the highest accuracy from the Figure above. High accuracy by itself does not determine the best result. Another metric should also be considered to obtain the "optimal" combinations. The max difference between each fold needs to be accounted for. In the MinMax scaler case with the libsvm classifier: it has a high average accuracy of 97.02%, but one of the folds resulted in a 68.17% accuracy. This difference may cause the model to be ultimately unreliable. If our input data is majorly composed of data similar to the fold, close to half of its predictions would be wrong. Therefore, the difference between the folds is also a crucial metric.

The combination of an SGD classifier and STD scaler has the highest of both metrics.

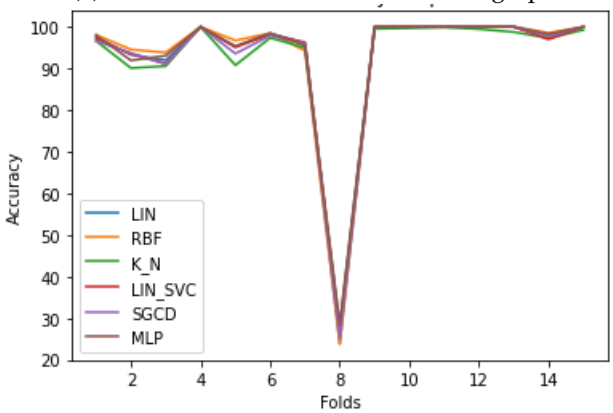
4.2 mRMR(MID) Feature Selected Data Testing

Now let us use the MID features to test the model. We decided to use 23 features. Less than 23 causes the minimal fold accuracy to decrease to 4%. More than 23, we saw a decreasing trend. As for the MIQ feature selection, we did not test for more than 30 features. This point is left for more experimenting and validation. Please feel free to question us for more detail.

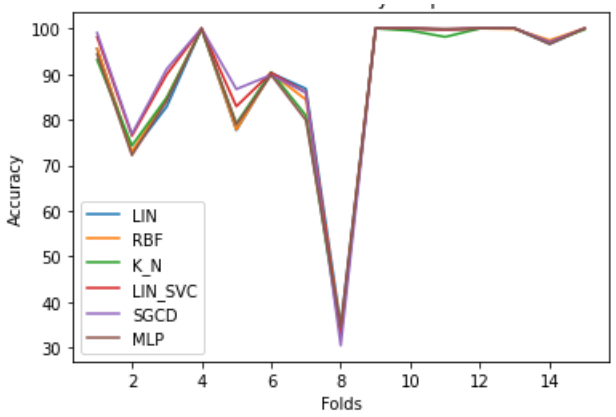
From our previous section, we know that both PCA and ICA are unbeneficial in improving the data's clarity and model's accuracy. Thus, we will test the MID features with the LDA dimension reduction algorithm to see which of the two is able to achieve a higher performance.



(a) std scaler 15 fold cross-validation graph



(b) normalized 15 fold cross-validation graph



(c) minmax 15 fold cross-validation graph

Figure 14. mRMR(MID), LDA and scaler combinations

The MID features seem to not cover a crucial feature for one of our folds. The difference between the folds was visible. Therefore, we used a line graph instead of detailed numbers.

Figure 14 shows that on the 8th fold, the model's accuracy steeply declines. Even after adding more mRMR(MID) features, the accuracy of the 8th fold did not improve. Instead, it decreased.

This shows that the MIQ features are more suitable for our model. The mean accuracy was around 90%, which is lower than any combination with the MIQ features.

5 Final Model Test

As shown in the experiment section, the combination of a MIQ mRMR feature selection, LDA dimension reduction,std scaler, and an SGD classifier results in the maximal mean accuracy and minimal fold accuracy difference. We created a transfer learning model, named GKC.

The decision surface of our model on the data is plotted in Figure 15.

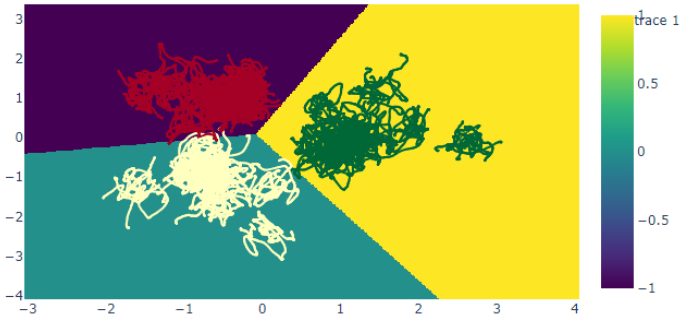


Figure 15. GCK Model Decision Surface

The yellow is the positive emotions, the purple is the negative and indigo is the neutral. The inaccuracy appears at the borders of each of the three classes.

6 Conclusion

After experimenting with the various methods and algorithms, we can conclude that an mRMR(MIQ) feature selection and LDA dimension reduction can achieve a high accuracy despite the scalar. The feature selection and dimension reduction drastically decreased the training time of the model.

The scalar played an important part when we test for the fold difference. Some scalars can cause some models to have massive fold differences, while it improves others' accuracy.

Finally, we learned that when working with a DE feature for EEG data, the LDA can support us to make the data a step more isolated. We also learned that LDA should be performed on each individual's dataset to further increase its isolation.

When we saw that our model's accuracy achieved a high level at 99%, we first wondered how this was possible. Other discoveries listed their accuracy to wander around in the 80s. After some research, we concluded that the reason for the accuracy is due to the 'simplicity' of our data. We only consider there to be three emotions to classify, but there are numerous types of emotions in reality. How do we know the subset of emotions? Emotions are non-binary and vague, but our dataset discarded the various emotions that may emerge while watching a movie.

Fortunately, our data was able to cleanly distribute data into separate sections. The vagueness comes into consideration in borderline cases. When positivity mixes with negativity, doubt mixes with trust, or maybe a mixture of more emotion, classification becomes a more challenging question to tackle.

Acknowledgments

The authors would like to thank STJU's professor Bao-Liang Lu that provided that BCMI dataset and gave us this assignment to a step further into the machine learnign field.

References

[1] Duan, Ruo-Nan, Jia-Yi Zhu and Bao-Liang Lu. "Differential Entropy feature for EEG-based emotion classification" *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)*(2013):81-84

[2] Pyle, D. "Data Preparation for Data Mining" *Morgan Kaufmann Publishers, Los Altos, California.*(1999)

[3] Ioffe, Sergey; Christian Szegedy (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift" . *arXiv:1502.03167*

[4] Hanchuan Peng, Fuhui Long, Chris Ding. "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy" *IEEE Transaction on Pattern Analysis and Machine Intelligence (2005)*(Vol. 27 No.8):1226-1238

[5] Joliffe Ian T., Cadima Jorge. "2016 Principal component analysis: a review and recent developments" *Phil. Trans. R. Soc. A*.374:20150202

[6] Anjali R. Beharelle, Steven L. Small. "Imaging Brain Networks for Language" *Neurobiology of Language* (2016)

[7] Tharwat, Alaa, Gaber, Tarek, Ibrahim, Abdelhameed, Hassanien, Aboul Ella. "Linear discriminant analysis: A detailed tutorial" *AI Communications* 30.169-190 10.3233/AIC-170729

Chanyoung Lee Student ID: 518030990041

Chanyoung Lee is a junior, computer science major, international student attended Shanghai Jiao Tong University.